# MXO 3 Series Oscilloscope User Manual

dataTec
Mess- und Prüftechnik. Die Experten.

**Ihr Ansprechpartner / Your Partner:**

**dataTec AG**
E-Mail: info@datatec.eu
datatec.eu




1335308602
Version 07

ROHDE&SCHWARZ
Make ideas real

This document describes the following MXO 3 series models with firmware version 2.10.x.x:

- MXO38 (1335.2050.08)
- MXO34 (1335.2050.04)

Mess- und Prüftechnik. Die Experten.

**Ihr Ansprechpartner /**
**Your Partner:**

**dataTec AG**
E-Mail: info@datatec.eu
**datatec.eu**

© 2026 Rohde & Schwarz
Muehldorfstr. 15, 81671 Muenchen, Germany
Phone: +49 89 41 29 - 0
Email: info@rohde-schwarz.com
Internet: www.rohde-schwarz.com
Subject to change – data without tolerance limits is not binding.
R&S® is a registered trademark of Rohde & Schwarz GmbH & Co. KG.
The terms HDMI, HDMI High-Definition Multimedia Interface, HDMI trade dress and the HDMI Logos are trademarks or registered trademarks of HDMI Licensing Administrator, Inc.
MATLAB® is a registered trademark of The MathWorks, Inc.
TEKTRONIX, TEK, and TEKPROBE are registered trademarks of Tektronix, Inc.
All other trademarks are the properties of their respective owners.

1335.3086.02 | Version 07 | MXO 3 Series

Throughout this document, R&S® is indicated as R&S.

# Contents


**1 Safety and regulatory information....23**
**1.1 Safety instructions....23**
**1.2 Labels on the product....29**
**1.3 Warning messages in the documentation....29**
**1.4 CE declaration of conformity....30**
**1.5 Where to find key documents on Rohde & Schwarz....30**
**1.6 Korea certification class A....30**
**2 Preface....31**
**2.1 Key features....31**
**2.2 Documentation overview....31**
2.2.1 Manuals and instrument help....32
2.2.2 Specifications and product brochure....32
2.2.3 Calibration certificate....33
2.2.4 Release notes, open source acknowledgment....33
2.2.5 Application notes, application cards, videos....33
**2.3 Options described in this document....33**
**3 Getting Started....35**
**3.1 Preparing for use....35**
3.1.1 Lifting and carrying....35
3.1.2 Unpacking and checking....35
3.1.3 Choosing the operating site....35
3.1.4 Setting up the product....36
3.1.5 Considerations for test setup....37
3.1.6 Connecting to power....38
3.1.7 Connecting to LAN....39
3.1.8 Connecting external devices....39
3.1.9 Switching on or off....40
**3.2 Instrument tour....41**
3.2.1 Front view....41
3.2.2 Side view....43
3.2.3 Rear view....44

3.2.4 Keys and controls....45
3.2.5 Checking the functionality....52
**4 Operating the instrument....55**
**4.1 Means of manual interaction....55**
**4.2 Touchscreen display....56**
4.2.1 Information on the display....56
4.2.2 Control elements on the GUI....58
**4.3 Applications....61**
**4.4 Toolbar....62**
4.4.1 Using the toolbar....62
4.4.2 Configuring the toolbar....63
4.4.3 Toolbar functions....63
**4.5 Signal bar....67**
**4.6 Working with waveforms....67**
**4.7 Rohde & Schwarz SmartGrid....69**
**4.8 Using the selection menu....71**
**4.9 Using dialogs....73**
**4.10 Entering data....74**
**4.11 Displaying results....76**
**4.12 Adding annotations....77**
**4.13 Information and notifications....79**
**4.14 Getting user assistance....80**
4.14.1 Displaying help....81
4.14.2 Using help....81
**5 Instrument setup....83**
**5.1 System settings....84**
5.1.1 About settings....84
5.1.2 Network settings....85
5.1.3 Network drive mapping....88
5.1.4 Remote settings....90
5.1.5 Localization settings....91
5.1.6 Date and time settings....91
**5.2 Option settings....93**

5.2.1 Software options settings....93
**5.3 Appearance settings....96**
5.3.1 Colors....96
5.3.2 Grid....99
5.3.3 Dialogs....101
5.3.4 Measure....102
5.3.5 Peak list....103
5.3.6 Miscellaneous....103
**5.4 Display settings....104**
5.4.1 Persistence settings....104
5.4.2 Signal settings....105
5.4.3 Backlight settings....106
**5.5 Front panel settings....107**
5.5.1 Hardkeys: function assignment....107
5.5.2 Knobs....108
5.5.3 LED....108
**5.6 Preset setup....109**
5.6.1 Preset settings....109
5.6.2 Factory preset....110
5.6.3 Secure erase....110
5.6.4 Restoring settings....111
**5.7 Maintenance settings....112**
5.7.1 Firmware update....112
5.7.2 Alignment....112
5.7.3 Power management....115
5.7.4 Self-test....115
5.7.5 Service....116
**5.8 Save / recall settings....117**
5.8.1 Autonaming....117
5.8.2 CSV export....119
**6 Acquisition and waveform setup....120**
**6.1 Horizontal setup....120**
6.1.1 About the horizontal system....120

6.1.2 Horizontal Setup settings ... 121
6.1.3 Zoom settings ... 122
6.1.4 Roll mode ... 122
**6.2 Acquisition ... 124**
6.2.1 About the acquisition system ... 124
6.2.2 Acquisition setup ... 126
6.2.3 Segmentation settings ... 131
6.2.4 High definition mode ... 132
6.2.5 History settings ... 133
6.2.6 Speed ... 133
**6.3 Vertical setup ... 134**
6.3.1 About the vertical system ... 135
6.3.2 Vertical Setup settings ... 135
6.3.3 Bandwidth settings ... 138
6.3.4 Probe settings ... 139
6.3.5 Other vertical settings ... 139
**6.4 Probes ... 140**
6.4.1 Common probe settings ... 141
6.4.2 Setup for passive probes ... 143
6.4.3 Setup for active voltage probes ... 144
6.4.4 Setup for current probes ... 150
6.4.5 Modular probes ... 152
6.4.6 Setup for optical isolated probe systems ... 155
6.4.7 Probe adapter R&S RT-Z2T ... 159
6.4.8 Probe info ... 160
6.4.9 Adjusting passive probes ... 162
**7 Trigger ... 163**
**7.1 Basics of triggering ... 163**
7.1.1 Trigger information ... 164
**7.2 Common trigger settings ... 165**
**7.3 Trigger sequence ... 167**
7.3.1 Sequence setup ... 168
**7.4 Trigger types ... 169**

7.4.1 Edge trigger....170
7.4.2 Edge trigger on external trigger source....171
7.4.3 Glitch trigger....172
7.4.4 Width trigger....174
7.4.5 Runt trigger....175
7.4.6 Window trigger....177
7.4.7 Timeout trigger....179
7.4.8 Interval trigger....180
7.4.9 Slew rate trigger....182
7.4.10 Setup & Hold....184
7.4.11 State trigger....186
7.4.12 Pattern trigger....187
7.4.13 Line trigger....192
**7.5 Trigger mode / holdoff....192**
**7.6 Hysteresis....194**
**7.7 Channel filter....196**
**7.8 Actions on trigger....197**
**7.9 Zone trigger....199**
7.9.1 Using trigger zones and areas....204
7.9.2 Zone trigger settings....206
7.9.3 Logic operator settings....208
7.9.4 History settings....210
**8 Signal configuration....211**
**8.1 Reference level setup....211**
**8.2 Gate setup....214**
**8.3 Timing reference....215**
8.3.1 General settings of the timing reference....215
8.3.2 Explicit clock signal....216
8.3.3 Clock data recovery by software CDR....217
**9 Waveform analysis....222**
**9.1 Zoom....222**
9.1.1 Zoom settings....222
9.1.2 Zooming for details....225

**9.2 Mathematics ... 226**
9.2.1 Creating math waveforms ... 227
9.2.2 General math waveforms settings ... 227
9.2.3 Operator settings ... 229
9.2.4 Filter settings ... 230
9.2.5 Equations ... 232
9.2.6 Scale settings for math waveforms ... 237
**9.3 History ... 239**
9.3.1 About history ... 239
9.3.2 History setup ... 240
9.3.3 Quick access history dialog ... 243
9.3.4 Using history ... 244
**9.4 Reference waveforms ... 245**
9.4.1 Working with reference waveforms ... 246
9.4.2 Settings for reference waveforms ... 247
**9.5 Waveform histograms ... 254**
9.5.1 Histogram settings ... 255
9.5.2 Creating histograms ... 258
9.5.3 Analysis of waveform histograms ... 259
**10 Measurements ... 260**
**10.1 Cursor measurements ... 260**
10.1.1 Cursors and results of cursor measurements ... 260
10.1.2 Using cursors ... 261
10.1.3 Settings for cursor measurements ... 263
**10.2 Automatic measurements ... 270**
10.2.1 Measurement setup in general ... 270
10.2.2 Reference levels for measurements ... 278
10.2.3 Gate settings for measurements ... 279
10.2.4 Amplitude/time measurements ... 280
10.2.5 Histogram measurements ... 289
10.2.6 Basic jitter analysis (option R&S MXO3-K12) ... 293
10.2.7 Protocol measurements (option R&S MXO3-K500) ... 298
10.2.8 Annotation for measurements ... 302

10.2.9 Measurement results....303
10.2.10 Analysis of measurement results....304

**11 Spectrum analysis....314**
**11.1 Fundamentals of spectrum analysis....314**
**11.2 Configuring spectrum waveforms....317**
**11.3 Spectrum setup....319**
**11.4 Spectrum scale....322**
**11.5 Spectrum gate....323**
**11.6 Spectrum peak list....324**
**11.7 Spectrum overlap....328**

**12 Applications....330**
**12.1 Mask testing....330**
12.1.1 About mask testing....330
12.1.2 Mask test setup: general settings....332
12.1.3 Mask definition: polygon....335
12.1.4 Mask definition: auto mask....337
12.1.5 Actions on mask test results....338
12.1.6 Working with masks....340
**12.2 Frequency response and impedance analyses (options R&S MXO3-K36/K337) 344**
12.2.1 Results and plot of analysis....345
12.2.2 Using frequency response and impedance analysis....348
12.2.3 Dependencies of the measurement setup....349
12.2.4 Settings for frequency response and impedance analysis....350
12.2.5 Saving analysis results....362
12.2.6 Reference waveforms of FRA curves....363
12.2.7 Example of impedance analysis....367
**12.3 Power analysis (option R&S MXO3-K31)....369**
12.3.1 Power analysis initial settings....369
12.3.2 Power quality....373
12.3.3 Power harmonics....379
12.3.4 Switching loss....386
12.3.5 Turn on/off time....392
12.3.6 Power efficiency....397

12.3.7 Safe operating area (SOA)....404
12.3.8 Common functions for power measurements....410
**12.4 3-phase power analysis (option R&S MXO3-K333)....416**
12.4.1 3-phase power analysis initial settings....417
12.4.2 Common functions for 3-phase power measurements....419
12.4.3 3-phase power quality....426
12.4.4 3-phase power harmonics....433
12.4.5 3-phase power efficiency....439
**12.5 Digital voltmeter....443**
12.5.1 Using the digital voltmeter....444
12.5.2 Settings of the digital voltmeter....445
**12.6 XY-plot....446**
12.6.1 Settings for XY-plots....446
12.6.2 Displaying an XY-diagram....447
**13 Data and file management....448**
**13.1 Accessing, retrieving and handling data....448**
**13.2 Save/recall dialog....449**
**13.3 General and measurement settings: savesets....451**
13.3.1 Using savesets....451
13.3.2 Waveform generator saveset....453
**13.4 Waveform data....455**
13.4.1 Save waveforms: settings....455
13.4.2 Saving waveforms....460
13.4.3 Waveform export files....462
**13.5 Histogram data export....471**
13.5.1 Files with waveform histograms....472
13.5.2 Files with histograms on measurement results....473
13.5.3 Data conversion....473
13.5.4 Settings for histogram export....474
**13.6 Results....475**
13.6.1 Save results: settings....475
13.6.2 Saving results....477
**13.7 Sessions....478**

13.7.1 Save a session....478
13.7.2 Load a session....480
**13.8 Screenshots....481**
13.8.1 Screenshot settings....482
13.8.2 Configuring and saving screenshots....485
13.8.3 Image viewer....486
**13.9 File browser dialog....486**
**14 Protocol analysis....488**
**14.1 Basics of protocol analysis....489**
14.1.1 Setup - general settings....489
14.1.2 Thresholds....490
14.1.3 Display....492
14.1.4 Filter....496
14.1.5 Format....497
14.1.6 Trigger....504
14.1.7 Measure....508
14.1.8 Export....509
14.1.9 Performing protocol analysis....510
14.1.10 Bit pattern editor....512
**14.2 SPI bus (IC electronics, option R&S MXO3-K510)....514**
14.2.1 About the SPI protocol....514
14.2.2 SPI configuration....515
14.2.3 SPI filter....519
14.2.4 SPI format configuration....521
14.2.5 SPI hardware trigger....527
14.2.6 SPI software trigger....528
14.2.7 Measure....531
14.2.8 SPI decode results....531
**14.3 QUAD-SPI bus (IC electronics, option R&S MXO3-K510)....533**
14.3.1 About the QUAD-SPI protocol....534
14.3.2 QUAD-SPI configuration....534
14.3.3 QUAD SPI filter....537
14.3.4 QUAD-SPI format configuration....540

14.3.5 QUAD SPI software trigger ..... 544
14.3.6 Measure ..... 547
14.3.7 QUAD-SPI decode results ..... 547
**14.4 OCTO-SPI bus (IC electronics, option R&S MXO3-K510) ..... 549**
14.4.1 About the OCTO-SPI protocol ..... 550
14.4.2 OCTO-SPI configuration ..... 551
14.4.3 OCTO SPI filter ..... 555
14.4.4 OCTO-SPI format configuration ..... 557
14.4.5 OCTO SPI software trigger ..... 561
14.4.6 Measure ..... 563
14.4.7 OCTO-SPI decode results ..... 564
**14.5 I²C (IC electronics, option R&S MXO3-K510) ..... 565**
14.5.1 About the I²C protocol ..... 566
14.5.2 I²C configuration ..... 568
14.5.3 I²C filter ..... 572
14.5.4 I2C format configuration ..... 575
14.5.5 I²C hardware trigger ..... 580
14.5.6 I²C software trigger ..... 584
14.5.7 Measure ..... 587
14.5.8 I²C decode results ..... 588
**14.6 I3C (IC electronics, option R&S MXO3-K550) ..... 590**
14.6.1 About the I3C protocol ..... 590
14.6.2 I3C configuration ..... 592
14.6.3 I3C filter ..... 596
14.6.4 I3C format configuration ..... 599
14.6.5 I3C software trigger ..... 605
14.6.6 Measure ..... 607
14.6.7 I3C decode results ..... 607
**14.7 UART (IC electronics, option R&S MXO3-K510) ..... 610**
14.7.1 About the UART / RS-232 interface ..... 610
14.7.2 UART configuration ..... 611
14.7.3 UART filter ..... 615
14.7.4 UART format configuration ..... 617

14.7.5 UART hardware trigger ... 623
14.7.6 UART software trigger ... 624
14.7.7 Measure ... 626
14.7.8 UART decode results ... 627
14.7.9 Performing UART decoding ... 628
**14.8 NRZ clocked & unclocked (option R&S MXO3-K510) ... 631**
14.8.1 NRZ basics ... 631
14.8.2 NRZ configuration ... 631
14.8.3 NRZ filter ... 635
14.8.4 NRZ format configuration ... 639
14.8.5 NRZ software trigger ... 646
14.8.6 Measure ... 650
14.8.7 NRZ decode results ... 650
**14.9 Manchester (option R&S MXO3-K510) ... 653**
14.9.1 Manchester basics ... 653
14.9.2 Manchester configuration ... 655
14.9.3 Manchester filter ... 660
14.9.4 Manchester format configuration ... 663
14.9.5 Manchester software trigger ... 668
14.9.6 Measure ... 672
14.9.7 Manchester decode results ... 672
**14.10 CAN (automotive electronics, option R&S MXO3-K520) ... 674**
14.10.1 CAN configuration ... 674
14.10.2 CAN filter ... 680
14.10.3 CAN format configuration ... 683
14.10.4 CAN hardware trigger ... 688
14.10.5 CAN software trigger ... 697
14.10.6 Measure ... 701
14.10.7 CAN decode results ... 701
**14.11 LIN (automotive electronics, option R&S MXO3-K520) ... 704**
14.11.1 About the LIN protocol ... 704
14.11.2 LIN configuration ... 705
14.11.3 LIN filter ... 708

14.11.4 LIN format configuration....710
14.11.5 LIN hardware trigger....715
14.11.6 LIN software trigger....718
14.11.7 Measure....721
14.11.8 LIN decode results....721
**14.12 SENT (automotive electronics, option R&S MXO3-K520)....724**
14.12.1 About the SENT protocol....724
14.12.2 SENT configuration....728
14.12.3 SENT filter....732
14.12.4 SENT format configuration....736
14.12.5 SENT software trigger....741
14.12.6 Measure....745
14.12.7 SENT decode results....745
**14.13 ARINC 429 (aerospace electronics, option R&S MXO3-K530)....747**
14.13.1 About the ARINC 429....747
14.13.2 ARINC 429 configuration....748
14.13.3 ARINC 429 filter....752
14.13.4 ARINC 429 format configuration....754
14.13.5 ARINC 429 software trigger....760
14.13.6 Measure....762
14.13.7 ARINC 429 decode results....762
**14.14 MIL-1553 (aerospace electronics, option R&S MXO3-K530)....764**
14.14.1 About the MIL-1553 protocol....764
14.14.2 MIL-1553 configuration....767
14.14.3 MIL-1553 filter....769
14.14.4 MIL-1553 software trigger....772
14.14.5 Measure....775
14.14.6 MIL-1553 decode results....775
**14.15 SpaceWire (aerospace communications, option R&S MXO3-K530)....777**
14.15.1 SpaceWire basics....777
14.15.2 SpaceWire configuration....778
14.15.3 SpaceWire filter....781
14.15.4 SpaceWire format configuration....784

14.15.5 SpaceWire software trigger....790
14.15.6 Measure....792
14.15.7 SpaceWire decode results....792
**14.16 SPMI (mobile electronics, option R&S MXO3-K550)....794**
14.16.1 SPMI configuration....794
14.16.2 SPMI filter....797
14.16.3 SPMI format configuration....800
14.16.4 SPMI software trigger....806
14.16.5 Measure....810
14.16.6 SPMI decode results....810
**14.17 RFFE (mobile electronics, option R&S MXO3-K550)....812**
14.17.1 About the RFFE protocol....812
14.17.2 RFFE configuration....814
14.17.3 RFFE filter....816
14.17.4 RFFE format configuration....820
14.17.5 RFFE software trigger....826
14.17.6 Measure....830
14.17.7 RFFE decode results....830
**14.18 10BASE-T1S (automotive Ethernet, option R&S MXO3-K560)....832**
14.18.1 10BASE-T1S basics....832
14.18.2 10BASE-T1S configuration....833
14.18.3 10BASE-T1S filter....838
14.18.4 PPIFormat.PPIFrame.Name10BASE-T1S format configuration....840
14.18.5 10BASE-T1S software trigger....846
14.18.6 Measure....848
14.18.7 10BASE-T1S decode results....849
**14.19 USB 1.1 (option R&S MXO3-K570)....851**
14.19.1 USB protocol basics....851
14.19.2 USB 1.1 configuration....856
14.19.3 USB 1.1 filter....860
14.19.4 USB1.1 format configuration....863
14.19.5 USB 1.1 trigger....869
14.19.6 Measure....872

14.19.7 USB 1.1 decode results ... 872
**14.20 USB 2.0 (option R&S MXO3-K570) ... 874**
14.20.1 About the USB protocol ... 874
14.20.2 USB 2.0 configuration ... 875
14.20.3 USB 2.0 filter ... 878
14.20.4 USB 2.0 format configuration ... 881
14.20.5 USB 2.0 trigger ... 886
14.20.6 Measure ... 890
14.20.7 USB 2.0 decode results ... 890
**15 Logic analysis on digital channels ... 892**
**15.1 Logic configuration ... 892**
15.1.1 Setup ... 893
15.1.2 Bus ... 895
15.1.3 Label settings ... 896
15.1.4 Skew settings ... 897
**15.2 Display ... 898**
15.2.1 Logic bus - decode table ... 899
**16 Waveform generator (option R&S MXO3-B6) ... 900**
**16.1 Setup of the waveform generator ... 900**
16.1.1 General settings ... 900
16.1.2 Modulation settings ... 905
16.1.3 Sweep settings ... 907
16.1.4 Arbitrary waveforms ... 908
**16.2 Configuring the waveform generator ... 912**
16.2.1 Configuring a function waveform ... 912
16.2.2 Configuring a modulation waveform ... 912
16.2.3 Configuring a sine sweep waveform ... 914
16.2.4 Configuring an arbitrary waveform ... 914
**17 Network operation ... 915**
**17.1 Connecting the instrument to the network (LAN) ... 915**
17.1.1 Connecting the instrument to the network ... 915
17.1.2 Assigning the IP address ... 916

17.1.3 Using host names ... 916
**17.2 Web interface ... 917**
17.2.1 Settings on the MXO 3 ... 917
17.2.2 Web browser ... 917
**17.3 Remote operation with VNC client ... 923**
**17.4 Remote control ... 923**
17.4.1 Remote control interfaces and protocols ... 923
17.4.2 Starting and stopping remote control ... 926
**17.5 Remote control - status reporting system ... 927**
17.5.1 Hierarchy of status registers ... 927
17.5.2 Structure of a SCPI status register ... 929
17.5.3 Contents of the status registers ... 930
17.5.4 Error queue ... 935
17.5.5 Reset values of the status reporting system ... 935
**18 Remote control commands ... 937**
**18.1 Conventions used in remote command description ... 937**
**18.2 Finding the appropriate command ... 938**
**18.3 Error messages on remote commands ... 938**
**18.4 Frequently used parameters and suffixes ... 939**
18.4.1 Waveform parameter ... 939
18.4.2 Slope parameter ... 940
18.4.3 Polarity parameter ... 940
**18.5 Programming examples ... 941**
18.5.1 SmartGrid layout with zoom ... 941
18.5.2 Creating a mask ... 945
18.5.3 Saving screenshots to file ... 945
18.5.4 Data transfer in roll mode ... 946
**18.6 Common commands ... 946**
**18.7 General remote settings ... 950**
**18.8 Instrument setup ... 953**
18.8.1 System ... 953
18.8.2 Network drive mapping ... 955
18.8.3 SmartGrid ... 957

18.8.4 Toolbar ... 963
18.8.5 Annotations ... 964
18.8.6 Appearance ... 976
18.8.7 Display ... 981
18.8.8 Save/Recall ... 984
18.8.9 Preset ... 986
18.8.10 Maintenance ... 987
**18.9 Acquisition and waveform setup ... 989**
18.9.1 Starting and stopping acquisition ... 989
18.9.2 Horizontal setup ... 990
18.9.3 Roll mode ... 992
18.9.4 Acquisition setup ... 993
18.9.5 Vertical setup ... 999
18.9.6 Probes ... 1004
18.9.7 Fast segmentation ... 1028
18.9.8 High definition mode ... 1029
18.9.9 Waveform data export ... 1030
**18.10 Trigger ... 1030**
18.10.1 Common trigger settings ... 1031
18.10.2 Trigger sequence ... 1033
18.10.3 Edge trigger ... 1035
18.10.4 Glitch trigger ... 1037
18.10.5 Width trigger ... 1038
18.10.6 Runt trigger ... 1040
18.10.7 Window trigger ... 1042
18.10.8 Timeout trigger ... 1045
18.10.9 Interval trigger ... 1046
18.10.10 Slew rate trigger ... 1048
18.10.11 Setup & Hold ... 1050
18.10.12 State trigger ... 1052
18.10.13 Pattern trigger ... 1054
18.10.14 Trigger mode, holdoff ... 1056
18.10.15 Hysteresis ... 1059

18.10.16 Channel filter....1061
18.10.17 Actions on trigger....1062
18.10.18 Zone trigger....1065
**18.11 Signal configuration....1072**
18.11.1 Reference levels....1072
18.11.2 Gate....1078
18.11.3 Timing reference....1081
**18.12 Waveform analysis....1089**
18.12.1 Zoom....1089
18.12.2 Mathematics....1098
18.12.3 History....1102
18.12.4 Reference waveforms....1105
18.12.5 Waveform histograms....1114
**18.13 Data management....1119**
18.13.1 Instrument settings....1119
18.13.2 Saveset....1126
18.13.3 Generator saveset....1126
18.13.4 Waveform data: export to file....1127
18.13.5 Histogram data: export to file....1132
18.13.6 Sessions....1133
18.13.7 Results....1135
18.13.8 Screenshots....1137
18.13.9 Data transfer to remote computer....1141
**18.14 Automatic measurements....1151**
18.14.1 General settings....1151
18.14.2 Settings for amplitude/time measurements....1156
18.14.3 Settings for histogram measurements....1160
18.14.4 Settings for jitter measurements....1161
18.14.5 Settings for protocol measurements....1164
18.14.6 Results....1166
18.14.7 Statistics....1169
18.14.8 Tracks....1170
18.14.9 Histograms on measurement results....1173

**18.15 Cursor measurements ..... 1175**
18.15.1 Cursor setup ..... 1176
18.15.2 Cursor results ..... 1186
18.15.3 Peak search using cursors ..... 1187
18.15.4 Cursor appearance ..... 1189
**18.16 Spectrum analysis ..... 1190**
18.16.1 Spectrum setup ..... 1190
18.16.2 Spectrum gate ..... 1200
18.16.3 Peak list ..... 1201
18.16.4 Spectrum with frames and overlap ..... 1206
18.16.5 Data export of spectrum waveforms ..... 1208
**18.17 Applications ..... 1208**
18.17.1 Mask testing ..... 1208
18.17.2 Frequency response and impedance analyses (options R&S MXO3-K36/K337) ..... 1222
18.17.3 Power analysis (option R&S MXO3-K31) ..... 1247
18.17.4 3-phase power analysis (option R&S MXO3-K333) ..... 1313
18.17.5 Digital voltmeter ..... 1357
18.17.6 XY-plot ..... 1360
**18.18 Protocols ..... 1361**
18.18.1 Configuration settings for all serial protocols ..... 1361
18.18.2 SPI (option R&S MXO3-K510) ..... 1367
18.18.3 QUAD-SPI (option R&S MXO3-K510) ..... 1401
18.18.4 OCTO-SPI (option R&S MXO3-K510) ..... 1433
18.18.5 I²C (option R&S MXO3-K510) ..... 1464
18.18.6 I3C (option R&S MXO3-K550) ..... 1500
18.18.7 UART / RS-232 (option R&S MXO3- K510) ..... 1533
18.18.8 NRZ Clocked (option R&S MXO3-K510) ..... 1566
18.18.9 NRZ Unclocked (option R&S MXO3-K510) ..... 1595
18.18.10 Manchester (option R&S MXO3-K510) ..... 1625
18.18.11 CAN (option R&S MXO3-K520) ..... 1655
18.18.12 LIN (option R&S MXO3-K520) ..... 1703
18.18.13 SENT (option R&S MXO3-K520) ..... 1736
18.18.14 ARINC 429 (option R&S MXO3-K530) ..... 1772

18.18.15 MIL-1553 (option R&S MXO3-K530)....1800
18.18.16 SpaceWire (option R&S MXO3-K530)....1829
18.18.17 SPMI (option R&S MXO3-K550)....1856
18.18.18 RFFE (option R&S MXO3-K550)....1887
18.18.19 10BASE-T1S (option R&S MXO3-K560)....1920
18.18.20 USB 1.1 (option R&S MXO3-K570)....1951
18.18.21 USB 2.0 (option R&S MXO3-K570)....1982
**18.19 Logic analysis on digital channels....2014**
18.19.1 Digital channels....2014
18.19.2 Logic configuration....2018
18.19.3 Digital resolution....2026
18.19.4 Export of digital data....2027
**18.20 Waveform generator (option R&S MXO3-B6)....2027**
18.20.1 Waveform generator setup....2027
**18.21 Status reporting....2045**
18.21.1 General commands....2046
18.21.2 STATus:OPERation register....2046
18.21.3 STATus:QUEStionable registers....2048
18.21.4 Reading out the CONDition part....2050
18.21.5 Reading out the EVENt part....2050
18.21.6 Controlling the ENABle part....2051
18.21.7 Controlling the negative transition part....2052
18.21.8 Controlling the positive transition part....2052
18.21.9 Other status commands....2053
18.21.10 Programming tips and examples....2054
**18.22 Deprecated commands....2056**
18.22.1 Base instrument....2056
18.22.2 Applications....2057
18.22.3 Serial protocols....2058
**19 Maintenance and support....2059**
**19.1 Cleaning....2059**
**19.2 Changing fuses....2059**
**19.3 Contacting customer support....2060**

19.4 Information for technical support....2060
19.5 Data security....2061
19.6 Transporting....2061
19.7 Storage....2061
19.8 Disposal....2061
List of commands....2063

# 1 Safety and regulatory information

The product documentation helps you to use the product safely and efficiently.

**Where do I find safety information?**

Safety information is part of the product documentation. It warns you of potential dangers and gives instructions on how to prevent personal injury or damage caused by dangerous situations. Safety information is provided as follows:

- In Section 1.1, "Safety instructions", on page 23. The same information is provided in many languages in printed format. The printed "Safety Instructions" for "Oscilloscopes and Accessories" are delivered with the instrument and the probes.
- Throughout the documentation, safety instructions are provided when you need to take care during setup or operation.

**Intended use**

The MXO 3 product is designed for measurements on circuits that are only indirectly connected to the mains or not connected at all. It is not rated for any measurement category.

The product is intended for the development, production and verification of electronic components and devices in industrial, administrative, and laboratory environments. Use the product only for its designated purpose. Observe the operating conditions and performance limits stated in the specifications document.

**Target audience**

The target audience of this document includes developers and technicians, administrators and maintenance personnel using oscilloscopes and probes. The required skills and experience of the users depend on the test setup and application of the product.

## 1.1 Safety instructions

Products from the Rohde & Schwarz group of companies are manufactured according to the highest technical standards. To use the products safely, follow the instructions provided here and in the product documentation. Keep the product documentation nearby and offer it to other users.

Use the product only for its intended use and within its performance limits. Intended use and limits are described in the product documentation such as the specifications document, manuals and the printed "Safety Instructions" document. If you are unsure about the appropriate use, contact Rohde & Schwarz customer support.

Using the product requires skilled persons or specially trained personnel. These users also need sound knowledge of at least one of the languages in which the user interfaces and the product documentation are available.

Reconfigure or adjust the product only as described in the product documentation or the specifications document. Any other modifications can affect safety and are not permitted.

Never open the casing of the product. Only service personnel authorized by Rohde & Schwarz are allowed to repair the product. If any part of the product is damaged or broken, stop using the product. Contact Rohde & Schwarz customer support at https://www.rohde-schwarz.com/support.

In these safety instructions, the term "product" covers instruments (oscilloscopes), probes and their accessories.

### Lifting and carrying the instrument

Check the specifications document for the maximum weight of the instrument. A single person can only carry a maximum of 18 kg safely depending on age, gender and physical condition. If your instrument is heavier than 18 kg, do not move or carry it by yourself.

Use the instrument handles to move or carry the instrument. Do not use the mounted accessories instead of the handles. Accessories are not designed to carry the weight of the instrument.

To move the instrument safely, you can use lifting or transporting equipment such as lift trucks and forklifts. Follow the instructions provided by the equipment manufacturer.

### Choosing the operating site

Only use the product indoors. The product casing is not waterproof. Water that enters can electrically connect the casing with live parts, which can lead to electric shock, serious personal injury or death if you touch the casing.

If Rohde & Schwarz provides accessories designed for outdoor use of your product, e.g. a protective cover, you can use the product outdoors.

You can operate the product up to an altitude of 2000 m above sea level. If a higher altitude is permissible, the value is provided in the specifications document. The product is suitable for pollution degree 2 environments where nonconductive contamination can occur. For more information on environmental conditions such as ambient temperature and humidity, see the specifications document.

### Setting up the product

Always place the product on a stable, flat and level surface with the bottom of the product facing down. If the product is designed for different positions, secure the product so that it cannot fall over.

If the product has foldable feet, always fold the feet completely in or out to ensure stability. The feet can collapse if they are not folded out completely or if the product is moved without lifting it. The foldable feet are designed to carry the weight of the product, but not an extra load.

If stacking is possible, keep in mind that a stack of products can fall over and cause injury.

If you mount products in a rack, ensure that the rack has sufficient load capacity and stability. Observe the specifications of the rack manufacturer. Always install the products from the bottom shelf to the top shelf so that the rack stands securely. Secure the product so that it cannot fall off the rack.

### Connecting the product

Before connecting the interfaces or measuring inputs of the product to other products or electrical circuits, make sure that the other products or electrical circuits provide special protection against electric shock. This protection principle is referred to as SELV (safety extra-low voltage) and is based on a low voltage level and increased insulation. Exceptions are indicated by a measurement category on the product and given in the specifications document.

### Connecting to power and grounding

The mains power supply input of the instrument complies with overvoltage category II. Connect the product to a fixed installation used to supply energy-consuming equipment such as household appliances and similar loads. Keep in mind that electrically powered products have risks, such as electric shock, fire, personal injury or even death.

Take the following measures for your safety:

- Do not use an isolating transformer to connect the instrument to the mains power supply.
- Before switching on the product, ensure that the voltage and frequency indicated on the product match the available power source. If the power adapter does not adjust automatically, set the correct value and check the rating of the fuse.
- If a product has an exchangeable fuse, its type and characteristics are indicated next to the fuse holder. Before changing the fuse, switch off the product and disconnect it from the power source. How to change the fuse is described in the product documentation.
- Only use the power cable delivered with the product. It complies with country-specific safety requirements. Only insert the plug into an outlet with protective conductor terminal.
- Only use intact cables and route them carefully so that they cannot be damaged. Check the power cables regularly to ensure that they are undamaged. Also ensure that nobody can trip over loose cables.
- If you connect the product to an external power supply, use the one delivered with the product or recommended in the product documentation. The external power supply must conform to the country-specific regulations.
- Ensure that you can disconnect the product from the power source at any time. Pull the power plug to disconnect the product. The power plug must be easily accessible. If the product is integrated into a system that does not meet these requirements, provide an easily accessible circuit breaker at the system level.
- Replace parts that are relevant to safety only by original parts, e.g. power cables or fuses.

### Performing measurements

Take the following measures for your safety:

- To ascertain a voltage-free state, use an appropriate voltage tester. Any measurement setup including an oscilloscope is not suitable for this purpose.
- The maximum input voltage on channel inputs and the external trigger input must not exceed the value specified in the specifications document.
- Observe all voltage and current ratings of the instrument, the probes, and the accessories. Exceeding the allowed voltages can lead to an electric shock.
  Limits and ratings are marked on the products and listed in the specifications documents.
  Consider that the rated voltage depends on the frequency. The voltage limitation curves or values are provided in the specifications document.
- Never cause any short circuits when measuring sources with high output currents.
- Use only probes and accessories that comply with the measurement category (CAT) of your measurement task. If the product is rated for any measurement category, the permitted category is indicated on the product and in the specifications document. If you use other than Rohde & Schwarz accessories, make sure that they are suitable for the instrument and the measurement task.
- Set the correct attenuation factor on the instrument according to the probe being used. Otherwise, the measurement results do not reflect the actual voltage level, and you might misjudge the actual risk.
- When working with high voltages and current probes, observe the additional operating conditions specified in these safety instructions.
- The probe pins are extremely pointed and can easily penetrate clothes and the skin. Handle the probe pins with great care. To exchange a probe pin, use tweezers or pliers to avoid injuries. When transporting the accessories, always use the box supplied with the probe.
- Prevent the probe from receiving mechanical shock. Avoid putting excessive strain on the probe cable or exposing it to sharp bends. Touching a broken cable during measurements can cause injuries.
- Set up all probe connections to the instrument before applying power.

### Working with hazardous voltages

Voltages higher than 30 V RMS, or 42 V peak, or 60 V DC are regarded as hazardous contact voltages. Direct contact with them can cause serious injuries.

Make sure that only electrically skilled persons use the products for measurements on hazardous contact voltages. These working conditions require special education and experience to perceive risks and to avoid hazards which electricity can create.

When working with hazardous contact voltages, use protective measures to preclude direct contact with the measurement setup:

- Do not touch exposed connections and components when power is applied.
- Switch off the test circuit while connecting and disconnecting probe leads.
- Use only insulated voltage probes, test leads and adapters.
- Make sure that the input leads fulfill the safety requirements for your measurement.

The delivered input leads might have a jacket wear indicator that indicates a worn jacket by different jacket color. In this case, do not use the input lead. Replace it with a new one.

- When connecting to the DUT, keep your fingers behind finger guard. Remove jewelry, watches, and other metallic objects. Only use 4 mm safety banana plugs.

### Using optical isolated probe systems

Optical isolated probes are classified as class 1 laser product. Class 1 lasers are safe under all conditions of normal use. The product fully contains the beam of a higher-class laser. Take the following measures for your safety:

- To avoid exposure to the laser beam, never remove any covers from the probe head or probe receiver, and never disassemble the product.
- Check the product before using to ensure that it is undamaged. If you have dropped the product or exposed it to excessive mechanical stress, always check the product. Do not use the product if any component is damaged.
- Send the product to service regularly.

Optical isolated probes measure differential voltages. The test circuit can have a different potential. The common mode voltage between the potentials of the DUT and the oscilloscope can be much higher than the differential voltage and is not measured by the probe. To avoid the risk of electrical shock, use protective measures:

- Always de-energize the test circuit before installing or removing the tip module.
- Never exceed the differential voltage rating.
- When measuring on a test circuit with high common mode voltage, use a tripod to hold the probe head.
- Keep the probe head and the tip cable away from circuits with different potential to prevent arc flash caused by a different potential.

### Working with current probes

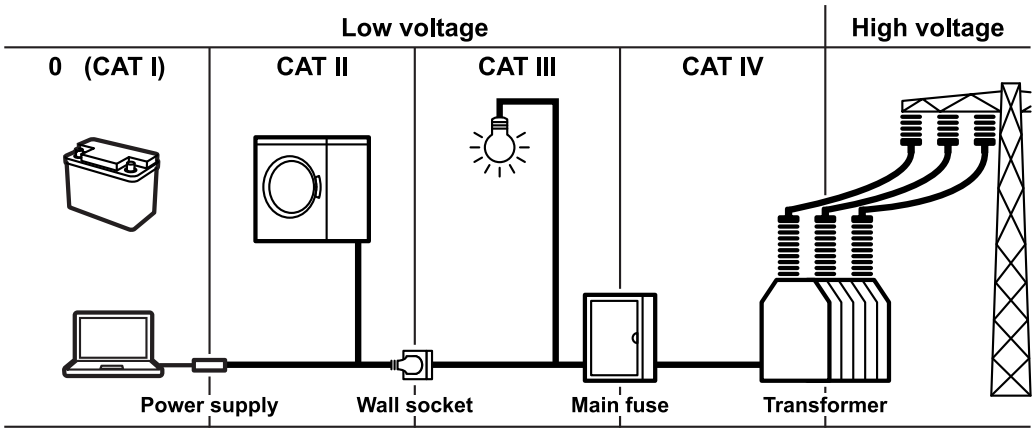
When working with current probes, you can measure high-frequency currents or currents that contain high-frequency components.

- Switch off the test circuit while connecting the probe.
- Do not attach the clamp to bare unisolated conductors. To avoid injury from a short circuit, measure at a location on an insulated wire where the insulation is sufficient for the circuit voltage.
- Connect the probe only to the secondary side of a breaker. With this measure, you can avoid injury if a short circuit occurs.
- The following effects can cause burns and fire or damage to the measurement site:
  - Eddy current loss can cause heating of the sensor head.
  - Dielectric heating can cause heating of cord insulation and other materials.
- When measuring current that includes a high-frequency component, consider the derating characteristics of the probe. Do not measure any current that exceeds the rated current.
- Using the probes with high frequencies or strong magnetic fields may cause the device to become abnormally hot, resulting in fire, equipment damage, or burns.

### Measurement categories

IEC 61010-2-030 defines measurement categories that rate instruments on their ability to resist short transient overvoltages that occur in addition to the working voltage. Use the measurement setup only in electrical environments for which they are rated.

- 0 - Instruments without rated measurement category
  For measurements performed on circuits not directly connected to mains, for example, electronics, circuits powered by batteries, and specially protected secondary circuits. This measurement category is also known as CAT I.
- CAT II:
  For measurements performed on circuits directly connected to the low-voltage installation by a standard socket outlet, for example, household appliances and portable tools.
- CAT III:
  For measurements performed in the building installation, such as junction boxes, circuit breakers, distribution boards, and equipment with permanent connection to the fixed installation.
- CAT IV:
  For measurements performed at the source of the low-voltage installation, such as electricity meters and primary overcurrent protection devices.




### Cleaning the product

Use a dry, lint-free cloth to clean the product. When cleaning, keep in mind that the casing is not waterproof. Do not use liquid cleaning agents.

### Meaning of safety labels

Safety labels on the product and its accessories warn against potential hazards.

| | |
|---|---|
|  | Potential hazard<br>Read the product documentation to avoid personal injury or product damage. |

| | |
|---|---|
|  | Electrical hazard<br>Indicates live parts. Risk of electric shock, fire, personal injury or even death. |

| | |
|---|---|
|  | Protective conductor terminal<br>Connect this terminal to a grounded external conductor or to protective ground. This connection protects you against electric shock if an electric problem occurs. |

## 1.2 Labels on the product

Labels on the casing inform about:

- Personal safety, see "Meaning of safety labels" on page 28
- Product and environment safety, see Table 1-1
- Identification of the product

***Table 1-1: Labels regarding product and environment safety***

| | |
|---|---|
|  | Chassis grounding terminal |
|  | Take care when handling electrostatic sensitive devices. |
|  | Labeling in line with EN 50419 for disposal of electrical and electronic equipment after the product has come to the end of its life.<br>For more information, see "Disposing of electrical and electronic equipment" on page 2062. |

## 1.3 Warning messages in the documentation

A warning message points out a risk or danger that you need to be aware of. The signal word indicates the severity of the safety hazard and how likely it will occur if you do not follow the safety precautions.

**WARNING**

Potentially hazardous situation. Could result in death or serious injury if not avoided.

**CAUTION**

Potentially hazardous situation. Could result in minor or moderate injury if not avoided.

**NOTICE**

Potential risks of damage. Could result in damage to the supported product or to other property.

## 1.4 CE declaration of conformity

The CE declaration of conformity of the instrument is delivered with the product. Keep the document for further reference.

The current version of this CE declaration of conformity is available at:

www.rohde-schwarz.com/company-documents/mxo3/

## 1.5 Where to find key documents on Rohde & Schwarz

Certificates issued to Rohde & Schwarz that are relevant for your country are provided at www.rohde-schwarz.com/key-documents, e.g. concerning:

- Quality management
- Environmental management
- Information security management
- Accreditations

## 1.6 Korea certification class A



이 기기는 업무용(A급) 전자파 적합기기로서 판매자 또는 사용자는 이 점을 주의하시기 바라며, 가정외의 지역에서 사용하는 것을 목적으로 합니다.

# 2 Preface

## 2.1 Key features

The MXO 3 series oscilloscope is a next generation oscilloscope with 4 or 8 channels, evolved for more challenges.

Using the MXO 3, you can:

- Acquire, process and display signals with unrivaled speed.
- Detect signal details, faults and anomalies instantly.
- Increase your statistical confidence.

The MXO 3 series oscilloscopes utilize advanced technologies:

- Fastest update rate of up to 4.5 million waveforms/s thanks to MXO-EP ASIC
- 12 bit resolution at all sample rates, and 18 bit vertical resolution in HD mode
- Deep standard memory: 125 Mpoints per channel
- High-precision digital trigger with best-in-class trigger jitter, also available in HD mode
- Nearly zero blind time: up to 99% real-time signal activity capture
- Lowest measurement noise
- Eco-friendly design: low power consumption, automatic power-up capability, scarcely audible

Features and applications help achieve fast and accurate results, for example:

- Segmented memory & history mode to analyze previous acquisitions
- Superior spectrum analysis with 50 000 FFT/s, which allows capture of spurious spectrum events
- Integrated arbitrary waveform generator
- Logic analysis with 16 digital channels
- Dual-path serial protocol analysis
- Frequency response analysis (Bode plots)
- Large high-resolution touchscreen, R&S SmartGrid, configurable toolbar, intuitive navigation
- Wide selection of compatible probes

## 2.2 Documentation overview

This section provides an overview of the MXO 3 user documentation.

### 2.2.1 Manuals and instrument help

You find the manuals on the product page at:

www.rohde-schwarz.com/manual/mxo3

**Getting started manual**

Introduces the MXO 3 and describes how to set up the product. A printed English version is included in the delivery.

**User manual and help**

Contains the description of all instrument modes and functions. It also provides an introduction to remote control, a complete description of the remote control commands with programming examples, and information on maintenance and instrument interfaces. Includes the contents of the getting started manual.

The contents of the user manual are available as help in the MXO 3. The help offers quick, context-sensitive access to the complete information on the instrument and its firmware.

The user manual is available for download or immediate display on the internet.

**Safety instructions**

Provides safety information in many languages. The printed document is delivered with the product.

**Instrument security procedures manual**

Deals with security issues when working with the MXO 3 in secure areas. It is available for download on the internet.

**Service manual**

Describes the performance test for checking compliance with rated specifications, firmware update, troubleshooting, adjustments, installing options and maintenance. The service manual is available for registered users on the global Rohde & Schwarz information system (GLORIS, https://gloris.rohde-schwarz.com).

### 2.2.2 Specifications and product brochure

The specifications document, also known as the data sheet, contains the technical specifications of the MXO 3. It also lists the firmware applications and their order numbers, and optional accessories. The product brochure provides an overview of the instrument and deals with the specific characteristics.

www.rohde-schwarz.com/brochure-datasheet/mxo3

### 2.2.3 Calibration certificate

The document is available on https://gloris.rohde-schwarz.com/calcert. You need the device ID of your instrument, which you can find on a label on the rear panel.

### 2.2.4 Release notes, open source acknowledgment

The release notes list new features, improvements and known issues of the current firmware version, and describe the firmware installation. The open source acknowledgment document provides verbatim license texts of the used open source software. It can also be read directly on the instrument.

www.rohde-schwarz.com/firmware/mxo3

### 2.2.5 Application notes, application cards, videos

Various documents and videos deal with special applications or background information on particular topics.

R&S®MXO3 - Mediacenter

Find various videos on Rohde & Schwarz products and test and measurement topics on YouTube: https://www.youtube.com/@Rohde-Schwarz

## 2.3 Options described in this document

In addition to the base unit, the following options are described in this documentation:

| Type | Designation | Order No. |
|---|---|---|
| R&S MXO3-B1 | MSO | 1335.2073.02 |
| R&S MXO3-B6 | Waveform and pattern generator | 1335.2850.02 |
| R&S MXO3-K12 | Basic jitter analysis | 1335.3170.02 |
| R&S MXO3-K31 | Power analysis | 1335.2880.02 |
| R&S MXO3-K36 | Frequency response analysis | 1335.2896.02 |
| R&S MXO3-K333 | 3-phase power analysis | 1335.3334.02 |
| R&S MXO3-K337 | Impedance analysis | |
| R&S MXO3-K500 | Bus analysis | 1335.2980.02 |
| R&S MXO3-K510 | Triggering and decoding low speed serial buses:<br>I2C, SPI, UART/RS-232/RS-422/RS-485, QUAD-SPI, NRZ clocked, NRZ unclocked, Manchester | 1335.2867.02 |
| R&S MXO3-K520 | Triggering and decoding automotive protocols:<br>CAN, CAN FD, CAN XL, LIN, SENT | 1335.2873.02 |

| Type | Designation | Order No. |
|---|---|---|
| R&S MXO3-K530 | Decoding aerospace protocols:<br>ARINC 429, MIL-STD-1553, SpaceWire | 1335.2996.02 |
| R&S MXO3-K550 | Decoding MIPI low speed protocols:<br>I3C, RFFE, SPMI | 1335.5214.02 |
| R&S MXO3-K560 | Decoding automotive Ethernet protocols:<br>10BASE-T1S | 1335.5943.02 |
| R&S MXO3-K570 | Decoding automotive USB protocols:<br>USB 1.1, USB 2.0 | 1335.3186.02 |

# 3 Getting Started

## 3.1 Preparing for use

Here, you can find basic information about setting up the instrument for the first time or when changing the operating site.

### 3.1.1 Lifting and carrying

See: "Lifting and carrying the instrument" on page 24.

### 3.1.2 Unpacking and checking

1. Unpack the product carefully.
2. Retain the original packing material. Use it when transporting or shipping the product later.
3. Using the delivery notes, check the equipment for completeness.
4. Check the equipment for damage.

   If the delivery is incomplete or equipment is damaged, contact Rohde & Schwarz.

### 3.1.3 Choosing the operating site

Specific operating conditions ensure proper operation and avoid damage to the product and connected devices. For information on environmental conditions such as ambient temperature and humidity, see the specifications document.

For safety information, see "Choosing the operating site" on page 24.

**Electromagnetic compatibility classes**

The electromagnetic compatibility (EMC) class indicates where you can operate the product. The EMC class of the product is given in the specifications document.

- Class B equipment is suitable for use in:
  - Residential environments
  - Environments that are directly connected to a low-voltage supply network that supplies residential buildings
- Class A equipment is intended for use in industrial environments. It can cause radio disturbances in residential environments due to possible conducted and radiated disturbances. It is therefore not suitable for class B environments.

If class A equipment causes radio disturbances, take appropriate measures to eliminate them.

### 3.1.4 Setting up the product

When setting up the instrument, follow the safety instructions:

- "Setting up the product" on page 24
- "Intended use" on page 23

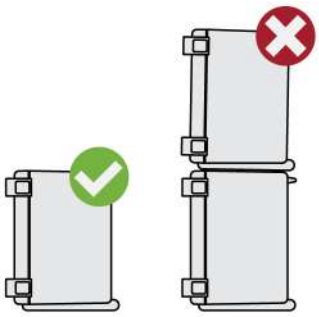
#### 3.1.4.1 Placing the product on a bench top

For standalone operation, place the instrument on a horizontal bench with even, flat surface. The instrument can be used in horizontal position, standing on its feet, or with the support feet on the bottom extended.

**To place the product on a bench top**

1. Place the product on a stable, flat and level surface.
2. **CAUTION!** The top surface of the product is too small for stacking. If you stack another product on top of the product, the stack can fall over and cause injury.

   If you want to save space, mount several products in a rack.



3. **CAUTION!** Foldable feet can collapse. For safety information, see "Setting up the product" on page 24.

   Always fold the feet completely in or out. With folded-out feet, do not place anything on top or underneath.
4. **NOTICE!** Overheating can damage the product.

   Prevent overheating as follows:

   - Keep a minimum distance of 10 cm between the fan openings of the product and any object in the vicinity to provide sufficient airflow and ventilation.
   - Do not place the product next to heat-generating equipment such as radiators or other products.

#### 3.1.4.2 Mounting the product on a monitor arm

You can mount the oscilloscope to a monitor arm with VESA mount.

► **NOTICE!** Only use a VESA mount compatible with standard 100 mm × 100 mm pattern.

Mount the VESA mount of the monitor arm to the VESA mounting holes on the rear panel of the oscilloscope.

#### 3.1.4.3 Mounting the product in a rack

**To prepare the rack**

1. Observe the requirements and instructions in "Setting up the product" on page 24.
2. **NOTICE!** Insufficient airflow can cause overheating and damage the product.
   Design and implement an efficient ventilation concept for the rack.

**To mount the oscilloscope in a rack**

1. Use a rackmount kit to prepare the instrument for rack mounting.
   a) Order the rackmount kit designed for your oscilloscope. For the order number, see specifications document.
   b) Mount the rackmount kit. Follow the assembly instructions provided with the rackmount kit.
2. Lift the product to shelf height. If the rack is high, use a safe climbing aid when placing on upper shelves.
3. Grip the product by the handles. Slide the product onto the shelf until the rack brackets fit closely to the rack.
4. Tighten all screws at the rack brackets with a tightening torque of 1.2 Nm to secure the product at the rack.

**To unmount the product from a rack**

1. Loosen the screws at the rack brackets.
2. Remove the product from the rack.
3. If placing the product on a bench top again, unmount the rackmount kit from the product. Follow the instructions provided with the rackmount kit.

### 3.1.5 Considerations for test setup

Observe safety instructions, see "Performing measurements" on page 26.

**Cable selection and electromagnetic interference (EMI)**

Electromagnetic interference (EMI) can affect the measurement results.

To suppress electromagnetic radiation during operation:

- Use high-quality shielded cables, for example, double-shielded RF and LAN cables.
- Always terminate open cable ends.
- Ensure that connected external devices comply with EMC regulations.

**Measuring accessories**

Use only probes and measuring accessories that comply with IEC 61010-031.

**Signal input and output levels**

Information on signal levels is provided in the specifications document. Keep the signal levels within the specified ranges to avoid damage to the product and connected devices.

**Preventing electrostatic discharge (ESD)**

Electrostatic discharge is most likely to occur when you connect or disconnect a DUT.

► **NOTICE!** Electrostatic discharge can damage the electronic components of the product and the device under test (DUT).

Ground yourself to prevent electrostatic discharge damage:

a) Use a wrist strap and cord to connect yourself to ground.
b) Use a conductive floor mat and heel strap combination.

During operation, if the firmware observes a serious unexpected disturbance (e.g. due to ESD), it resets some hardware components and initiates a new alignment to ensure proper instrument functioning. Then it restores the user settings to the state before the disturbance.

### 3.1.6 Connecting to power

For safety information, see "Connecting to power and grounding" on page 25.

The MXO 3 can be used with different AC power voltages and adapts itself automatically to it.

The nominal ranges are indicated on a label near the power connector on the instrument and in the specifications document.

1. Plug the AC power cable into the AC power connector on the rear panel of the product. Only use the AC power cable that is delivered with the product.
2. Plug the AC power cable into a power outlet with ground contact.

### 3.1.7 Connecting to LAN

For remote operation of the instrument using a computer, you need a local area network (LAN) connection.

**Network environment**

Before connecting the product to a LAN, consider the following:

- Install the latest firmware to reduce security risks.
- For internet or remote access, use secured connections if applicable.
- Ensure that the network settings comply with the security policies of your company. Contact your local system administrator or IT department before connecting your product to your company LAN.
- When connected to the LAN, the product may potentially be accessed from the internet, which may be a security risk. For example, attackers might misuse or damage the product.

**To connect a LAN cable**

► Connect an RJ-45 cable to the LAN connector on the rear panel, and to the LAN.

For supported LAN interfaces, refer to the specifications document.

By default, the MXO 3 is configured to use DHCP. Configuration of a static IP address is not required.

### 3.1.8 Connecting external devices

#### 3.1.8.1 Connecting USB devices

You can use USB connectors to connect various devices:

- Connect a keyboard and a mouse for operation and data entry, alternatively or in addition to the touchscreen functionality.
- Connect USB flash drives to save screenshots, and to transfer files to and from the instrument without a network connection.

► Connect the USB device to any of the USB connectors.

You can connect or disconnect USB devices during operation of the instrument.

#### 3.1.8.2 Connecting external monitors

Using the HDMI connector on the rear panel, you can connect an external monitor or projector to the oscilloscope.

1. **NOTICE!** Missing ground contact can damage the oscilloscope.

Ensure that the monitor and the oscilloscope are connected to a ground contact.
See also: Section 3.1.6, "Connecting to power", on page 38.

2. **NOTICE!** Use only standard-compliant monitor cables.
   Connect the monitor or projector to a display connector.

   The oscilloscope recognizes the external monitor. During the running session, you can disconnect the monitor and connect it again.

If the monitor provides touch functionality, an additional connection can be required, e.g. a USB connection. Refer to the documentation of your monitor.

The oscilloscope outputs a Full HD image with 1920x1080 pixel. If the screen resolution of the monitor is set higher than the instrument's resolution, the application window uses a 1920x1080 area of the monitor display. For full-screen display, adjust the monitor's screen resolution.

### 3.1.9 Switching on or off

The instrument is switched on or off with the power switch and the [Power] key. The [Power] key is located in the bottom left corner of the front panel. The power switch is located at the rear panel of the instrument.

***Table 3-1: Overview of power states***

| Status | LED | Power switch |
|---|---|---|
| Off | (unlighted) | [0] (off) |
| Standby | yellow | [I] (on) |
| Ready | green | [I] (on) |

**To switch on the product**

The product is off but connected to power.

1. Set the switch on the power supply to position [I].
   The power key on the front panel lights up.
2. Press the [Power] key on the front panel.
   The instrument performs a system check, boots the operating system, and then starts the MXO 3 firmware.
   The [Power] key turns green and the illuminated keys on the front panel light up.
   If the previous session was terminated regularly, the oscilloscope uses the last settings.

Before you start measurements, be sure to comply with the warm-up phase specified in the specifications document.

**To shut down the product**

The product is in the ready state.

► Press the [Power] key.

All current settings are saved, and the software shuts down. The [Power] key turns yellow. The standby power supplies only the power switch circuits.

The "Power" icon in the "Menu" shuts down only the firmware application. To shut down the instrument completely, use the [Power] key.

Remote command for firmware shutdown: `SYSTem:EXIT` on page 954.

**To disconnect from power**

The product is in the standby state.

1. **NOTICE!** Risk of data loss. If you disconnect the product from power when it is in the ready state, you can lose settings and data. Shut it down first.

   Set the switch on the power supply to position [0].

   The LED of the [Power] key is switched off.

2. Disconnect the product from the power source.

## 3.2 Instrument tour

This section describes the front, rear and side view of the instrument including all function keys and connectors.

### 3.2.1 Front view

This section provides an overview of the MXO 3 front panel.

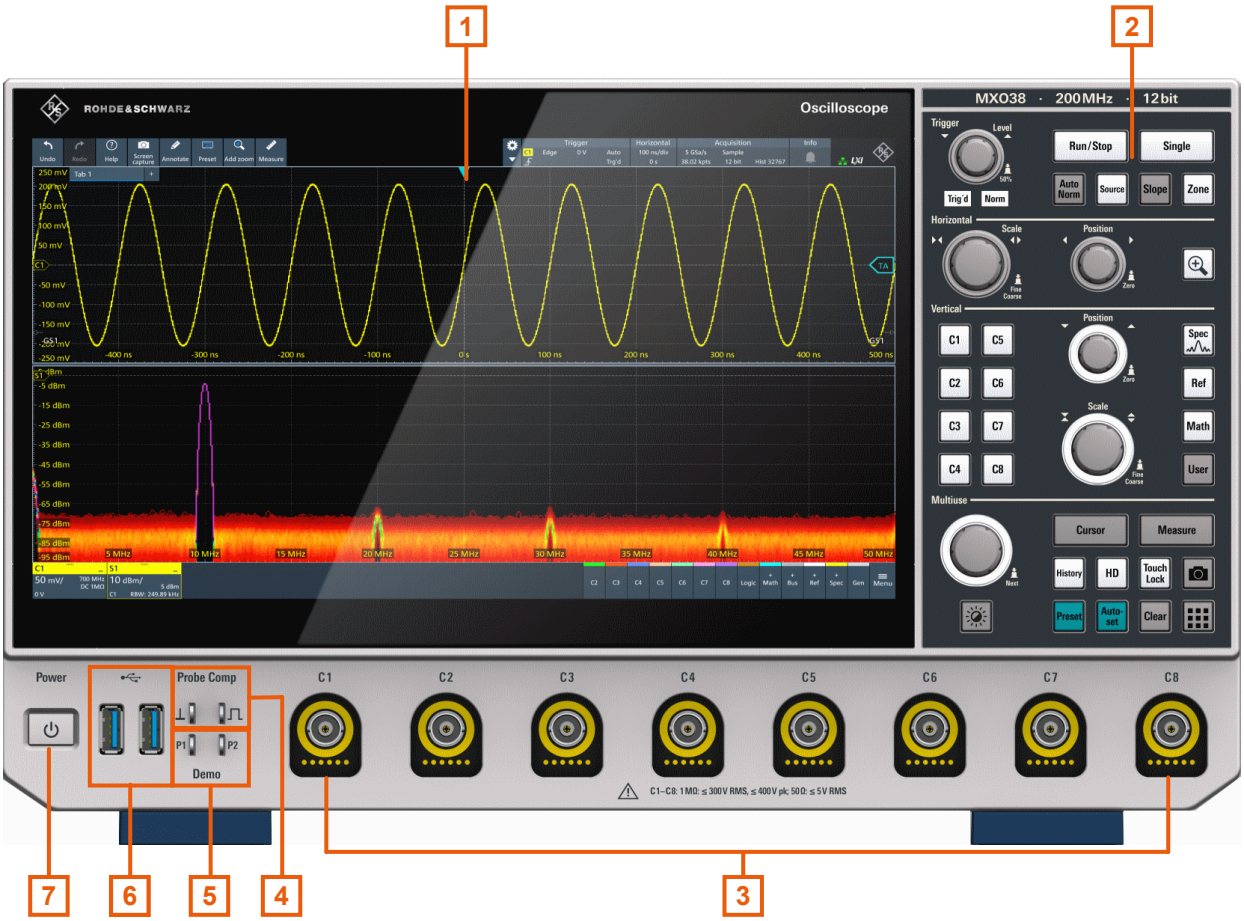



*Figure 3-1: Front panel of an 8-channel MXO 3*

1 = Display
2 = Keys and controls
3 = Analog input channels (4 or 8 channels depending on the model)
4 = Connectors for probe compensation
5 = Connectors for demo signal output
6 = USB connectors
7 = [Power] key

### Analog channel inputs

The MXO 3 has 4 or 8 channel inputs to connect the input signals, depending on the model.

You can connect probes with Rohde & Schwarz probe interface or BNC connectors. The input impedance is selectable, the values are 50 Ω and 1 MΩ.

The maximum input voltage on channel inputs is:

- 400 V ($V_p$) and 300 V (RMS) at 1 MΩ input impedance, derates at 20 dB/decade to 5 V (RMS) above 250 kHz.
- 30 V ($V_p$) and 5 V (RMS) at 50 Ω input impedance

### Probe Comp

Probe compensation terminal to adjust passive probes to the input channel.

⎍ Square-wave signal for probe compensation.

⊥ Ground connector for probes.

For details, see the specifications document.

**Demo**
The pins are intended for demonstration purposes.

**USB**
Two USB 3.1 gen1 interfaces type A. They are used to connect a mouse or keyboard, a USB flash drive for storing and reloading instrument settings and measurement data, and to update the firmware.

### 3.2.2 Side view

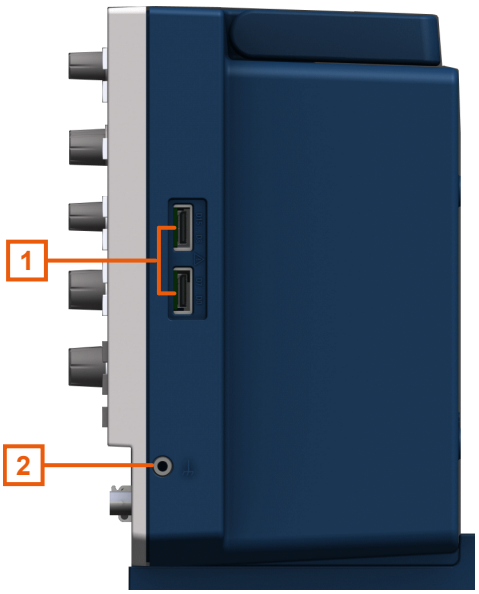



*Figure 3-2: Side view of MXO 3*

1 = Connectors for logic probes (option R&S MXO3-B1)
2 = Grounding terminal

**D15...D8, D7...D0**
Two connectors for logical probes with 8 digital channels each (D0 to D7 and D8 to D15). Using logic channels requires the Mixed Signal Option R&S MXO3-B1.

The maximum input voltage at 100 kΩ input impedance is ±40 V (Vp) and 32 V (RMS), derates to 7 V (RMS) with 20 dB/decade at frequencies above 25 MHz. The maximum input frequency for a signal with the minimum input voltage swing and normal hysteresis setting is 300 MHz.

**Grounding terminal ⏚**
4 mm banana jack to connect ground of DUT and test fixtures, and wrist strap.

### 3.2.3 Rear view

On the rear panel of the MXO 3, you find more connectors and the power supply switch.

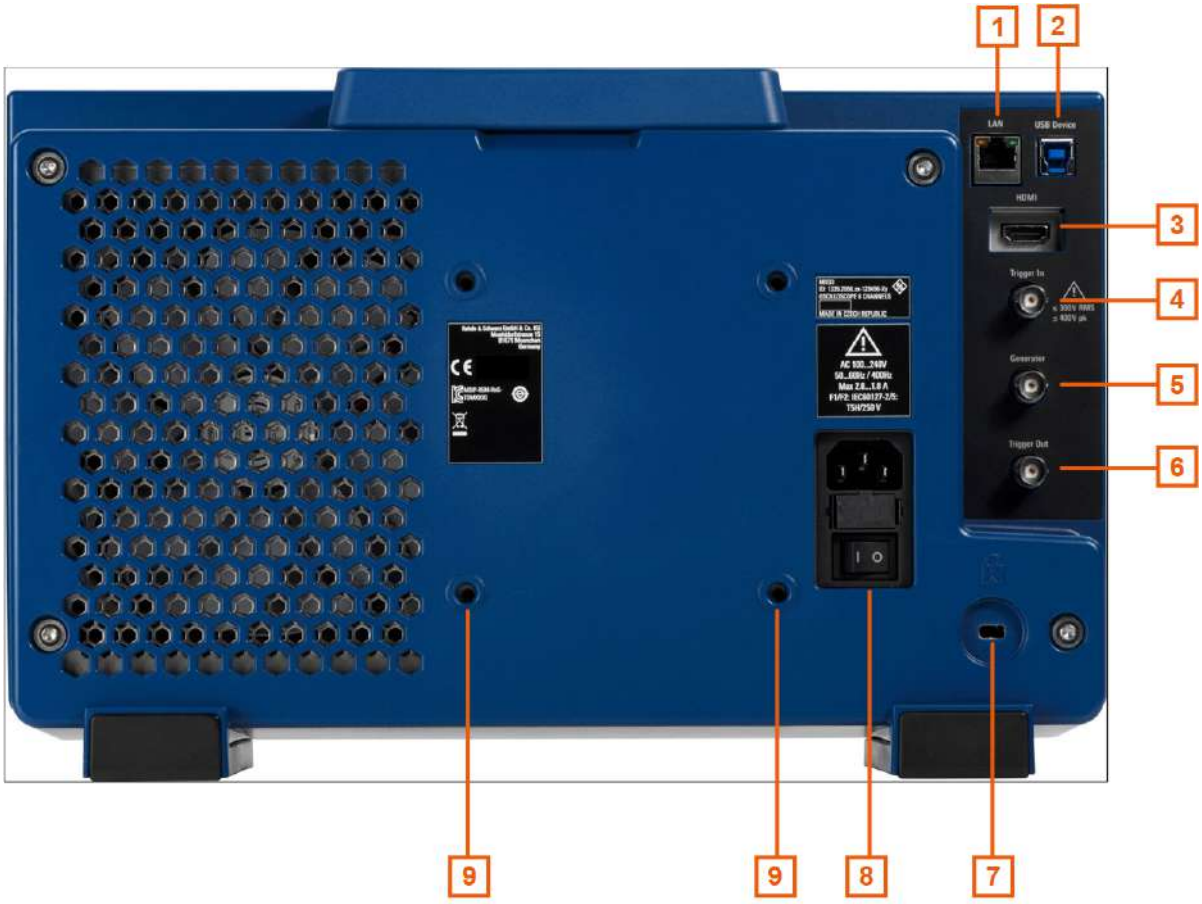



*Figure 3-3: Rear panel view of MXO 3*

1 = LAN connector
2 = USB Device connector, type B
3 = HDMI display output
4 = BNC connector for external trigger input
5 = BNC connector for optional function generator output (option R&S MXO3-B6)
6 = Trigger Out connector
7 = Kensington lock slot to secure the instrument against theft
8 = AC power supply connector and main power switch
9 = VESA mounting holes to mount the instrument on a monitor arm

**LAN**
8-pin RJ-45 connector used to connect the instrument to a Local Area Network (LAN). It supports 10/100/1000BASE-T, LXI compliant.

**USB Device**
USB 3.1 gen1 interface of type B (device USB), to be used for remote control of the instrument.

**HDMI**
HDMI Version 2.0 connector for an external monitor or projector. The monitor shows the complete content of the instrument's screen.

**Trigger In**
The external trigger input is a BNC connector that is used to control the measurement by an external signal. The trigger level can be set from -5 V to 5 V.

For the external trigger input, the maximum input voltage is 400 V ($V_p$) and 300 V (RMS) at 1 MΩ input impedance.

Transient overvoltages on all input connectors must not exceed 400 V ($V_p$).

**Trigger Out**
BNC output that can function as pass/fail output or trigger output.

**Generator: Function generator**
BNC output of the function generator, requires option R&S MXO3-B6.

**AC power supply connector and mains power switch**
The instrument supports a wide range power supply. It automatically adjusts to the correct range for the applied voltage. There is no line voltage selector.

The AC mains power switch disconnects the instrument from the AC power line.

When you power up the instrument, be sure to comply with the warm-up phase specified in the specifications document before you start measurements.

## 3.2.4 Keys and controls

For an overview of the front panel keys, see Figure 3-1.

### 3.2.4.1 Power key

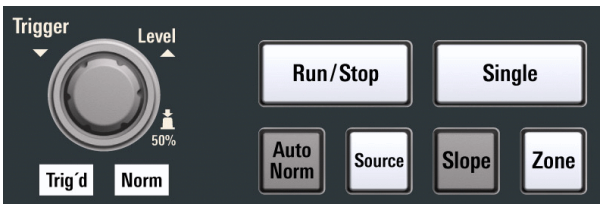
The [Power] key is located on the lower left corner of the front panel. It starts up and shuts down the instrument's software.

The light of the key shows the instrument state, see Section 3.1.9, "Switching on or off", on page 40.

Remote command: `SYSTem:SHUTdown` on page 954.

### 3.2.4.2 Trigger controls

The keys and knob in the Trigger functional block adjust the trigger and start or stop acquisition.




**[Level]**
The rotary knob sets the trigger level for all trigger types that require one level. For trigger types with two levels, the knob sets the lower level. Turn clockwise to move up the trigger level. Press the knob to set the trigger level to 50% of the signal amplitude.

Remote command:
`TRIGger:EVENt<ev>:LEVel<n>[:VALue]` on page 1032

**[Run / Stop]**
Starts and stops the continuous acquisition.

The [Run / Stop] key lights green when the acquisition is running. When the acquisition is stopped, the key lights red.

Remote command:
`RUN` on page 989
`STOP` on page 990

**[Single]**
Starts a defined number of acquisitions.

The [Single] key lights green when the acquisition is running. When the acquisition is stopped, the key lights red.

Press the key again to stop a running acquisition. To set the number of acquisitions, select "Menu" > "Acquisition", and set "N-single/avg count".

Remote command:
`SINGle` on page 990

**[Auto Norm]**
Toggles the trigger mode between "Auto" and "Normal". The current setting is shown on the trigger label.

| | |
|---|---|
| Auto | The instrument triggers repeatedly after a time interval if the trigger conditions are not fulfilled. If a real trigger occurs, it takes precedence. The time interval depends on the timebase. |
| Norm | The instrument acquires a waveform only if a trigger occurs. The "Norm" indicator turns green. |

Remote command:
`TRIGger:MODE` on page 1056

**[Source]**
Toggles the trigger source: C1, C2, C3, etc. The key lights up in the color of the selected channel.

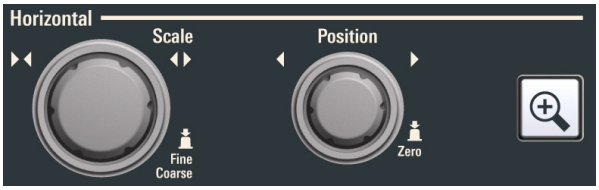
**[Slope]**
Toggles the trigger slope or trigger polarity, dependent on the trigger type. The current setting is shown on the trigger label.

**[Zone]**
Opens the configuration for the zone trigger. A zone trigger combines the trigger condition with the intersection or non-intersection of one or more zones or masks.

Remote command:
`TRIGger:ZONE:EXPRession[:DEFine]` on page 1066

### 3.2.4.3 Horizontal controls

The rotary knobs in the Horizontal functional block adjust the horizontal parameters. These settings are effective for all channel waveforms. In addition, a [Zoom] key is available.




**[Scale]**
The rotary knob adjusts the horizontal scale for all signals. The horizontal scale is also known as timebase.

Turn clockwise to stretch the waveforms. Doing so, the scale value *time/div* decreases. Press the knob to toggle between coarse and fine scale adjustment.

In a spectrum diagram, the knob changes the span.

Remote command:
`TIMebase:SCALe` on page 990
`CALCulate:SPECtrum<sp>:FREQuency:SPAN` on page 1194

**[Position]**
The rotary knob changes the horizontal position of the waveforms. Turn clockwise to move the waveforms to the right. To set the value to zero, press the knob. The current value is shown in the "Horizontal" label above the diagram.

The horizontal position defines the zero point of the diagram. It is the time distance to the reference point, which marks the rescaling center of the horizontal scale.

In a spectrum diagram, the knob changes the center frequency.

Remote command:
`TIMebase:HORizontal:POSition` on page 991
`CALCulate:SPECtrum<sp>:FREQuency:CENTer` on page 1193

**[Zoom]**
Activates a zoom and supports the zoom setup.

If no zoom is on, the first press opens a zoom diagram for the active diagram. If at least one zoom is shown, the keypress opens the "Zoom" dialog. If only one zoom is defined, the next press closes the dialog and removes the zoom. If several zooms are defined and the dialog is open, the key toggles the zooms.

### 3.2.4.4 Vertical controls

The keys and knobs in the Vertical functional block select a signal and adjust the vertical scale and position of the selected signal.

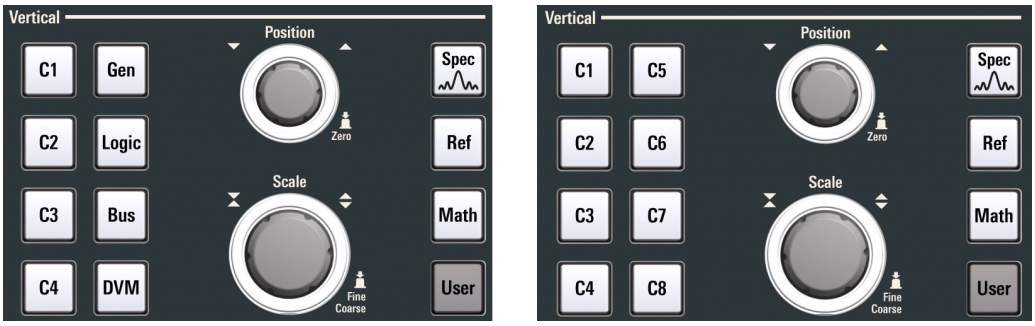



***Figure 3-4: Vertical controls of MXO34 (left) and MXO38 (right)***

**[C<n>]**
Turns on and selects a channel. If the channel is active, the key lights up in the corresponding channel color.

The effect of the keypress depends on state of the channel:
- If channel is off: turns on the channel and selects it.
- If the channel is on and in focus (selected): opens the corresponding channel dialog.
- If the channel is on, but not in focus (not selected): selects the channel waveform.
- If the channel is selected, and the dialog is open: turns off the channel, and closes the dialog.

The vertical rotary knobs are focused on the selected waveform. They are illuminated in the color of the selected waveform.

Remote command:
`CHANnel<ch>:STATe` on page 999

**[Position]**
The vertical [Position] knob changes the vertical offset or the position of the selected waveform. The horizontal axis and the selected waveform are moved vertically. You can select whether the knob changes the offset or the position in the "Menu" > "Settings" > "Frontpanel" > "Knobs" dialog.

- Position indicates the vertical location in divisions.
- Offset moves the vertical center of the selected channel to the offset value.

If the selected waveform is a math or reference waveform, serial bus, or logic channel, the knob changes its vertical position.

The knob lights up in the color of the selected waveform. Turn clockwise to move up the waveform. To set the value to zero, press the knob.

The current offset value is shown in the channel icon.

Remote command:
CHANnel<ch>:POSition on page 1001
CHANnel<ch>:OFFSet on page 1000

**[Scale]**
Sets the vertical scale, which defines the displayed amplitude of the selected waveform.

The [Scale] knob lights up in the color of the selected waveform.

Turn the knob clockwise to stretch the waveform. Doing so, the scale value V/div decreases. Press the knob to toggle between fine and coarse adjustment. For analog waveforms, the scale value is shown in the signal icon.

To get the maximum resolution of the waveform amplitude, make sure that the waveforms cover most of the diagram's height.

Remote command:
CHANnel<ch>:SCALe on page 1000
CALCulate:MATH<m>:VERTical:SCALe[:VALue] on page 1100
REFCurve<rc>:SCALe on page 1110

**[Gen]**
Available only for MXO 3 with 4 input channels.

Opens the "Waveform Generator" dialog, if option R&S MXO3-B6 is installed.

The waveform generator can generate various function and arbitrary waveforms, sweeps, and parallel patterns. For detailed characteristics, refer to the specifications document.

**[Logic]**
Available only for MXO 3 with 4 input channels.

Opens the "Logic" dialog to configure and enable the logic buses L1 to L4. A logic bus (or parallel bus) has up to 16 logic (digital) channels. If logic buses are active, pressing the key toggles these buses.

The key lights up if at least one logic bus is enabled.

**[Bus]**
Available only for MXO 3 with 4 input channels.

Opens and closes the "Protocol" tab in the "Apps" dialog, where you can select the required serial protocol and proceed to the protocol setup. The key lights up if the decoding of a serial bus is active.

**[DVM]**
Available only for MXO 3 with 4 input channels.

Opens the digital voltmeter application.

**[Spec]**
Opens and closes the "Spectrum" dialog. The key lights up if a spectrum is active.

**[Ref]**
Opens the "Reference" dialog box, where you can configure and display reference waveforms. Press the key repeatedly to switch the reference waveform.

If a reference waveform is selected, the vertical rotary knobs are illuminated in white or light gray (default colors), depending on the selected waveform.

**[Math]**
Opens the "Math" dialog, where you can configure the calculation of mathematical waveforms using various mathematic operations on other waveforms. Press the key repeatedly to toggle the selected math waveform. If no math waveform is active, the key closes the dialog.

If a math waveform is selected, the vertical rotary knobs are illuminated in blue (default color).

**[User]**




Performs the action that is assigned to the key in "Settings" > "Front panel" > "Hard-keys". You can assign one of the following functions to the key:

- Force trigger (default)
- Save a saveset with instrument settings
- Load a saveset with instrument settings

### 3.2.4.5 Multiuse keys

The controls in the bottom functional block have various functions.




**[Multiuse]**
The multiuse knob changes the element that is in focus. It lights up when a function is active on the knob.

- If a numeric entry field in a dialog has the focus: turn to increase or decrease the value. Press the knob to toggle fine and coarse increment.
- If a cursor set has the focus: press to toggle the cursor line, and turn to adjust its position.
- If a zoom has the focus: press to toggle the edges of the zoom, and turn to adjust its position.

- If a gate has the focus: press to toggle the complete gate and the edges of the gate, and turn to adjust its position.
- To change the intensity, press the [Intensity] key and turn the knob.

**[Intensity]**
Adjusts the intensity of the waveforms on the screen: Press the [Intensity] key and turn the [Multiuse] knob. The current value is shown in the input box in the upper right corner of the screen.

**[Cursor]**
Starts a cursor measurement: sets vertical and horizontal cursors in the active diagram, and displays results of the cursor measurement. If cursors are already set, the keypress opens the "Cursor" dialog, where you can adjust the settings. The next keypress closes the dialog.

Cursors are markers which are placed at points of interest on a waveform. The instrument measures the cursor positions and delta values between parallel cursors.

If cursors are already set but not in focus, pressing the key sets the focus on the cursor set. You can adjust the cursor lines using the [Multiuse] knob.

If a cursor set has the focus, the key opens the "Cursor" dialog. If one cursor set is "On", the next press closes the dialog. If several cursors are "On", pressing the key toggles the cursor sets.

**[Measure]**
Opens the "Measurement" dialog, where you can set up various measurements. If the dialog is open, pressing the key closes the dialog.

**[History]**
Activates the history and opens the history player. The next keypress closes the player. The history shows stored acquisitions that were acquired before the current one. The key is illuminated as long as the history is active.

**[HD]**
Activates the high definition mode and opens the HD settings dialog.

If the dialog is open, pressing the key closes the dialog. If the dialog is closed, the key disables the high definition mode. The key is illuminated as long as the HD mode is active.

**[Touch Lock]**
Locks the touchscreen to prevent unintended use. When the touchscreen is off, the key is illuminated. Press again to unlock the touchscreen.

**Camera** 
Performs the action that is assigned to the key in "Settings" > "Front panel" > "Hardkeys". By default, the key saves a screenshot of the waveform display.

**[Preset]**
Resets the instrument to a predefined state and starts the continuous acquisition. All measurements, mask tests, zoom, and most individual settings are deleted, and all channels except for channel 1 are disabled.

You can define preset configurations and save them to a file. The [Preset] key can be configured to set either factory defaults or a user-defined preset configuration.

**[Autoset]**
The instrument analyzes the enabled channel signals, and adjusts appropriate horizontal, vertical, and trigger settings to display stable waveforms.

In particular, the horizontal scale is set to a common value for all active channels, record length and sample rate are set to automatic detection. Vertical scale and offset are set for each channel individually. Probe settings, channel coupling and impedance remain unchanged. The trigger source is set to the signal with the lowest frequency, and the trigger type to positive edge. The trigger level is adjusted, and the trigger position is set to the reference point. Measurements, math and spectrum waveforms are kept. Undo and redo of an autoset is possible.

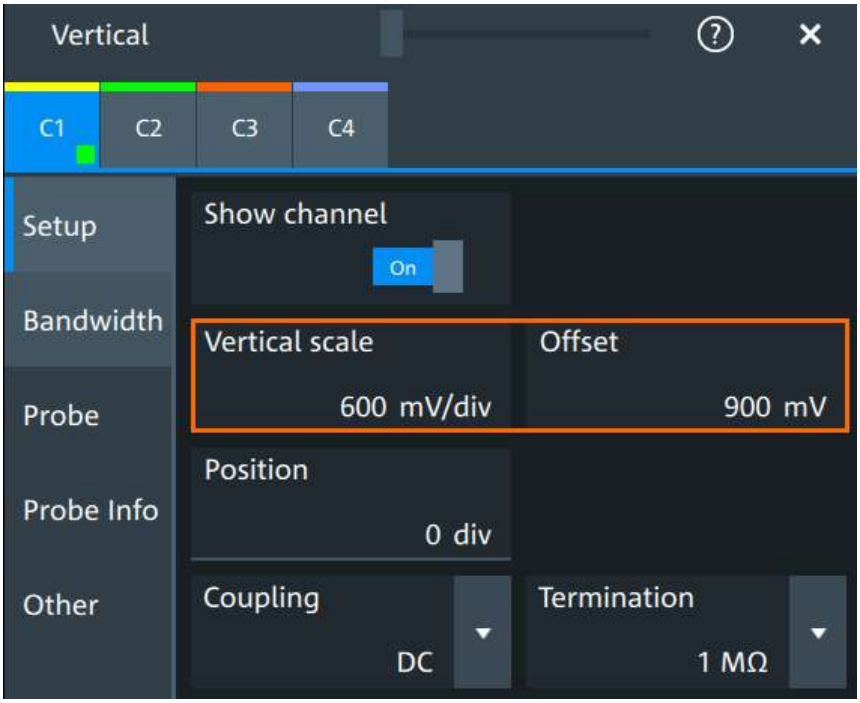
**[Clear]**
Deletes all measurement results including long-term measurement and statistics, all waveforms, and the history.

**Apps**
Opens the "Apps" dialog, where you can start an application, serial protocol or another analysis function.

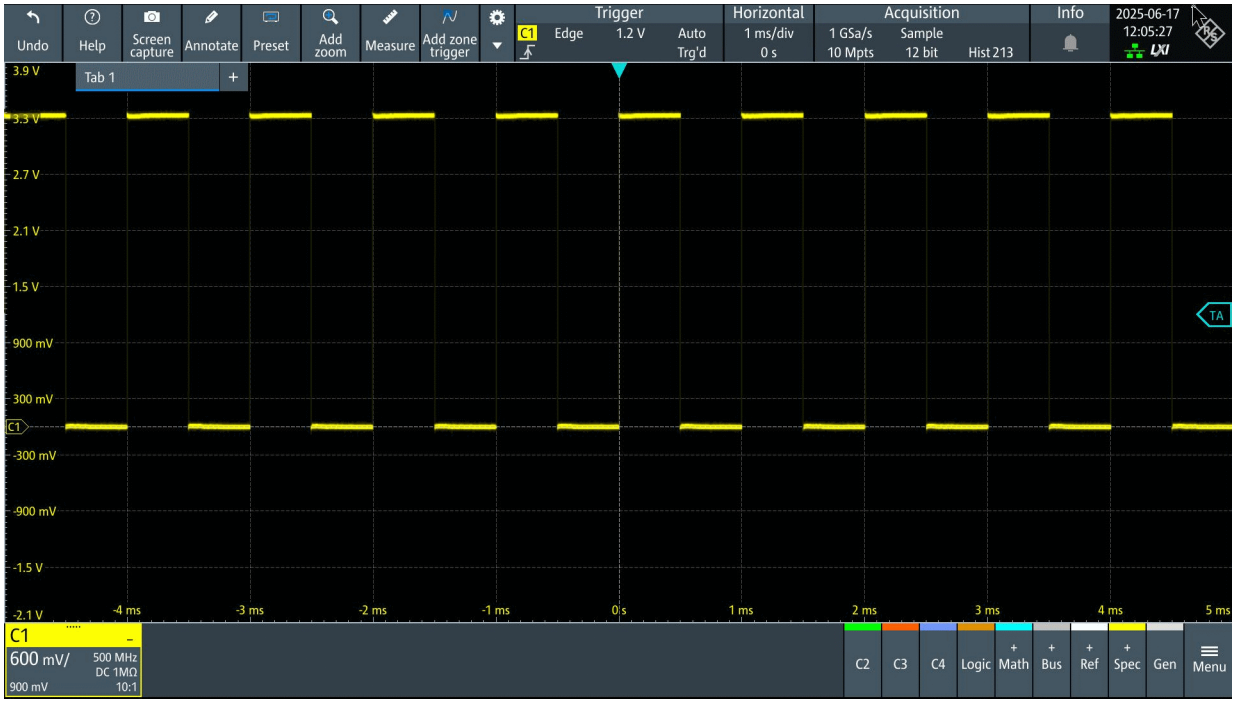
### 3.2.5 Checking the functionality

To check if the instrument works correctly, you can use the probe compensation signal and check the displayed signal.

1. Perform a self-alignment of the instrument:
   a) Warm up the instrument. The minimum warm-up time is indicated in the specifications document.
   b) Remove all probes from the input connectors.
   c) Open the "Menu", and select "Settings" > "Maintenance" > "Alignment".
   d) Tap "Start Alignment". Wait until the alignment has finished successfully.
2. Connect the probe's ground connector to the ground pin ⊥, and the tip to the square wave pin ⎍.
3. Press the [Preset] key.
4. Connect the probe to the input connector [C1].
   The instrument recognizes the probe, and a signal is displayed in the diagram.
5. Tap the signal icon of C1 to open the vertical settings.
6. In the "Vertical" > "Setup" dialog, adjust the following:
   - "Scale" = 600 mV/div
   - "Offset" = 900 mV




7. Turn the horizontal [Scale] knob to set the horizontal scale to 1 ms/div.
8. Press the [Source] key to set the trigger source to C1. The color of the key indicates the selected channel.
9. Turn the [Level] knob to set the trigger level to 300 mV.
10. Check the rectangle signal on the screen. The displayed signal should have an amplitude of about 3.3 V, which covers 5.5 divisions.




11. Repeat steps 4 to 10 for all other channels. Make sure to adjust the vertical settings for the connected channel, and also to set the trigger source to the connected channel.

# 4 Operating the instrument

There are three ways to operate the MXO 3.

**Manual operation**

Use the touchscreen, keys and rotary knobs, or an optional mouse and/or keyboard. The principles of manual operation are explained in this section.

**Remote control**

Create programs to automatize repeating settings, tests, and measurements. The instrument is connected to a computer that runs the program.

This way of operation is described in: Section 18, "Remote control commands", on page 937.

**Remote operation**

For remote monitoring and operation of the instrument, a VNC server is installed on the MXO 3. You need a LAN connection to the computer, and a VNC client or web browser to connect to the instrument.

For details, refer to Section 17, "Network operation", on page 915.

## 4.1 Means of manual interaction

The MXO 3 provides the following means of manual interaction, which you can use alternatively or complementary:

- Touchscreen:
  Using the touchscreen allows for direct interaction with the instrument. Use your finger to place waveforms on the screen, mark areas for zoom, set parameters in dialogs, enter data, and much more. The control elements and actions on the screen are based on common concepts, and you easily become familiar with the user interface.

  Tapping the screen works like clicking mouse buttons:

  – Tap = click: Selects a waveform or parameter, or provokes an action.
  – Double-tap = double-click: Has the same effect as touch and hold, it opens the on-screen keyboard or keypad, or a specific editor if available.

  Use gestures to scale the waveform:

  – Spread or pinch two fingers horizontally to change the time scale.
  – Spread or pinch two fingers vertically to change the vertical scale of the active waveform.
- Function keys and rotary knobs:
  The front panel provides frequently used functions and controls to operate the instrument. Use knobs to set levels and scales, and keys to initiate actions and to open dialogs.

- Optional mouse and/or keyboard
  These devices work in the normal manner.

## 4.2 Touchscreen display

### 4.2.1 Information on the display

The GUI shows the waveforms and measurement results, and also information and everything that you need to control the measurements. All waveform-related GUI elements are explained in this section. An overview of control elements - like dialog, toolbar - is given in Section 4.2.2, "Control elements on the GUI", on page 58.

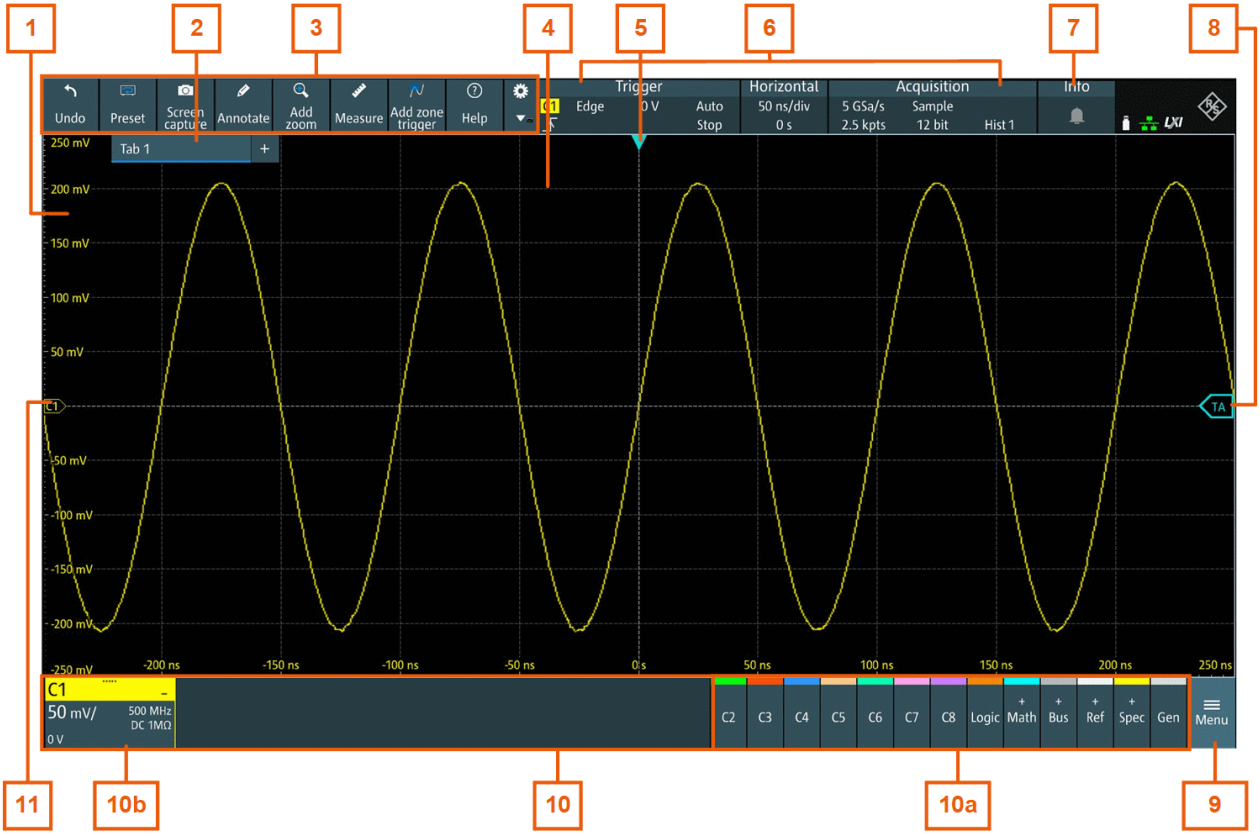



***Figure 4-1: Information in the graphical user interface***

1 = Diagram
2 = Layout tab
3 = Toolbar
4 = Grid
5 = Trigger position
6 = Trigger, horizontal and acquisition label
7 = Info
8 = Trigger level

9 = Menu
10 = Signal bar with signal activators (10a), and active waveforms (10b)
11 = Waveform markers indicate the offset of the waveforms

**Diagram**
A diagram shows one or more waveforms: analog and digital channels, reference and math waveforms, tracks, and serial buses. Zoom details, spectrum and other special waveforms are shown in separate diagrams. One diagram can hold up to 16 analog waveforms, i.e. analog channels, reference and math waveforms, and tracks.

To arrange the diagrams on the screen, the Rohde & Schwarz SmartGrid function helps you to find the target place simply and quickly. For details, see Section 4.7, "Rohde & Schwarz SmartGrid", on page 69. You can also adjust the diagram size by dragging the diagram border.

**Layout tab**
A layout shows a set of diagrams and result tables. You can configure up to four layout tabs using the SmartGrid functionality. For details, see Section 4.7, "Rohde & Schwarz SmartGrid", on page 69.

**Grid**
The grid shows the vertical and horizontal divisions. The division lines are labeled with the correspondent values. The grid labels have the color of the waveform to which they belong. If several waveforms are shown in one diagram, the grid has the color of the selected waveform.

**Trigger position and trigger level**
The blue markers show the horizontal position of the trigger and the vertical trigger level. You can touch and move the trigger markers in the diagram to set the positions. The trigger point is the zero point of the diagram.

The trigger position can be moved outside the diagram.

**Trigger**
The "Trigger" label shows the main trigger settings. If you tap the label, the corresponding dialog opens.

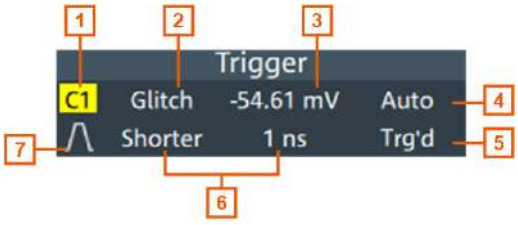



***Figure 4-2: Trigger label***

1 = Trigger source
2 = Trigger type
3 = Trigger level
4 = Trigger mode
5 = Trigger state
6 = Trigger type specific settings
7 = Trigger slope

See also: Section 7.1.1, "Trigger information", on page 164.

**Horizontal**
The "Horizontal" label shows the main timebase settings. If you tap the label, the corresponding dialog opens.

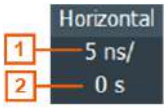



***Figure 4-3: Horizontal label***

1 = Time scale
2 = Horizontal position

**Acquisition**
The "Acquisition" label shows the main acquisition settings, and the progress of long acquisitions. If you tap the label, the corresponding dialog opens.




***Figure 4-4: Acquisition label***

1 = Sample rate (before interpolation)
2 = Record length. A yellow warning icon indicates insufficient memory.
3 = Acquisition mode
4 = Number of acquired waveforms
5 = Resolution. Active high definition mode is indicated by "HD".

**Info**
The "Info" button on the toolbar points to the status messages and other information. To open the message box, tap the button.

See also: Section 4.13, "Information and notifications", on page 79.

**Signal bar**
The signal bar summarizes all waveforms as described in Section 4.5, "Signal bar", on page 67.

## 4.2.2 Control elements on the GUI

The GUI provides everything that you need for control, to analyze waveforms, and to get measurement results. Figure 4-5 shows the control elements at a glance.




*Figure 4-5: Control elements on the graphical user interface*

1 = Toolbar
2 = Tab in a dialog
3 = Dialog
4 = Result table
5 = Signal bar
6 = Menu

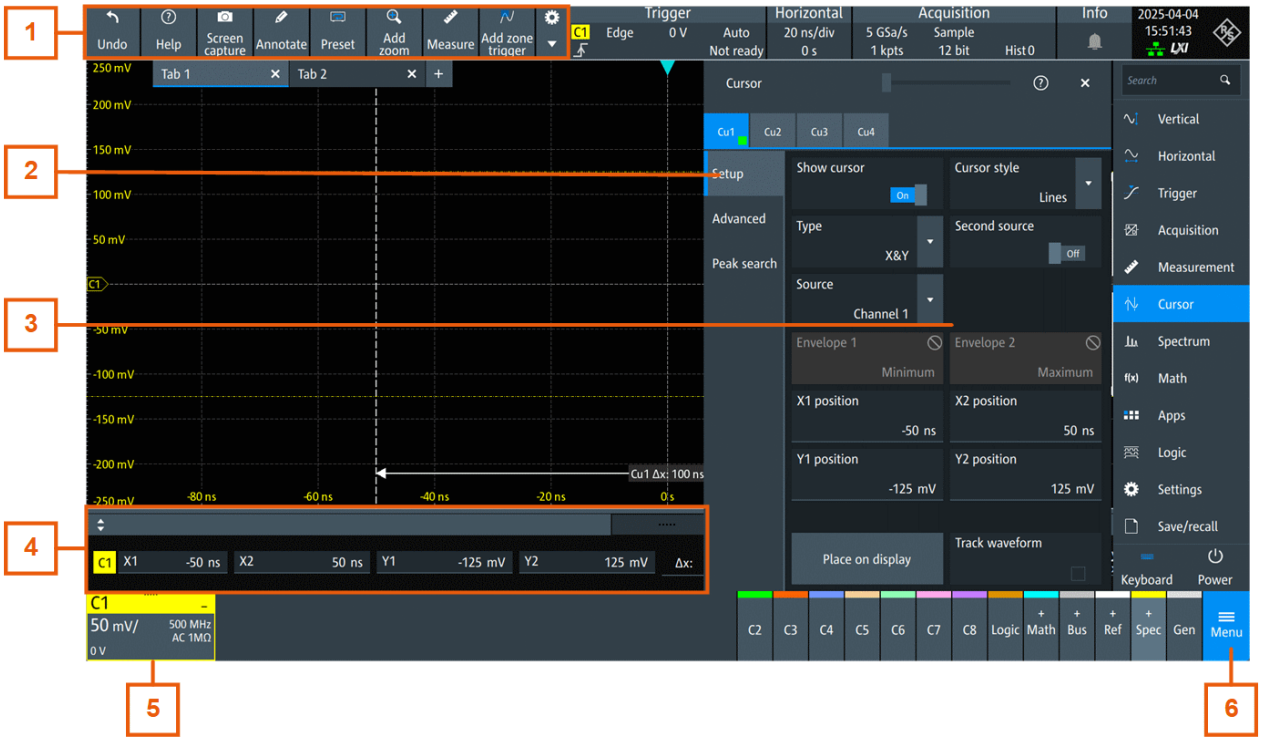



*Figure 4-6: Control elements on the graphical user interface*

1 = Toolbar
2 = Tab in a dialog
3 = Dialog
4 = Result table
5 = Signal bar
6 = Menu

### Toolbar (1)

The icons on the toolbar provide quick and easy access to the most important functionality. For a detailed description, refer to Section 4.4, "Toolbar", on page 62.

If you adjust the settings of an analyzing function, e.g., cursor measurement, the overlay menu is shown instead of the icons. The overlay menu provides the most important settings of the current action. If you need more settings, "Setup" opens the corresponding dialog.

### Dialog (2, 3)

The tabs of the dialogs contain all task-oriented settings and operations, and buttons for calling related tabs. The usage of dialogs is described in Section 4.9, "Using dialogs", on page 73.

### Result table (4)

If you perform cursor or automatic measurements, the result table shows the results of the action below the diagrams. You can move the result tables to the left or right of the diagrams.

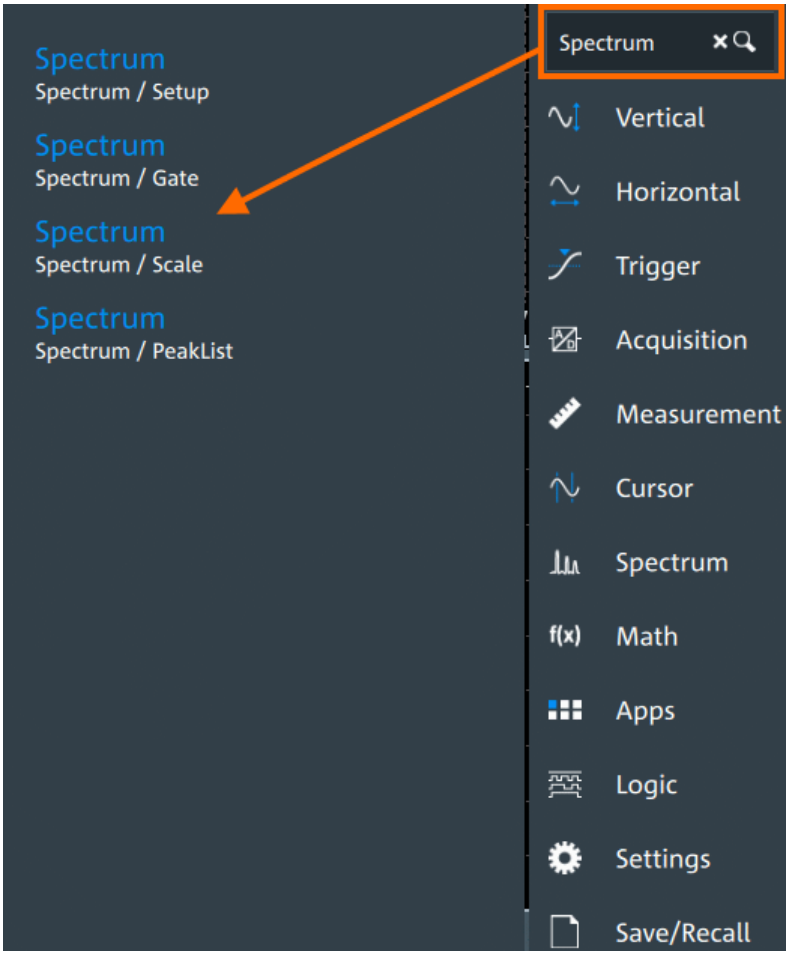
See also: Section 4.11, "Displaying results", on page 76.

**Signal bar (5)**
The signal bar summarizes all waveforms as described in Section 4.5, "Signal bar", on page 67.

**Menu (6)**
The menu provides access to the complete functionality of the MXO 3.

At the bottom of the menu, you find the "Keyboard" icon to open the on-screen keyboard. The "Power" icon shuts down the firmware.

At the top of the menu, you can find a search field for a quick access to any parameter you need. A list of all found results is shown on the left side of the menu. Tap the result entry and the dialog containing this parameter opens.

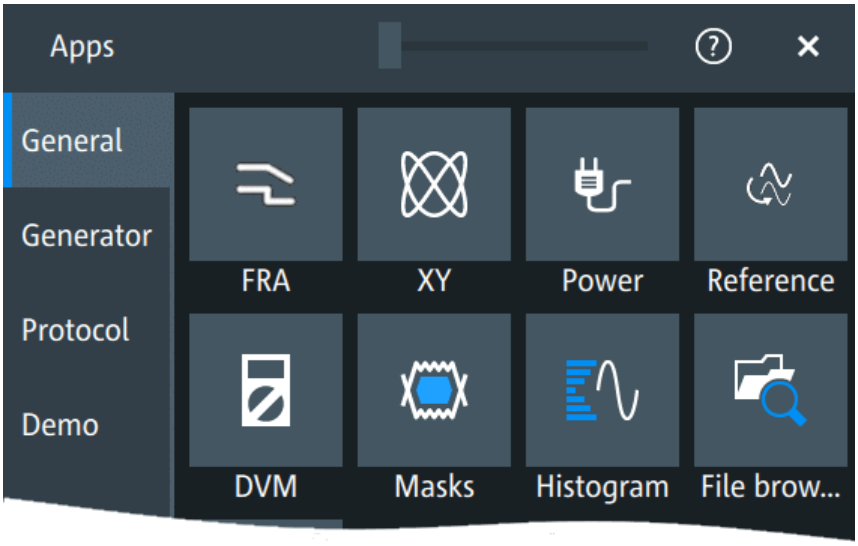



## 4.3 Applications

The "Apps" dialog provides fast access to all available applications, for example, to serial protocols, frequency response analysis or file browser.

► To open the "Apps" dialog:

- Open "Menu" > "Apps".
- Press the [Apps] key on the front panel.




## 4.4 Toolbar

The toolbar provides direct access to important control and measurement functions.




By default, the toolbar shows the most frequently used functions. You can configure the content of the toolbar, see Section 4.4.2, "Configuring the toolbar", on page 63.

### 4.4.1 Using the toolbar

Using the toolbar is easy and straightforward.

Some of the toolbar functions are one-click actions. These actions are performed immediately when you tap the icon.

Other toolbar functions are analyzing functions. These actions are interactive actions.

**To use analyzing functions (interactive actions)**

1. Tap the icon of the function in the toolbar.
2. Check and adjust the settings in the overlay menu.
3. Select the source waveform if needed. For zoom, drag a rectangle, or tap the diagram to define the zoom area.
4. If the overlay menu does not close automatically, tap the icon.

### 4.4.2 Configuring the toolbar

You can configure the content of the toolbar so that only the required functions are displayed.

The toolbar configuration is part of the user preferences. It is retained when you switch off and on the instrument, and you can save it in the user preferences and user-defined preset.

1. To open the toolbar configuration, tap the icon in the toolbar:

2. Select the required toolbar functions:
   a) In the "Show/Hide tools" section, disable all functions that you do not need.
   b) In the "Show/Hide tools" section, enable the functions that you want to add to the toolbar.
   c) In the "Drag to rearrange tools" section, move the icons to arrange them as required.
3. To empty the toolbar, select "Deslect all".
4. To get the factory configuration of the toolbar, select "Restore default".

A detailed description of the toolbar functions is given in Section 4.4.3, "Toolbar functions", on page 63.

Remote commands:

- `DISPlay:TOOLbar:COUNt?` on page 963
- `DISPlay:TOOLbar:DESelect` on page 963
- `DISPlay:TOOLbar:RESTore` on page 964

### 4.4.3 Toolbar functions

This section describes all toolbar functions.



You can configure the content of the toolbar, see Section 4.4.2, "Configuring the toolbar", on page 63.

Undo....64
Redo....64
Help....64
Recall....64
Save setup....64
Screenshot....64
Annotate....65
Clear....65
Autoset....65
Preset....65

Run / stop and Run single....65
Force trigger....65
Add zoom....65
Add cursor....65
Measure....65
Add mask....66
Histogram....66
Add gate....66
Edit spectrum....66
Save waveform....66
Add zone trigger....66
Delete....66
File browser....66
DVM....66
XY....66
Image viewer....67
Gen1....67



**Undo**
Undoes the last setting actions step by step. On a keyboard, you can use the shortcut [Ctrl]+[Z].

Some actions cannot be revoked: locking the touchscreen with [Touch Lock], and saving data.

The undo stack is deleted when reloading settings from file, and during reference waveform actions (save, load and preset with active reference waveform).



**Redo**
Recovers the undo steps in reverse order. On a keyboard, you can use the shortcut [Ctrl]+[Y].



**Help**
Enables the context help display. The help window opens when you tap a parameter.

See also: "To display the context help" on page 81.



**Recall**
Opens a window to select and load instrument settings that were previously stored in a saveset.



**Save setup**
Saves the current instrument settings in a saveset.

You can reload the saveset using the "Recall" toolbar icon, or using "Menu" >"Save/ recall" > "Recall" tab > "Saveset".

The filename is created according to the autonaming pattern, defined in "Menu" > "Settings" > "Save / recall" > "Autonaming" tab.



**Screenshot**
Saves a screenshot of the current display using the settings defined in "Menu" >"Save/ recall" > "Save" tab > "Screenshot".



**Annotate**
Displays drawing tools for marking areas on the diagram, and for adding text. You can also select the color.

See also: Section 4.12, "Adding annotations", on page 77.



**Clear**
Deletes all measurement results including all waveforms and statistics.

Remote command:
`DISPlay:CLR` on page 984



**Autoset**
Performs an autoset. The instrument analyzes the enabled channel signals, and adjusts appropriate horizontal, vertical, and trigger settings to display stable waveforms.

The icon has the same functionality as the corresponding key on the front panel. It is useful when you operate the instrument remotely.



**Preset**
Resets the instrument to a predefined state and starts the continuous acquisition. All measurements, mask tests, zoom, and most individual settings are deleted, and all channels except for channel 1 are disabled.

The icon has the same functionality as the corresponding key on the front panel. It is useful when you operate the instrument remotely.





**Run / stop and Run single**
Starts and stops the continuous acquisition, or starts a defined number of acquisition cycles. The icons have the same functionality as the corresponding keys on the front panel. They are useful when you operate the instrument remotely.



**Force trigger**
Starts an immediate single acquisition. If the acquisition is running in normal mode and no valid trigger occurs, use "Force trigger" to confirm that a signal is available. Then you can use the displayed waveform to determine how to trigger on it.



**Add zoom**
Adds a zoom diagram, an area of the acquired waveform which is visually enlarged. Click a diagram to create a zoom or draw a rectangle to define the range of the zoom.



**Add cursor**
Adds a cursor set. Select the cursor type and the source to be measured.

See also: Section 10.1, "Cursor measurements", on page 260.



**Measure**
Adds one or more measurements to the waveform.

Tap the icon, and select the category and the measurements in the overlay menu. Select the waveform to be measured, and close the overlay menu.



**Add mask**
Starts the on-screen mask definition and the testing against the defined mask.

Tap the icon and draw a rectangle for the first mask segment. Use the overlay menu to add more segments to the mask, to create more masks, and to edit mask segments on the display. You can also move the mask segments on the screen.



**Histogram**
Creates a histogram. Tapping the icon opens the overlay menu, where you select the histogram type. Tap a diagram, or drag a rectangle on the diagram to define the histogram window. The histogram for the selected waveform is created.



**Add gate**
Creates a new gate.

Tap the icon, and then tap the diagram or draw a rectangle to define the limits of the gate. The "Gate" dialog opens, where you can adjust the geate settings and add measurements to use the gate.



**Edit spectrum**
Edit the existing spectrum settings in the overlay menu.



**Save waveform**
Saves the selected waveform.



**Add zone trigger**
Defines a zone trigger, which combines the trigger condition with the intersection or non-intersection of one or more zones.

See also: Section 7.9, "Zone trigger", on page 199.



**Delete**
Removes waveforms, diagrams, zooms, trigger zones and other elements from the display.

Tap the "Delete" icon. A recycle bin icon marks all objects that can be deleted. Tap this icon to remove an object. Tap the "Delete" icon again to disable the function.



**File browser**
Opens the file browser dialog.

See: Section 13.9, "File browser dialog", on page 486.



**DVM**
Starts the digital voltmeter application.

See: Section 12.5, "Digital voltmeter", on page 443.



**XY**
Starts the XY-plot application.

See: Section 12.6, "XY-plot", on page 446.



**Image viewer**
Starts the screenshot viewer.

See: Section 13.8.3, "Image viewer", on page 486.




**Gen1**
Turns on or off the waveform generator. For better visibility, a gererator icon is darker than the other icons in the toolbar.

## 4.5 Signal bar

The signal bar is the control center for all waveforms. All enabled waveforms are shown on the left side of the signal bar. On the right side of the signal bar, you see the signal activators of inactive waveforms and buses. On the right of the signal bar, you find signal activators for the waveform generator and ScopeSync connections.

When you tap a signal activator, the corresponding dialog opens. to enable the waveform. The signal activators for analog and digital channels and for math waveforms also activate the waveform.

Each waveform is represented by a signal icon. If the waveform is shown in a diagram, the signal icon displays its main settings. If you tap a signal icon, the dialog with settings for this waveform opens. If you tap the "Minimize" icon on the signal icon, the waveform switches from the diagram area to the signal icon: the icon is grayed out. See Section 4.6, "Working with waveforms", on page 67 for a detailed description.




*Figure 4-7: Signal label of an active analog channel*

1 = Vertical scale
2 = Offset
3 = Bandwidth
4 = Coupling and termination

If many waveforms are active, then waveforms of the same type are grouped in one signal icon. Tap the group icon to open the individual signal icons.

## 4.6 Working with waveforms

The MXO 3 can create and display many waveform types. The most important are:

- Channel waveforms:
  For each analog input channel, one channel waveform is shown.
- Reference waveforms:
  Reference waveforms can be used as reference for comparison and analysis.
- Math waveforms:

Mathematic waveforms can be created with mathematic operations performed on channel, reference, and other math waveforms.

- Zoom waveforms:
  Zoom waveforms show the details of waveforms.
- Digital waveforms:
  The Mixed Signal Option R&S MXO3-B1 provides 16 digital channels grouped in two logic probes (pods) with 8 channels each.

### Waveform handling

The MXO 3 can show and analyze many waveforms. To handle this multitude while keeping track of it, the MXO 3 provides intelligent support:

- The color system helps to distinguish the waveforms.
  The color of the vertical rotary knobs indicates the signal that is focused (selected).
  The color of each waveform can be changed, the color of its signal icon and of the illuminated keys is adjusted to the new color. Alternatively, a color table can be assigned to a waveform.
  Settings: "Menu" > "Settings" > "Appearance" > "Colors" tab.
- You can arrange the waveforms in one diagram, or in separate diagrams. The Rohde & Schwarz SmartGrid function helps to arrange the waveforms.
  See also: "Diagram" on page 57.

### Waveform states

Depending on its place on the screen and the effect of settings, a waveform has one of the following states:

- Off
- Active:
  The waveform is shown in a diagram
- Selected:
  One of the active waveforms that has the focus. In each diagram, one of the assigned waveforms is selected – it appears "on top" in the diagram, and the grid labels have the color of the selected waveform.
  The vertical [Position] and the [Scale] knobs are illuminated with the color of the selected waveform.
- Minimized:
  The waveform signal icon is greyed out, and the waveform is removed from the diagram.

### To switch a waveform on

A channel waveform is activated when you connect the probe. You can switch it on and off according to your needs.

► Choose one of the following ways:

- Press the channel key.
- Select the signal activator in the signal bar.
- In the "Vertical" dialog box, select the channel. Tap "State" > "On".

The waveform is now active, selected, and is shown in the diagram.

Remote command: CHANnel<ch>:STATe on page 999

**To select a waveform**

► Choose one of the following ways:

- Tap the waveform in the waveform diagram.
- Tap the signal icon.
- To select a channel, reference, or math waveform, press the corresponding key.

**Note:** Zoom waveforms in zoom diagrams cannot be selected.

**To minimize a waveform**

► Tap the "Minimize" icon in the upper right corner of the waveform's signal icon in the signal bar.

The waveform disappears from the diagram and the signal icon turns lighter gray.

► To set the waveform back to its previous diagram immediately, use "Undo".

**To arrange a waveform using the SmartGrid**

► See Section 4.7, "Rohde & Schwarz SmartGrid", on page 69.

**To switch off a waveform**

► Do one of the following:

- Tap the "Delete" icon in the toolbar, and then signal icon of the waveform. You can also tap the waveform directly. If several waveforms overlap or lie close together, a selection list is shown.
- To switch off a minimized waveform, tap the "Close" icon in the upper right corner of the signal icon.
- If the waveform is an input channel:
  – Disable "State" in the "Vertical" > "Setup" tab.
  – Select the channel. Then press its channel key twice.

Remote command: CHANnel<ch>:STATe on page 999

## 4.7 Rohde & Schwarz SmartGrid

The Rohde & Schwarz SmartGrid helps to create and arrange the diagrams on the screen with drag&drop. The diagram layout depends on the position where you drop the signal icon, in relation to an existing diagram.

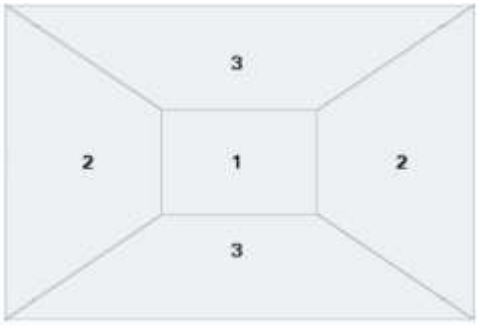



***Figure 4-8: SmartGrid positions***

1 = In the existing diagram, overlay of signal
2 = New diagram on the left or right
3 = New diagram above or below

The diagram configuration is deleted when you use "Preset" or `*RST`.

**To arrange a waveform using the SmartGrid**

1. Select the layout that you want to rearrange.
2. Drag the signal icon to the diagram area, and move it around.
   The Rohde & Schwarz SmartGrid appears and a blue or highlighted area shows where the waveform will be placed.




3. Drop the waveform in the target area.
   The waveform appears in an existing or in a new diagram, and it is selected for further actions.
4. To change the size of a diagram, touch the border between two diagram frames and drag it to the required position.
5. If a diagram is empty, you see a trash can icon in the diagram center. Use the icon to delete the diagram, or drag waveform icons into the diagram.

Remote commands: see Section 18.8.3, "SmartGrid", on page 957.

**Working with layouts**

A SmartGrid configuration of one or more diagrams is called "tab" or "layout". You can define several layouts and switch between them.




1 = Tab 1 (layout 1, blue underline indicates that the set is currently displayed)
2 = Tab 2 (layout 2)
3 = Tap to remove a layout
4 = Tap to add a layout

You can add up to four layouts at the upper left corner of the screen:

1. Tap on the ⊞ icon next to the layout.

   A new layout is created.

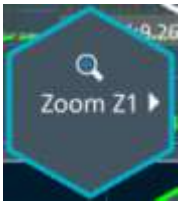
2. To change the layout name, touch and hold the tab name. The on-screen keyboard opens to enter the new name. Names must be unique.

3. To remove the layout, tap on the ⊠ next to the layouts title.

## 4.8 Using the selection menu

The selection menu, or comb menu, helps select objects on the screen if several objects are close beside. These objects are, for example, waveforms, zoom edges, gate limits, trigger zones and cursor lines. When you tap or click a point in a diagram, the firmware determines the objects around the point and provides an appropriate menu. Each found object is represented by a comb. The selection menu is created and arranged anew at each finger tap, there are no predefined positions for the objects.

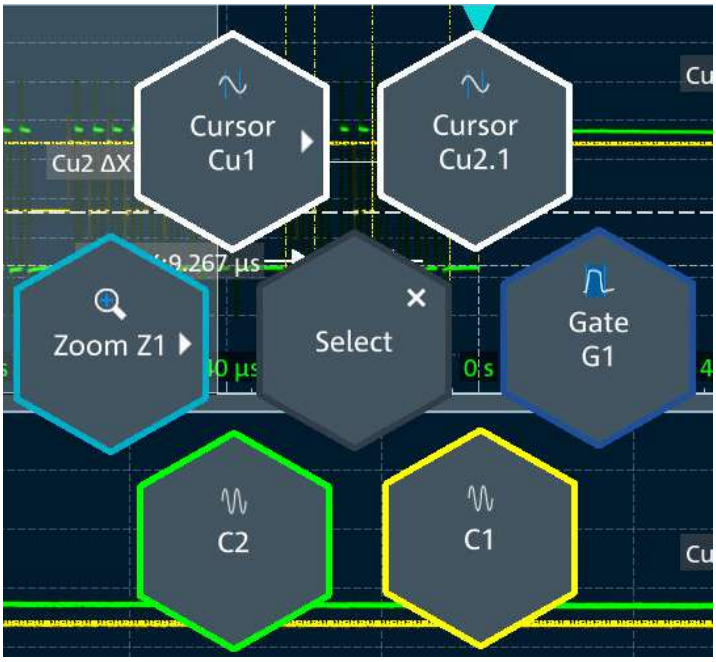
*Table 4-1: Selection menu icons*

| Symbol | Description |
|---|---|
|  | Comb |
|  | : indicates that a submenu is available.<br>If more than 6 objects are detected, related objects are grouped to submenus.<br>Tap on the object group to display the submenu |
|  | : indicates that there are more than 6 objects that are available. In this case some of the objects are not directly displayed.<br>Tap on one of the combs and rotate the selection to display all the choices. |

### To use the selection menu

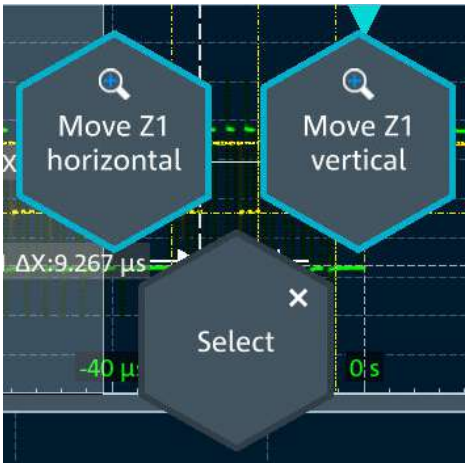
1. Tap the screen at the point of interest.

   The selection menu provides all objects that were found around the point. In the image below, the objects of "Cursor Cu1" and "Zoom Z1" are grouped in submenus.



2. Select the comb of the needed object, or object group.
3. If you have selected an object, the object gets the focus. Change its position directly on the screen or with the [Multiuse] knob, or perform other actions depending on the selected object.
4. If you have selected an object group, the submenu opens.

   The image shows the submenu of "Zoom Z1".




5. In the submenu, select the comb of the needed object.
6. Modify the selected object as required.

## 4.9 Using dialogs

All functionality is provided in dialogs as known from computer programs. This section provides an overview of the accessing methods and describes how to use the dialogs.

Each dialog has icons in the upper right corner:

| | |
|---|---|
|  | Closes the dialog. |
|  | Opens the help window for the dialog. |
|  | Shift sideways to change the transparency of the dialog. |



For direct access to important control and measurement functions, use the toolbar, see Section 4.4, "Toolbar", on page 62.

**To open a dialog**

► Perform one of the following actions:

- Open the "Menu", and select the menu entry.
- Press the function key on the front panel.
- Double-tap a result to open the corresponding settings dialog.
- To open the "Vertical" dialog of a waveform, tap the signal icon.
- Tap the "Horizontal", "Acquisition" or "Trigger" label to open the corresponding dialog.

**To close a dialog**

► Tap the "Close" icon in the upper right corner.

## 4.10 Entering data

To set parameter values and enter other data, you use the various knobs and the on-screen keypad or keyboard.

**To use scale, position and level knobs**

The instrument has dedicated rotary knobs to set vertical and horizontal positions and scale, and the trigger level.

1. Turn the knob to change the value.
2. Press the knob:
   - [Scale]: to toggle the increment.
   - [Position]: to set to zero.
   - [Level]: to set the trigger level to 50% of the signal.

**To enter values with the on-screen keypad**

To enter numeric values and units in dialogs, an on-screen keypad is available.

1. Double-tap the entry field. The on-screen keypad opens.




2. Enter a numeric value using the following methods:
   - To use the default value, tap "Reset" (if available).
   - To get the value that was used before the keypad was displayed, tap "CE".
   - To use the minimum or maximum value, tap "Min" or "Max", respectively.
   - To increase the displayed value in fixed steps, tap "Inc".
     To decrease the value in fixed steps, tap "Dec".
   - To enter a user-defined value, tap the numbers and complete the entry by tapping the unit button.
     – "±" changes the sign of the value.
   - To calculate a value, tap the calculator to display the arithmetic operators. Use the displayed fields to calculate your values.

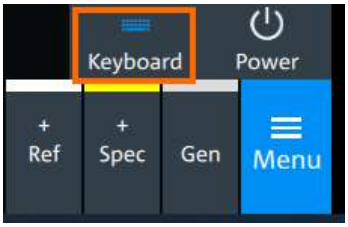



3. To activate caps lock when you enter a text value, double-tap the shift key (up arrow key).

4. Tap ↵ to complete the entry.

**To enable the on-screen keyboard**

1. Tap "Menu".
2. Tap on "Keyboard" to enable the on-screen keyboard.




If the on-screen keyboard is enabled, the keyboard icon is colored blue. If it is disabled, the color is white.

**To enter data with the on-screen keyboard**

1. Double-tap the entry field to open the on-screen keyboard.




2. Enter the text as you would on a normal keyboard.
   - To change the language, tap 🌐. The current language is shown on the blank key.
   - To display numbers and signs, tap &123.
3. Tap ↵ to complete the entry or ⌨ to minimize the keyboard.

## 4.11 Displaying results

The results of measurements, protocol decoding and others are displayed immediately.

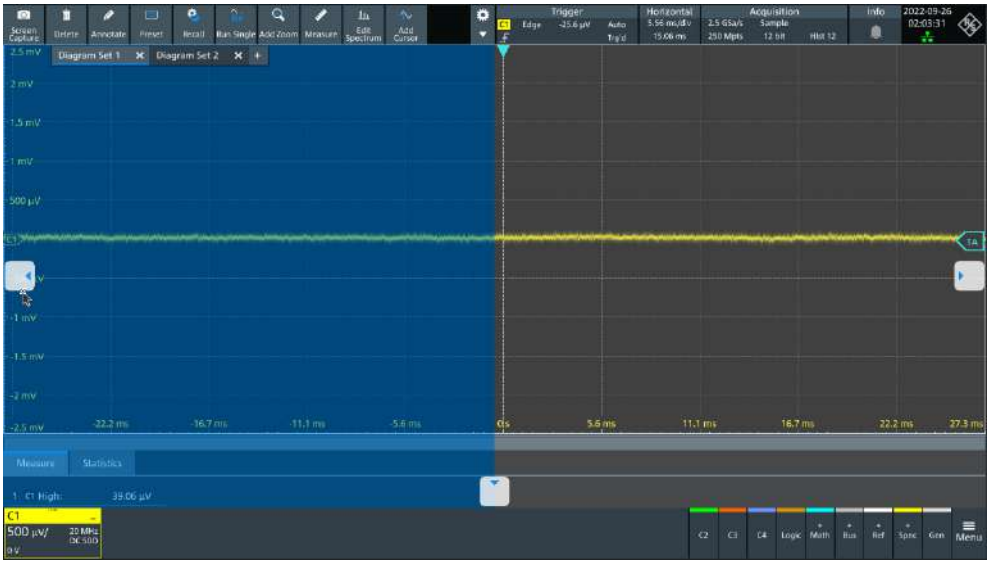
The font size can be adjusted.

**To arrange the results on the display**

1. Touch and hold the "....." field on top of the results table.




2. Drag on the screen. The SmartGrid indicates where the result table can be placed. Drop the box on one of the buttons. The results are shown at the left the right, or below the diagrams.



**To open the corresponding settings**

► Double-tap one of the result values.

The corresponding dialog opens.

**To adjust the font size of results**

1. Open the "Menu" > "Settings" > "Appearance" dialog.
2. Select the "Dialogs" tab.
3. Set the "Result dialog" > "Font size".

## 4.12 Adding annotations

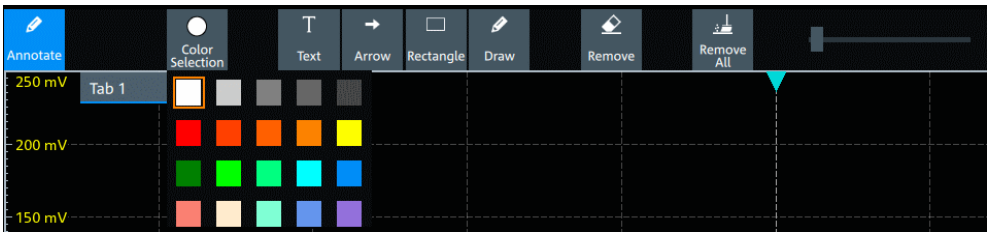
The MXO 3 provides an easy way to add annotations to the screen. With the toolbar "Annotate" you can add text, forms or even draw.

**To add an annotation**

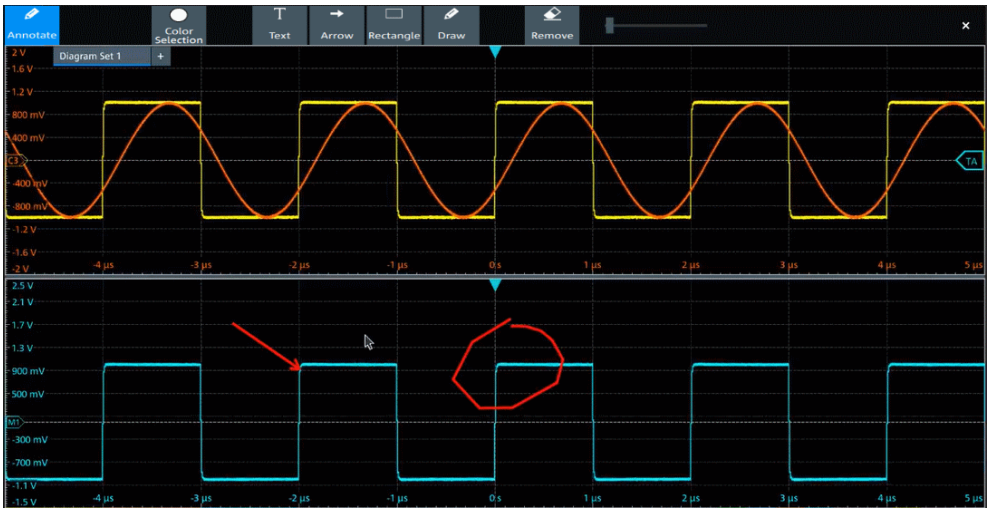
1. On the toolbar, tap on the "Annotate" icon.




The annotation overlay menu opens.




2. Tap "Color selection" and select the color that you want to use for your annotations.
3. Add one or more of the following:
   - One of the predefined forms: "Arrow" or "Rectangle"
   - "Text": adds a text element to the screen and opens on-screen keyboard. You can change the text later, by tapping on an existing annotation and typing in the new text.
   - "Draw": you can draw any form on the screen.




4. To move one of the annotations, tap on it and drag it to the required position.

**To remove an annotation**

1. In the "Annotate" overlay menu, tap on "Remove".
2. Tap on any existing "Arrow", "Rectangle" or "Text" annotation to remove it.
3. The "Remove" function, acts as an eraser on drawings: it enables you to erase only certain parts of your drawing. Drag your finger over any part of your drawing to erase it.
4. To remove all annotations from the screen, tap on "Remove all".

Remote commands: see Section 18.8.5, "Annotations", on page 964.

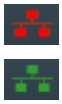
## 4.13 Information and notifications

The MXO 3 provides various information:

- Instrument information, e.g. connection status, firmware version
- Notifications, e.g. status messages, incompatible settings, insuffcient memory
- Progress information, e.g. for long acquisitions, data saving

### Instrument information

In the upper right corner of the screen, you see the Rohde & Schwarz logo, date and time, the symbolic information on LAN connection and the notifications status.



The connection status icon in the upper right corner indicates the status of the LAN connection.

- Green indicates that the instrument is connected to the LAN.
- Red indicates a connection error - mostly the LAN cable is not connected.
- Blinking indicates that the device indicator on the instrument's homepage is switched on, and a device identification command was sent.

► To see the instrument information, select the Rohde & Schwarz logo.

You can find information about the instrument model, bandwidth, network settings, firmware version and operating time.

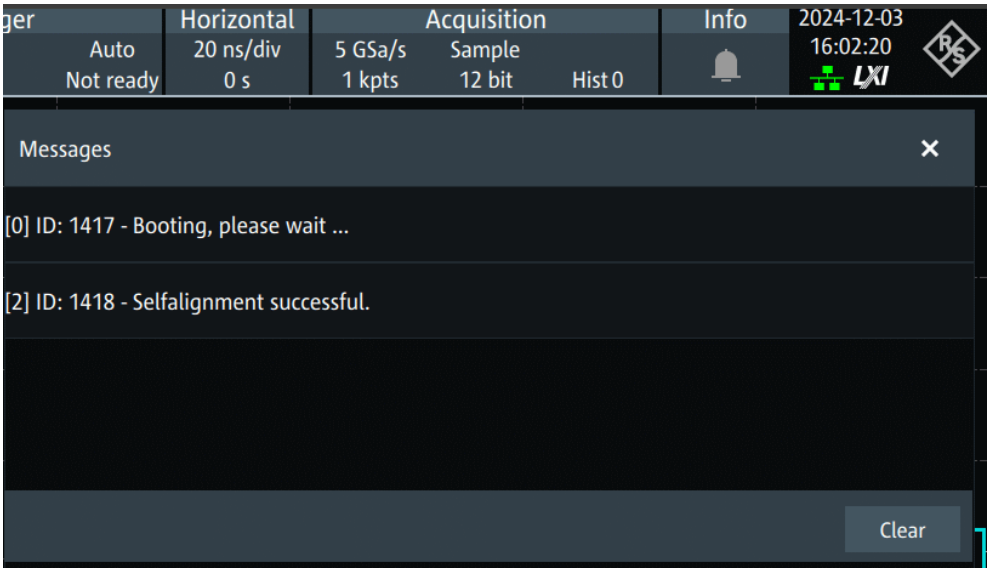
► To hide the date and time or change the display format, select the date/time display.

### Notifications

Notifications are status messages, information on mismatching settings, insufficient memory and similar information. They are displayed for a few seconds and saved.

► To read the notifications, select "Info".

You can delete the list of notifications.




**Progress information**

If an acquisition, operation or process takes more time until results are displayed, a progress bar or symbol indicates that a process is running.

The progress of saving operations and update or creation of a reference waveform is shown in a progress bar. You can cancel the process. A notification informs about the result.




*Figure 4-9: Progress bar for saving waveform data*

The progress of running acquisitions is shown in the acquisition label. A circle indicates the state of the current, incomplete acquisition. Progress in roll mode, average mode and single acquisition with N > 1 is indicated by a progress bar and counter.




*Figure 4-10: Progress of incomplete long acquisition in average mode, Run single = 5, first acquisition is finished*

## 4.14 Getting user assistance

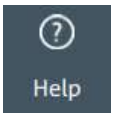
If you need information on the functionality, you can use the integrated product help. It provides contextual information on a setting or dialog. If the help window is open,

you can browse and search for further information using links, table of contents, and search.

### 4.14.1 Displaying help

**To display the context help**

1. Enable the "Help" icon on the toolbar.




2. Tap the parameter for which you need information.

   The "Help" window opens and displays the comprehensive description and the corresponding remote command. You can browse the help for further information.

**To open a dialog help**

1. Open a dialog.
2. Tap the "Help" icon in the dialog header.
3. Tap a subtab or menu item.

   The help window opens with the dialog help page, where you can select the topics.

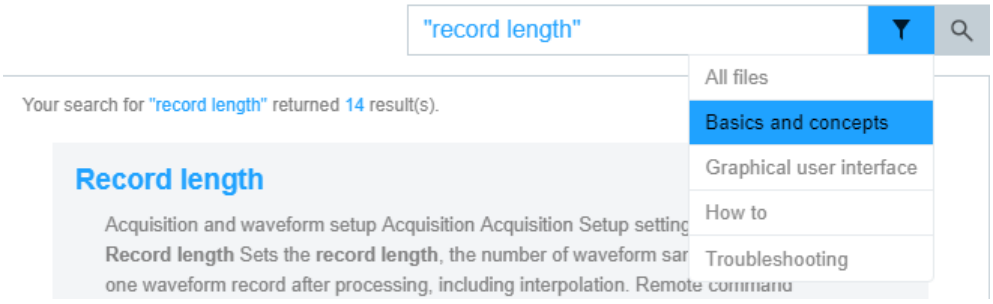
### 4.14.2 Using help

You can find a specific information and navigate the contents by following means:

- Table of contents
- Buttons in the help window title:



  – "Home": Go to the start page
  – "Back", "Forward": Browse the pages that you visited before
- Search with filter:

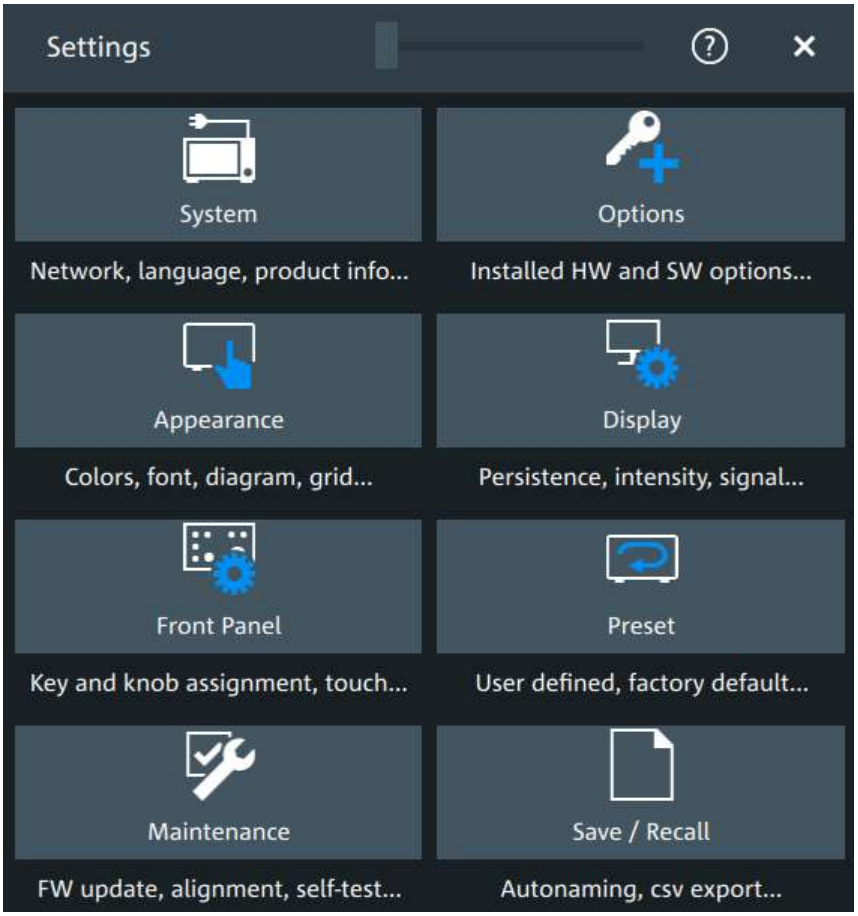


– Enter the word to be found, or a phrase in quotes.
– Tap the filter icon and select the information type.

# 5 Instrument setup

Access: "Menu" > "Settings".

In the "Settings" dialog, you can adapt various instrument settings to your requirements, such as language, display appearance, and assign functions to some keys.




The following settings and procedures are described in the current section:

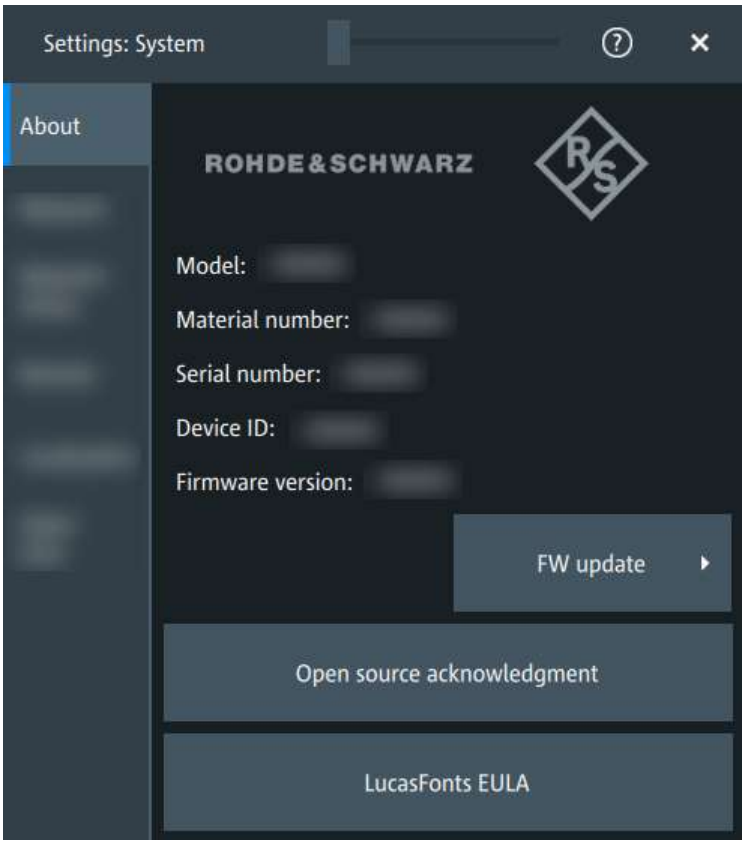

- System settings ..... 84
- Option settings ..... 93
- Appearance settings ..... 96
- Display settings ..... 104
- Front panel settings ..... 107
- Preset setup ..... 109
- Maintenance settings ..... 112
- Save / recall settings ..... 117

# 5.1 System settings

In the "Settings" > "System" dialog box, you find all instrument, firmware and network-related information. Here you can also set the language that is used in the dialogs.

## 5.1.1 About settings

Access: "Menu" > "Settings" > "System" > "About".




**Instrument**
Displays general information about the instrument, including:

- "Material number"
- "Serial number"
- "Device ID"
- "Firmware version"

Remote command:
`*IDN?` on page 947

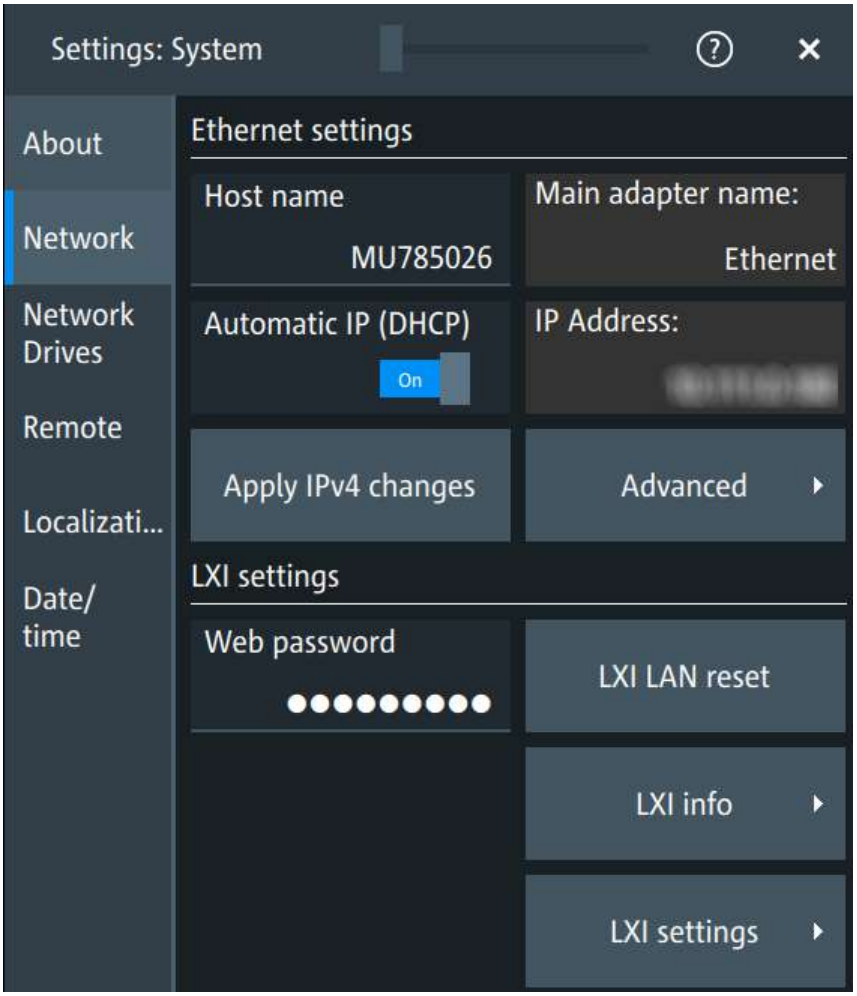
**FW update**
Opens the "FW update" dialog box. See Section 5.7.1, "Firmware update", on page 112.

**Open source acknowledgment**
Displays the "Open Source Acknowledgment" document.

**LucasFonts EULA**
Displays the copyright of the font provider.

### 5.1.2 Network settings

Access: "Menu" > "Settings" > "System" > "Network".




**Host name**
Indicates the currently defined host name. This value is required to configure the instrument for work in a network.

You can change the host name here. After changing the host name, you have to reboot the instrument.

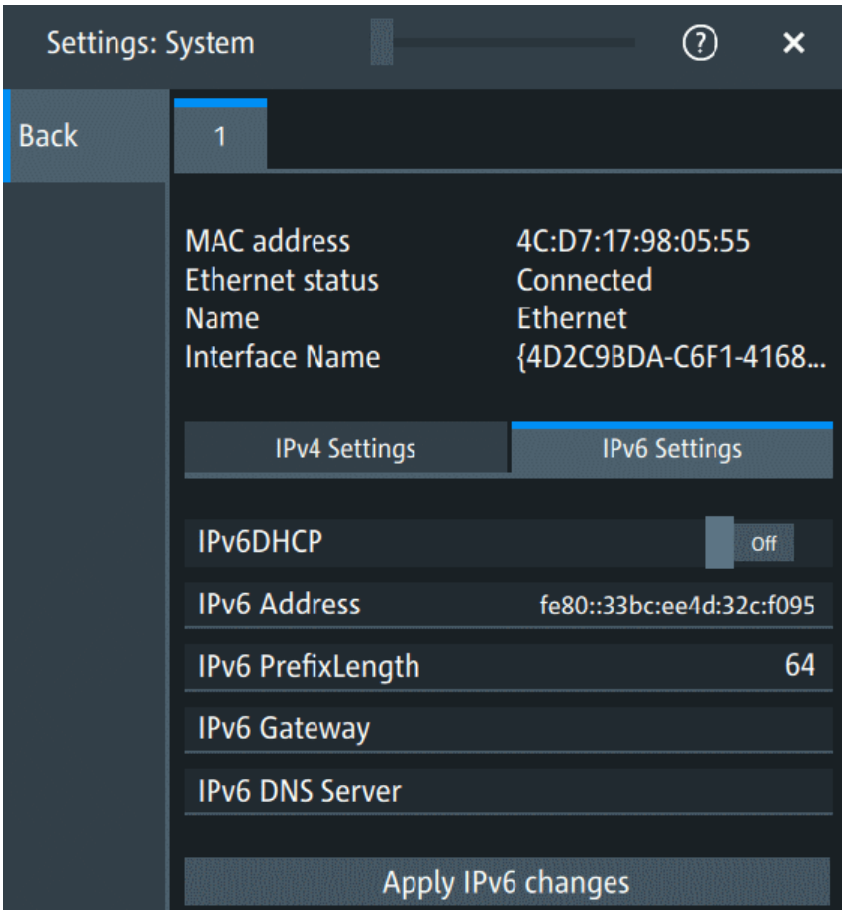
**NOTICE!** Risk of network problems. Incorrect configuration of the host name can create network problems. Consult your network administrator before changing the name.

Remote command:
SYSTem:COMMunicate:NET[:HOSTname] on page 953

**Automatic IP (DHCP)**
If enabled, the IP address of the oscilloscope is obtained automatically.

**IP address**
Shows the current IP address of the instrument. If "Automatic IP (DHCP)" is "Off", you can change the IP address here. For complete manual configuration, select "Advanced". See also: Section 17.1, "Connecting the instrument to the network (LAN)", on page 915.

**Advanced**
Opens a dialog box to configure the network connection.




You need these settings to configure the LAN connection manually, without DHCP.

**NOTICE!** Connection errors can affect the entire network. Make sure to assign valid address information before connecting the instrument to the LAN. Contact your network administrator to obtain a valid IP address and other connection data.

See also: Section 17.1, "Connecting the instrument to the network (LAN)", on page 915.

**Web password**
Password for LAN configuration. The default password is *LxiWebIfc*.

**LXI LAN reset**
Resets the LAN configuration to its default configuration. Use the reset when you cannot access the instrument via LAN due to invalid configuration.

The LAN settings are configured in the "Advanced" dialog, or using the instrument's web browser (see Section 17.2, "Web interface", on page 917).

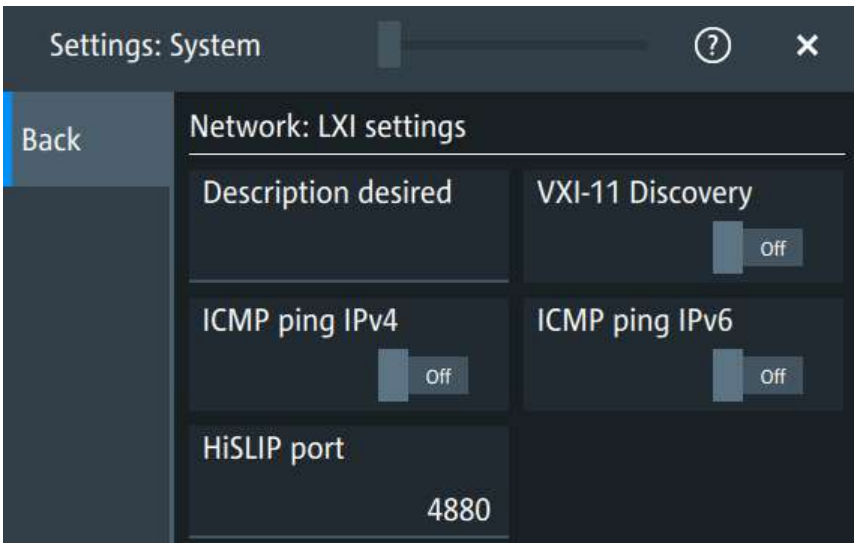
The LAN reset uses the LAN Configuration Initialize (LCI) reset mechanism for the instrument. According to the LXI standard, a LAN reset must set the following network parameters to a default state:

| Parameter | Value |
|---|---|
| TCP/IP mode | DHCP + auto IP address |
| Dynamic DNS | Enabled |
| ICMP ping | Enabled |
| Password for LAN configuration | LxiWebIfc |

**LXI info**
Opens a dialog with relevant LXI information. You can reload the information, for example, after changing connection data in the instrument's web browser.

For information on the LXI standard, refer to the LXI website at https://www.lxistandard.org.

**LXI settings**
Opens a dialog with important LXI settings. These settings are also available in the instrument's web browser.




**Description desired ← LXI settings**
Network description of the instrument.

**VXI-11 Discovery ← LXI settings**
Enables the automatic detection in a LAN via the VXI-11 discovery protocol and programming by IVI drivers.

**ICMP ping IPv4, ICMP ping IPv6 ← LXI settings**
Enables the ping utility to verify the connection between the LXI-compliant instrument and another device. Ping is useful for diagnosing IP network or router failures. The ping utility is not password-protected.

**HiSLIP port ← LXI settings**
Sets the port that is used by the High Speed LAN Instrument Protocol (HiSLIP).

## 5.1.3 Network drive mapping

You can map up to 8 network drives or shared folders to the MXO 3 and store the measurement data directly in one of these folders.

Requirements:

- The MXO 3 and the target folder must be in the same network. Connection is established via DHCP. Direct connection is not supported.
- You need administrative rights and credentials to access the folder.
- Make sure that the network connection is not restricted by security settings.

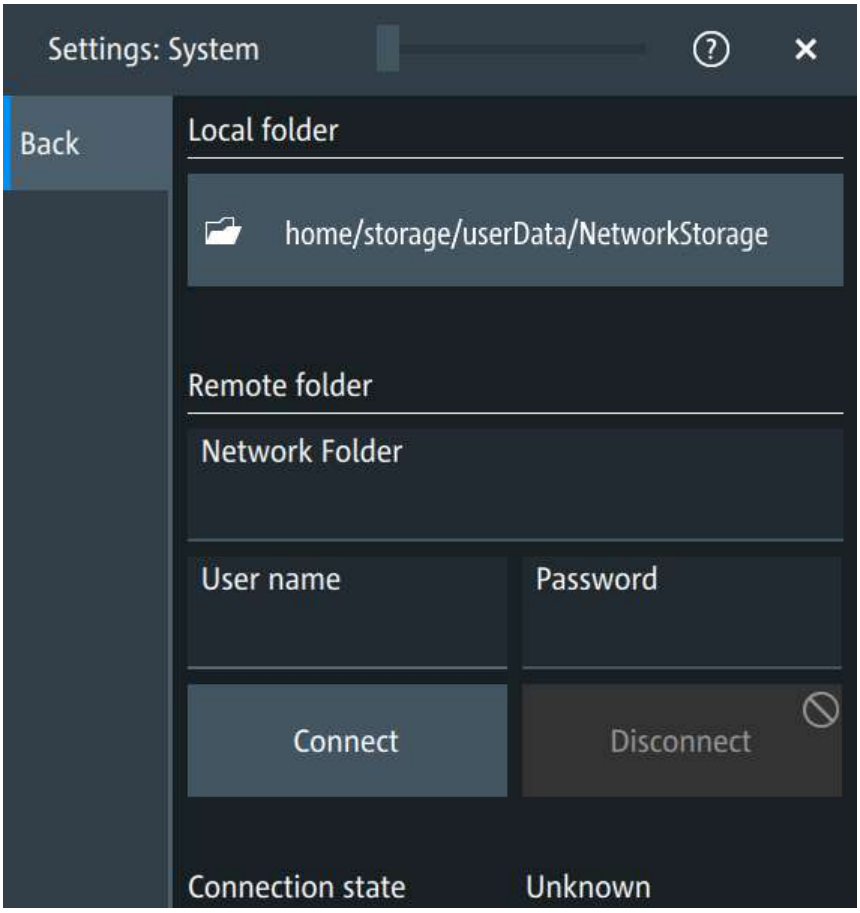
### 5.1.3.1 Connecting a network drive

1. Open "Menu" > "Settings" > "System" > "Network drives".
2. Select "+ Add".
3. Tap the "Empty" button.
4. Enter the "Local folder" on the MXO 3, the "Network folder", and access credentials. For details, see Section 5.1.3.2, "Network drive settings", on page 88.
5. Tap "Connect".

If you want to connect a folder on your personal computer, share this folder and give full permissions to all users before you connect the folder to the MXO 3: Folder properties > "Sharing" > "Advanced Sharing".

### 5.1.3.2 Network drive settings

Access: "Menu" > "Settings" > "System" > "Network drives" > "+ Add" > select drive




**Local folder**
Set the local folder on the MXO 3, to which the network folder is mapped. Under this name, you find the connected folder in the file browser on the instrument and in the "File manager" of the web interface.

Remote command:
`MMEMory:NDRive<m>:LPATh` on page 956

**Network folder**
Sets the network folder that is mapped to the local folder. Use the full path in UNC syntax, for example, `\\net1\data\team1\osc`.

Remote command:
`MMEMory:NDRive<m>:NPATh` on page 956

**User name, Password**
Enter the user name and the password to access the network folder.

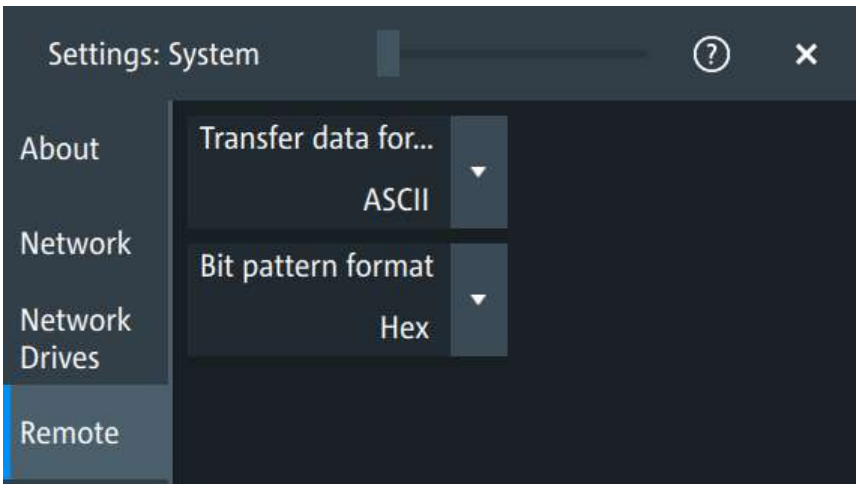
Remote command:
`MMEMory:NDRive<m>:UNAMe` on page 957
`MMEMory:NDRive<m>:PWORd` on page 956

**Connect, Disconnect, Connection state**
Connect or disconnects the remote network folder to the local folder. An "Info" message informs on success or failure, and the overall connection state is shown in the dialog.

Remote command:
`MMEMory:NDRive<m>:CONNect` on page 955
`MMEMory:NDRive<m>:DCONnect` on page 956

### 5.1.4 Remote settings

Access: "Menu" > "Settings" > "System" > "Remote".




The following settings are required for remote control and data transfer via a connected computer.

**Transfer data format**
Selects the data format that is used for transmission of waveform data from the MXO 3 to the controlling computer.

If you need physical data (e.g. in Volt or Ampere) for further analysis, use the floating point formats for export. No data conversion is needed.

| | |
|---|---|
| "Ascii" | Data values are returned in ASCII format as a list of comma-separated values in floating point format. |
| "FLOAT" | Binary format. Up to 7 significant digits are stored (32 bit). If there are more than 7 digits, the number is rounded off. |
| "DOUBLE" | Binary format. Up to 15 significant digits of the numbers (64 bit) are stored. |
| "INT8/16/32" | Signed integer data with length 8/16/ 32 bit. |

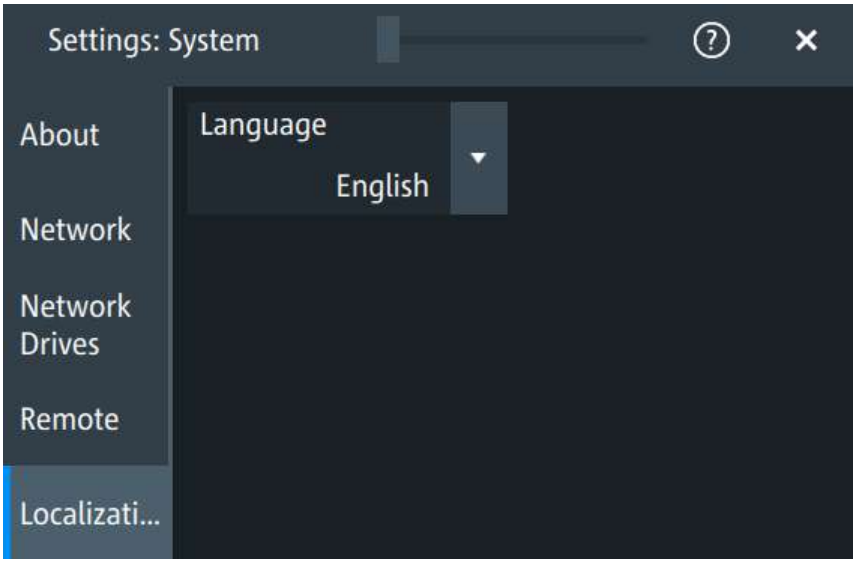
Remote command:
`FORMat[:DATA]` on page 950

**Bit pattern format**
Sets the format for all bit pattern queries.

Remote command:
FORMat:BPATtern on page 952

### 5.1.5 Localization settings

Access: "Menu" > "Settings" > "System" > "Localization".

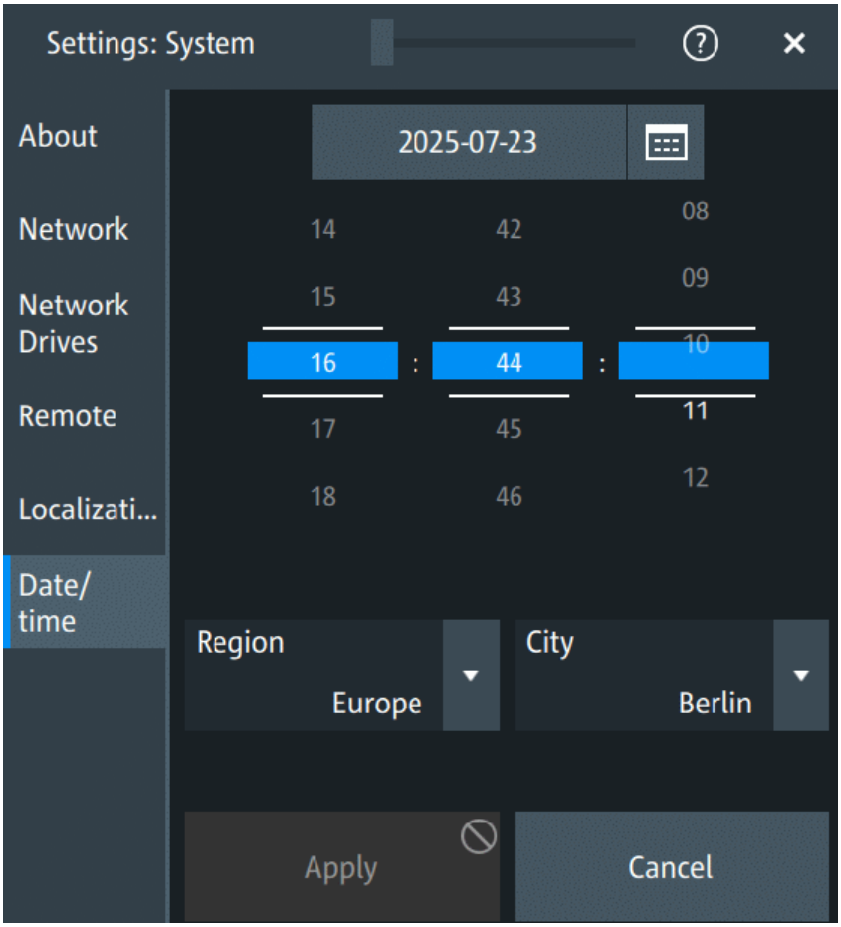



**Language**
Selects the language in which the menu, dialogs, results and other screen information is displayed. You can change the language on runtime.

Available languages are, for example, English, German, French, Japanese. For a complete list of supported languages, refer to the specifications document.

### 5.1.6 Date and time settings

Access: "Menu" > "Settings" > "System" > "Date/time".




The instrument uses the system time of the operating system, but you can change date and time for the MXO 3 firmware.

1. To change the date:
   a) Tap the date button.
   b) Select the date in the calendar.
   c) Tap the blue calendar icon to close the calendar.
2. To change the time, move the hour, minute and second sliders to the required values.
3. Adjust the "Region" and "City" to indicate your location.
4. Tap "Apply".

Remote command:

- `SYSTem:DATE` on page 953
- `SYSTem:TIME` on page 955

## 5.2 Option settings

Additional options for the MXO 3 are enabled using a license key. To obtain the license key, consult your sales representative.

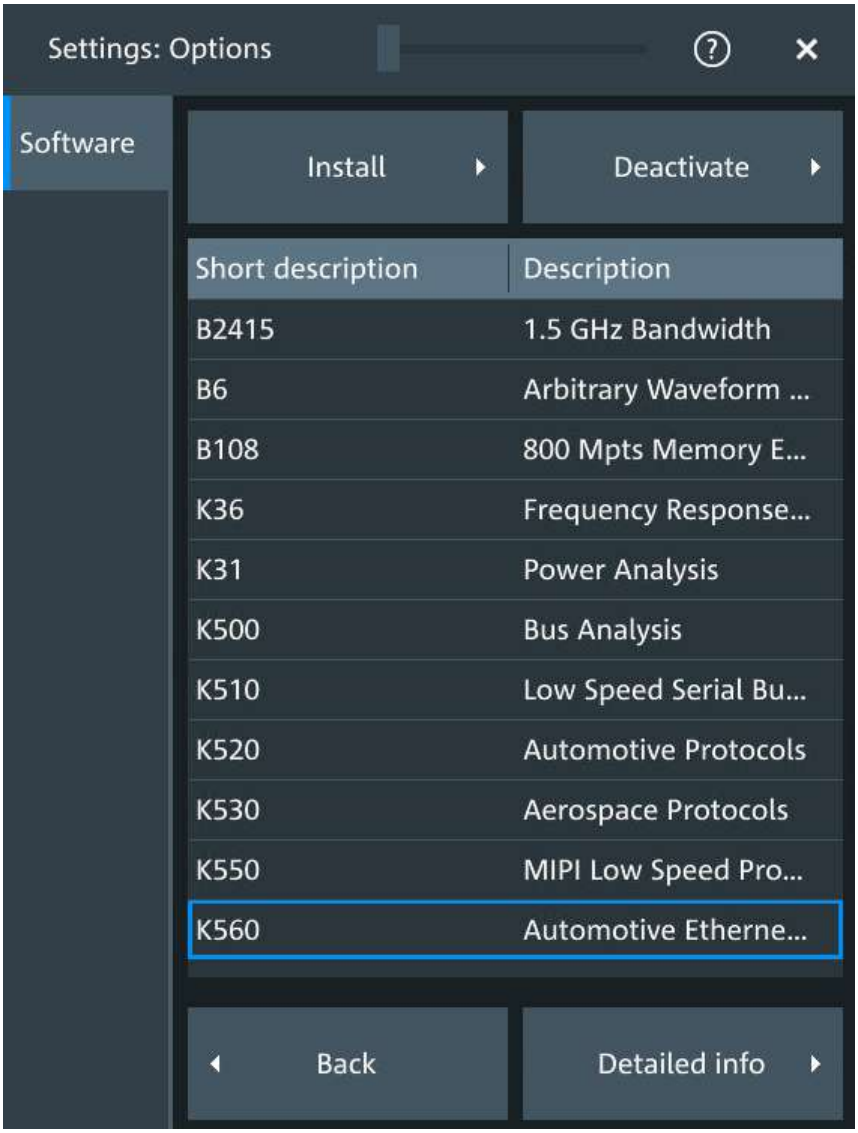
You can obtain registered or unregistered licenses.

**Unregistered licenses**

Unregistered licenses are not assigned to a particular instrument. The instrument accepts only registered licenses. If your license is delivered unregistered, use the online tool R&S License Manager to register the license for your instrument. The registration of a permanent license is irreversible, so ensure that you register it for the correct instrument. For registration, you need the device ID of the instrument on which the option will be installed.

### 5.2.1 Software options settings

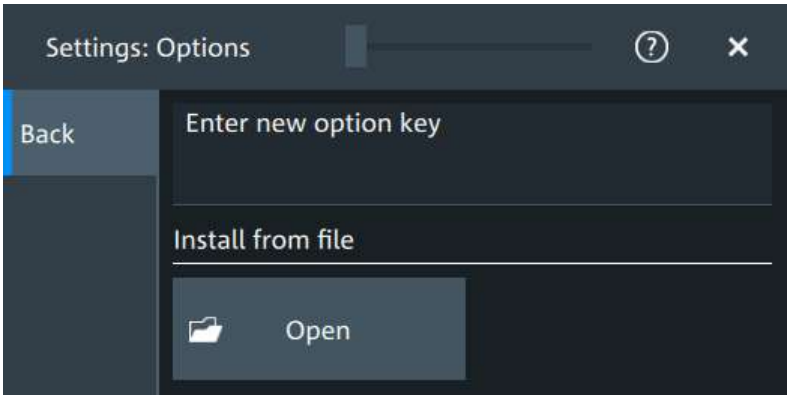
Access: "Settings" > "Options" > "Software"




In this dialog, you can access settings for installing and deactivating options. In the "Detailed info" dialog, you can get an overview of all options installed on your MXO 3.

#### 5.2.1.1 Install options

Access: "Settings" > "Options" > "Software" > "Install"




In the "Install" tab, you can install new options using license keys.

**Enter new option key**
Enter the license key here to activate the option. For license keys delivered as a file, use "Install from file" on page 95.

**Install from file**
If you got a license file, install the license here.

Tap "Open" to open the file selection dialog, or enter the complete path and filename.

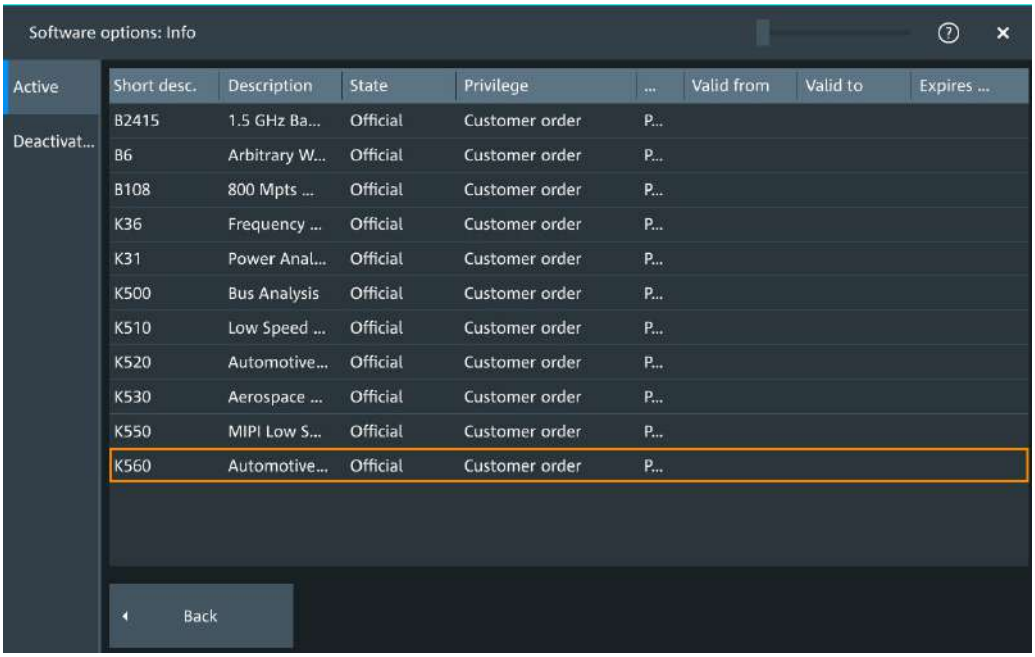
#### 5.2.1.2 Deactivate options

Access: "Settings" > "Options" > "Software" > "Deactivate"

To deactivate an option, use the "Deactivate" dialog. When you deactivate, note the response key, or save the response to a file. Registered options cannot be deactivated, they are bound to the instrument.

#### 5.2.1.3 Detailed info

In the "Detailed info" dialog, you can get an overview of all options installed on your MXO 3 with status and expire information.




## 5.3 Appearance settings

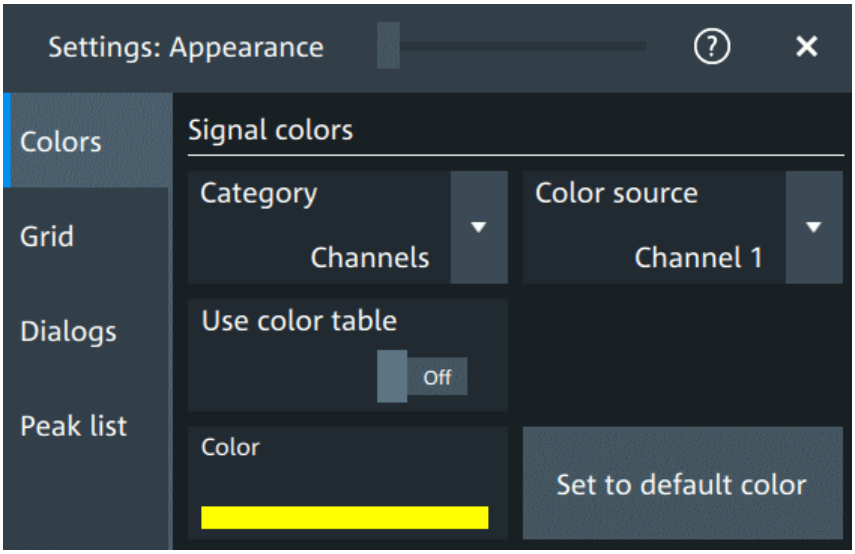
In the "Settings" > "Appearance" dialog box, you define the look and feel of the display element, e.g. waveform colors, result position, or grid behavior.

Appearance settings are not included in user-defined presets. To reset all appearance settings at once, use the factory preset.

### 5.3.1 Colors

Access: "Settings" > "Appearance" > "Colors"

By default, various colors are assigned to the different waveform types for better visibility and distinguishability. You can change the color assignment and assign another color or a color table to waveforms. For digital channels, you can select any color, or use the default color with state coloring.




**Category**
Selects the category of the waveform, e.g. "Channels", "Math", "Reference".

You can then select one of the waveforms of the selected category with "Color source".

**Color source**
Selects the waveform to which the color or the color table is assigned.

**Use color table**
If enabled, the selected waveform is displayed according to its assigned color table.

If disabled, the selected color is displayed, and the intensity of the signal color varies according to the cumulative occurrence of the values.

The setting is not available for digital channels and parallel buses.

Remote command:
`DISPlay:COLor:SIGNal:USE` on page 977

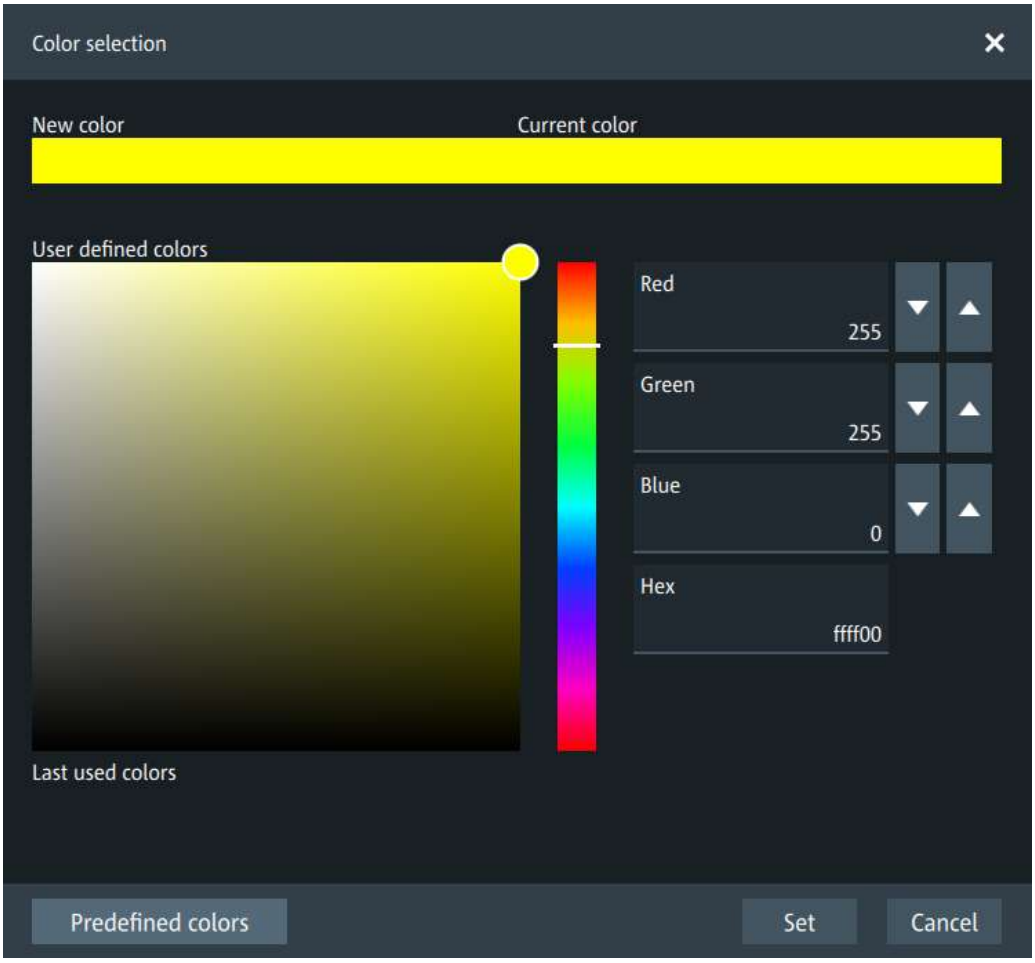
**Assigned color table**
Assigns a color table to the source waveform instead of a dedicated color. Color tables define the color of the waveform pixels depending on the cumulative occurrence of the associated values.

The following color tables are provided:
- "False colors": color changes gradually in a wide color spectrum.
- "Single Event": single events and seldom events appear yellow, a higher cumulative occurrence is shown with blue color. This view helps to identify specific events.
- "Spectrum": colors display the wavelengths of the light. Low cumulative occurrence is displayed blue like high wavelength.
- "Temperature": color changes gradually from blue (low temperature) to red (high temperature) with increasing cumulative occurrence.

Remote command:
`DISPlay:COLor:SIGNal:ASSign` on page 977

**Color**
Shows the current color of the selected waveform. To change the color, tap the button.

In the "Color" dialog, you can pick from a list of basic colors. To define another color, select "User defined colors", and adjust the color settings.




The color of the waveform, its signal icon, and the illuminated keys are adjusted to the new color. If the waveform is a spectrum, the knobs for vertical scale and position retain the color of the spectrum's source.

Remote command:
`DISPlay:COLor:SIGNal:COLor` on page 977

**Set to default color**
Resets the color of the selected waveform to the factory default.

Remote command:
`DISPlay:COLor:SIGNal:COLor <Signal>,DEF`

**Use state coloring**
The setting is available for digital channels and logic buses.

If enabled, the high and low levels and the state transition get different predefined colors.

If disabled, you can select a color for the digital channel or bus. The channel is displayed with the selected color, and the high level has a thicker line than the low level.

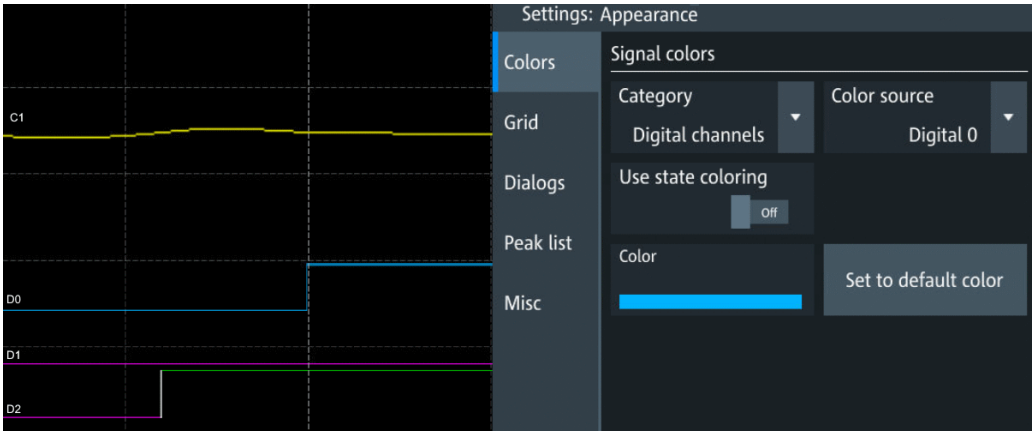



*Figure 5-1: D0 has a selected color (blue). D1 and D2 have state coloring. D1 is on low level, D2 changes from low to high.*

## 5.3.2 Grid

Access: "Settings" > "Appearance" > "Grid"



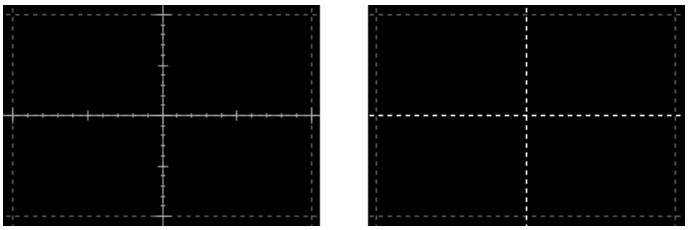
**Show grid**
If selected, a grid is displayed in the diagram area. A grid helps you associate a specific data point to its exact value on the x- or y-axis.

Remote command:
`DISPlay:DIAGram:GRID` on page 978

**Show labels**
If selected, labels mark values on the x- and y-axes in specified intervals in the diagram.

Remote command:
`DISPlay:DIAGram:LABels` on page 979

**Show fine grid scale**
If selected, the crosshair is displayed as a ruler with scale markers. If disabled, the crosshair is shown as dashed lines.



Remote command:
`DISPlay:DIAGram:FINegrid` on page 978

**Show crosshair**
If selected, a crosshair is displayed in the diagram area. A crosshair allows you to select a specific data point by its coordinates.

Remote command:
`DISPlay:DIAGram:CROSshair` on page 978

**Keep X-grid fixed**
If enabled, the vertical grid lines remain in their position when the horizontal position is changed. Only the values at the grid lines are adapted.

Remote command:
`DISPlay:DIAGram:XFIXed` on page 979

**Keep Y-grid fixed**
If enabled, the horizontal grid lines remain in their position when the position of the curve is changed. Only the values at the grid lines are adapted.

Fixed horizontal grid lines correspond to the behavior of traditional oscilloscopes.

Remote command:
`DISPlay:DIAGram:YFIXed` on page 979

**Show div label (x)**
If selected, the time scale value is shown at the diagram bottom instead of the horizontal grid labels. For example, 10 ns/div is shown instead of the values 0, 10, 20, 30... ns.

Remote command:
DISPlay:DIAGram:XPDivlabel on page 979

**Channel label**
Select the information to be shown in the waveform labels at the waveform.

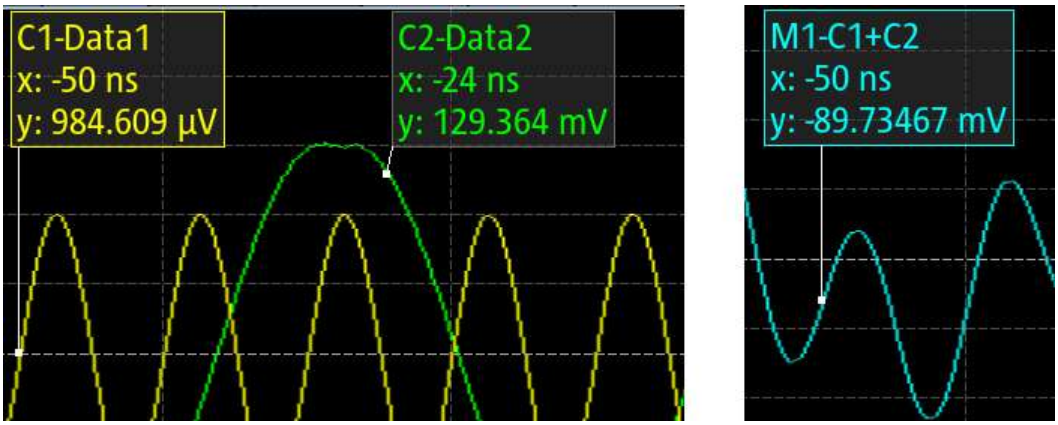
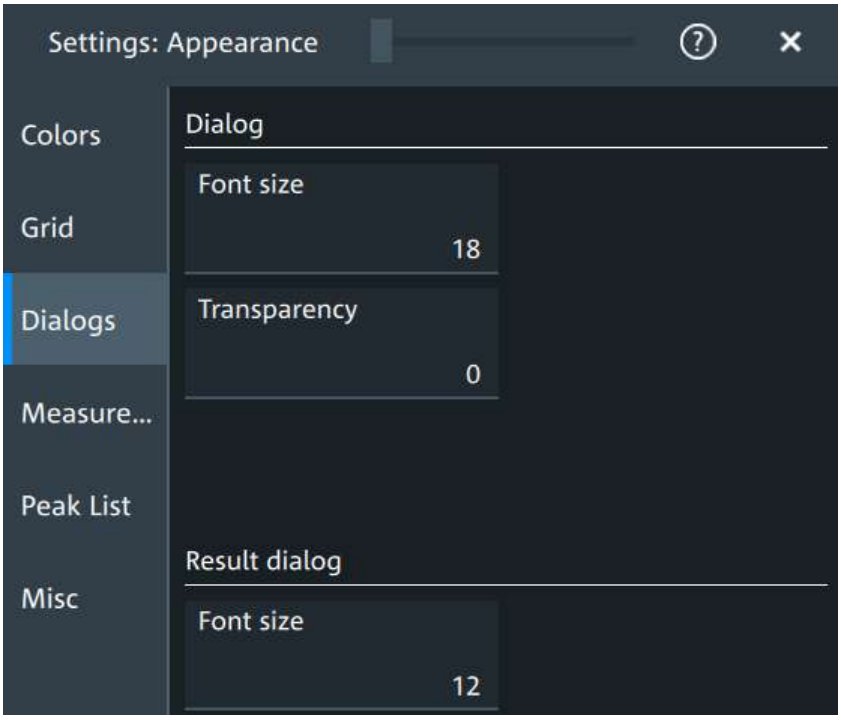



*Figure 5-2: Waveform labels with complete information, for analog channels and math waveform*

| | |
|---|---|
| "Show signal shortcut" | Shows the generic short name of the waveform, e.g, C1, C2, R1, M1. |
| "Show signal label" | Shows the user-defined label. These labels are defined in the setup dialog of the waveform, e.g. in the "Vertical" > "Setup" tab for analog channels, and in the "Math" > "Setup" tab for math waveforms. |
| "Show X position" | Shows the x-position of the attachment point of the waveform label to the waveform. |
| "Show Yposition" | Shows the y-position of the attachment point of the waveform label to the waveform. |

### 5.3.3 Dialogs

Access: "Settings" > "Appearance" > "Dialogs"

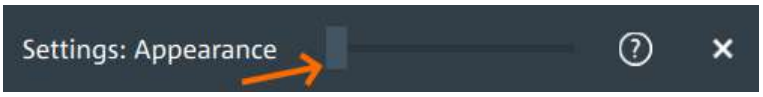



**Font size (Dialog)**
Sets the font size of the text in dialog boxes.

Remote command:
`DISPlay:DIALog:FONTsize` on page 980

**Transparency (Dialog)**
Sets the transparency of the dialog box background. For high transparency values, you can see the waveform display in the background, and possibly check the effect of the changed setting. For lower transparency values, readability in the dialog box improves.

You can also set the dialog transparency by moving the transparency bar at the top of the dialog.



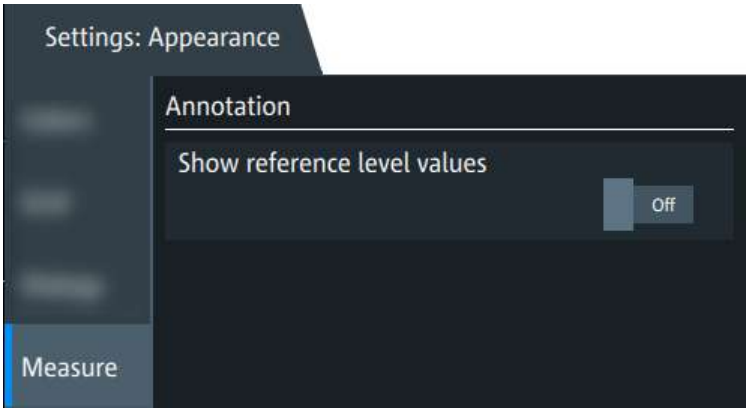

Remote command:
`DISPlay:DIALog:TRANsparency` on page 980

**Font size (Result dialog)**
Sets the font size of the text in result tables.

Remote command:
`DISPlay:RESult:FONTsize` on page 980

## 5.3.4 Measure

Access: "Settings" > "Appearance" > "Measure"

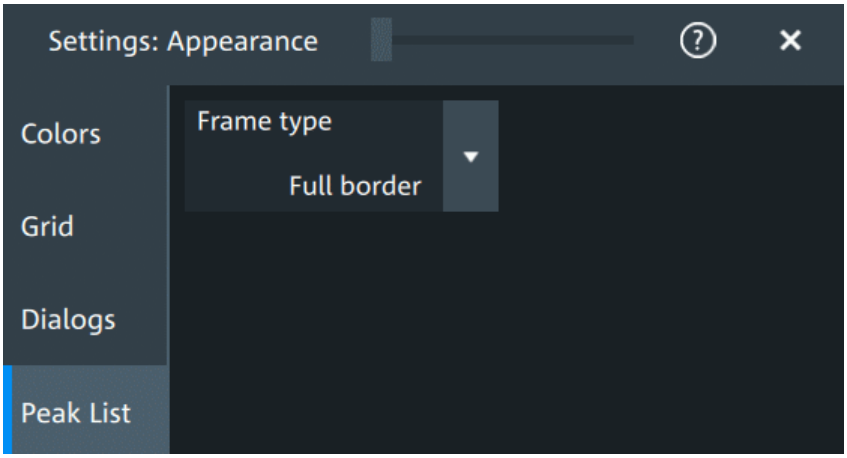



**Show reference level values**
If selected, the values of the reference levels are displayed.

Remote command:
`DISPlay:MEASurements:LEVels` on page 981

### 5.3.5 Peak list

Access: "Settings" > "Appearance" > "Peak list"



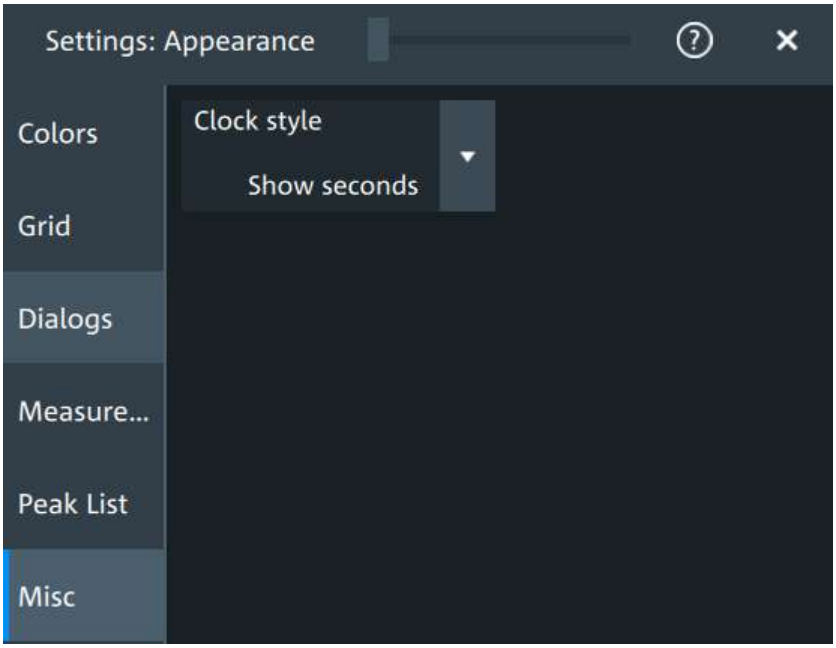

Peak list settings apply to peak list measurements. They are visible when the peak list for spectrum measurements is enabled.

**Frame type**
Defines the layout of the labels, full border or none.

Remote command:
`CALCulate:SPECtrum<sp>:PLISt:LABel:BORDer` on page 981

### 5.3.6 Miscellaneous

Access: "Settings" > "Appearance" > "Misc"




**Clock style**
Select how the clock in the upper right corner of the screen is displayed:



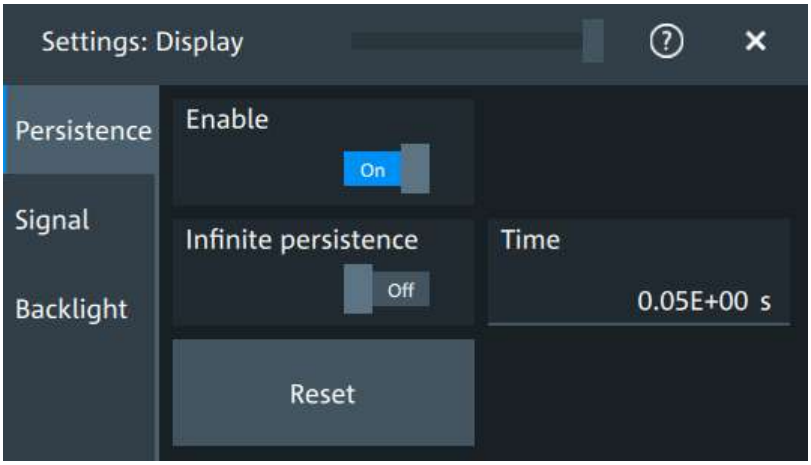

- "Show seconds": format is hh:mm:ss
- "Hide seconds": format is hh:mm
- "Hide clock": no time is shown

## 5.4 Display settings

In the "Settings" > "Display" dialog, you can define the display settings like brightness and signal intensity.

### 5.4.1 Persistence settings

Access: "Menu" > "Settings" > "Display" > "Persistence".




**Enable**
If enabled, each new data point in the diagram area remains on the screen for the duration that is defined using Time, or as long as Infinite persistence is enabled.

If disabled, the waveform points are displayed only for the current acquisition.

Remote command:
`DISPlay:PERSistence[:STATe]` on page 982

**Infinite persistence**
If infinite persistence is enabled, each new waveform point remains on the screen until this option is disabled. Use infinite persistence to display rare events in the signal.

Remote command:
`DISPlay:PERSistence:INFinite` on page 982

**Time**
Sets a time factor that controls how long the waveforms points fade away from the display. Thus, the MXO 3 emulates the persistence of analog phosphor screens.

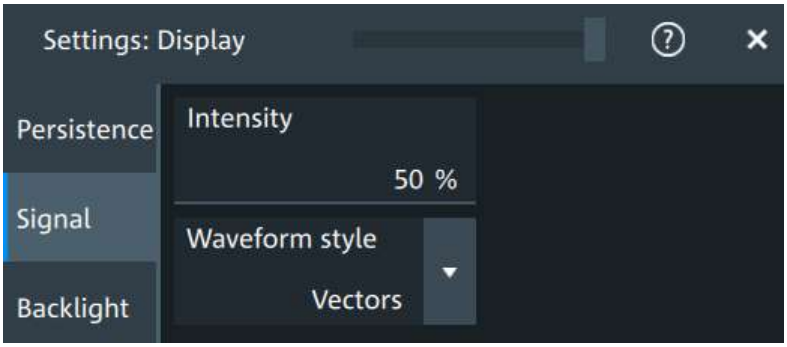
Remote command:
`DISPlay:PERSistence:TIME` on page 982

**Reset**
Resets the display, removing all persistent waveform points.

Remote command:
`DISPlay:PERSistence:RESet` on page 982

## 5.4.2 Signal settings

Access: "Menu" > "Settings" > "Display" > "Signal".




**Intensity**
The intensity determines the strength of the waveform line in the diagram. Enter a percentage between 0 (not visible) and 100% (strong). The default value is 50%.

You can also use the [Intensity] knob to adjust the waveform intensity directly.

Remote command:
`DISPlay:INTensity` on page 983

**Waveform style**
Selects the style in which the waveform is displayed.

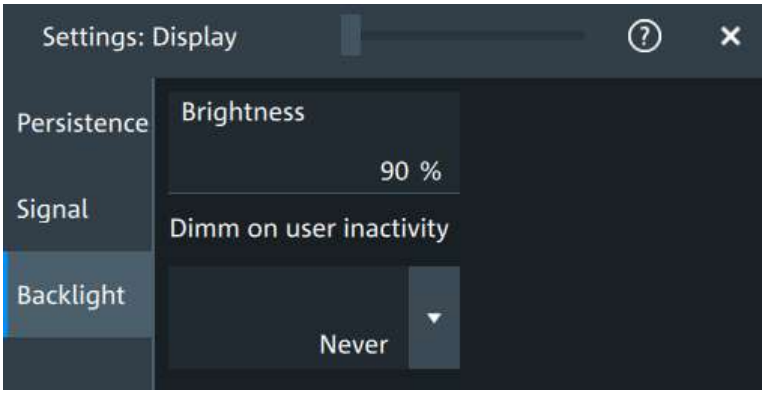
"Vectors" The individual waveform points are connected by a line.
Define the strength of the line using the [Intensity] knob.

"Dots" Only the individual waveform points are displayed.
To see the dots of one waveform, perform one acquisition with [Single] and N=1 (N-single/avg count). During continuous acquisition, or a [Single] acquisition with N > 1, the dots of multiple subsequent waveforms are displayed on the screen, and the waveform looks like a line.

Remote command:
`DISPlay:DIAGram:STYLe` on page 983

## 5.4.3 Backlight settings

Access: "Menu" > "Settings" > "Display" > "Backlight".



**Brightness**
Sets the background luminosity of the touchscreen.

Remote command:
`DISPlay:BACKlight[:BRIGtness]` on page 983

**Dimm on user inactivity**
Selects a time, after which the monitor brightness is reduced, if the instrument was inactive. Remote control of the instrument is also considered as an activity.

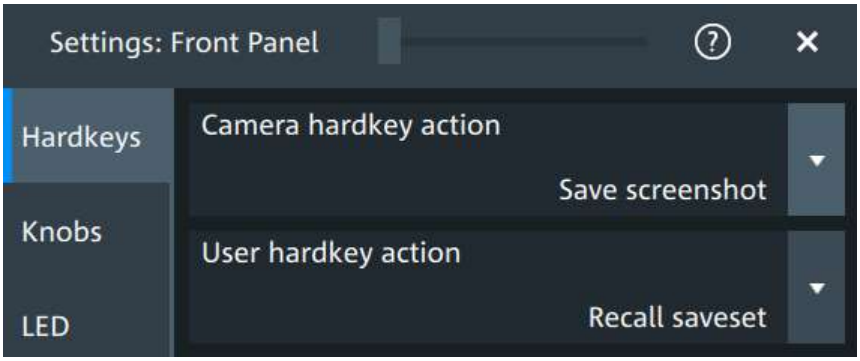
Remote command:
`DISPlay:BACKlight:DIMMing` on page 984

## 5.5 Front panel settings

In the "Front panel" dialog box, you can assign functions to keys and knobs and adjust the brightness of the keys.

### 5.5.1 Hardkeys: function assignment

Access: "Settings" > "Front panel" > "Hardkeys"




**Camera hardkey action**
The Camera key on the bottom right is a shortcut key that initiates an associated action.

You can assign one of the following actions:
- Save screenshot
- Open screenshot setup

Configure the settings for the selected action.
- Screenshots: "Save/recall" key > "Save" tab > "Screenshot", see Section 13.8, "Screenshots", on page 481.

**User hardkey action**
The [User] key below the spectrum keys is a shortcut key that initiates an associated action.

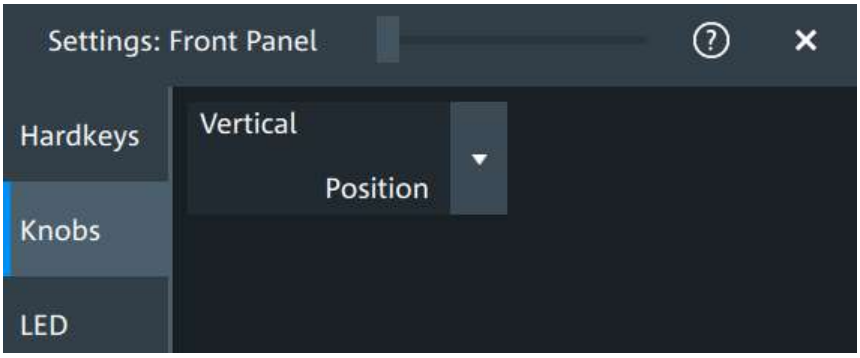
You can assign one of the following actions:

- Force trigger
- Save saveset
- Recall saveset

For details of saveset configuration, see Section 13.3, "General and measurement settings: savesets", on page 451.

### 5.5.2 Knobs

Access: "Settings" > "Front panel" > "Knobs"

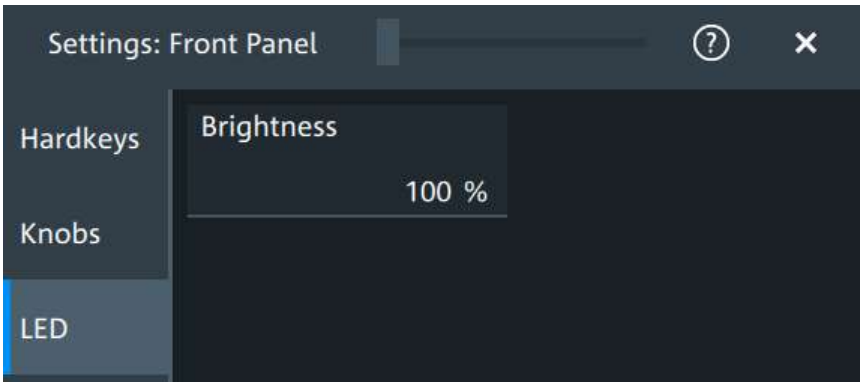



**Vertical**
The vertical Position knob can change the waveform position or the offset of the selected waveform. Select the action that you want to perform.

See also: "[Position]" on page 48.

### 5.5.3 LED

Access: "Settings" > "Front panel" > "LED".




**Brightness**
Defines the luminosity of illuminated front panel keys and knobs.

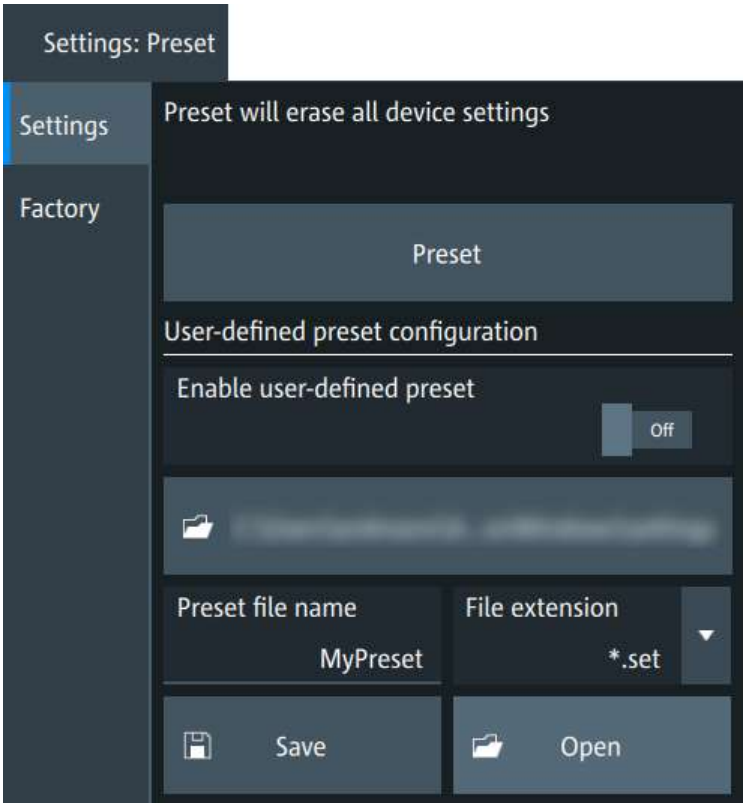
# 5.6 Preset setup

The preset sets the MXO 3 to a default configuration, or to a user-defined configuration. A user-defined preset file is one of the saveset files, which are stored locally. It contains the measurement setup, while the user's display settings are not included. You can save the current configuration to a preset file, and enable the saved preset file to be applied with the "Preset" toolbar icon.

See also: Section 13.3, "General and measurement settings: savesets", on page 451.

## 5.6.1 Preset settings

Access: "Menu" > "Settings" > "Preset" > "Settings".




**Preset**
Set the MXO 3 to a default configuration, or to a user-defined configuration.

**Enable user-defined preset**
If enabled, the settings from the selected saveset are restored when the [Preset] key is pressed.

If disabled, [Preset] sets the instrument to the factory defaults.

Remote command:
`USERdefined:PRESet[:ENABle]` on page 987

**Preset file selection**
Select the saveset file that contains the required settings.

| | |
|---|---|
| <Directory> | Shows the currently used directory. To change the directory, tap the button and select the correct directory in the file dialog. |
| "Preset file name" | Enter the filename of the preset file. This file is loaded on user-defined preset. The filename is also used when you save a preset file with "Save". |
| "File extension" | The file extension is `.set`. |

Remote command:
`USERdefined:PRESet:NAME` on page 987
`USERdefined:PRESet:OPEN` on page 987

**Save, Open**
"Save" writes the current settings immediately to the file that is named in "Preset file name" in the selected directory. "Open" loads a saveset file as preset settings.

Remote command:
`USERdefined:PRESet:SAVE` on page 987
`USERdefined:PRESet:OPEN` on page 987

### 5.6.2 Factory preset

Access: "Menu" > "Settings" > "Preset" > "Factory".

**Factory preset**
Resets the MXO 3 to the factory default settings, to the initial state. Factory settings comprise all instrument settings.

Remote command:
`SYSTem:PRESet` on page 954

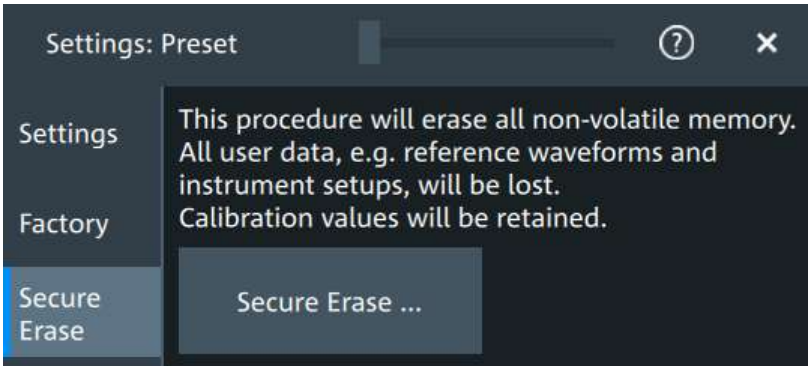
### 5.6.3 Secure erase

Access: "Menu" > "Settings" > "Preset" > "Secure erase"

To purge all non-volatile memory, a special delete function is available. This function erases all user data, including instrument setups and reference waveforms. Calibration data is retained on the instrument. The instrument reboots after erasing the data.

The procedure is required if user data must not leave the test area with the instrument, e.g. when working in highly secured environments.

For more information, refer also to "MXO 3 Instrument Security Procedures". It is available for download on the Rohde & Schwarz webpage.




### 5.6.4 Restoring settings

When you have changed many settings, and you are not sure which settings are causing which effect, you can restore the default settings and start anew. The following methods are available:

- Saving the settings as a user-defined preset and rest the settings to user-defined default values
- Restoring all settings on the MXO 3 to the factory-defined values
- Loading settings from a file

**To save a user-defined preset**

1. Open the "Menu" > "Settings" > "Preset" tab.
2. Enter a name for the preset file. Select the file format.
3. Tap "Save".

   **Note:** If you want to store the file in another directory than the displayed one, select "Save as". See also: Section 13.9, "File browser dialog", on page 486.

**To rest the settings to user-defined default values**

1. Open the "Menu" > "Settings" > "Preset" > "Settings" tab.
2. In "Preset file name", enter the name of the file that contains the required settings.
3. To use these settings as preset values, select "Enable user-defined preset".
4. Tap "Preset".

   To preset the instrument whenever needed, press the [Preset] key.

**To restore all settings to the factory defaults**

1. Open the "Menu" > "Settings" > "Preset" > "Factory" tab.
2. Tap the "Factory preset" button.

   All settings on the MXO 3 are reset to their factory-defined values.

As long as no user-defined preset file is loaded and "Enable user-defined preset" is disabled, the preset also resets the settings to factory defaults.

## 5.7 Maintenance settings

In the "Settings" > "Maintenance" dialog box, you can update the firmware, perform self-alignment.

### 5.7.1 Firmware update

Access: "Menu" > "Settings" >"Maintenance" > "FW update".

Your instrument is delivered with the latest firmware version. Firmware updates are provided on the internet at:

The "Release Notes" describe the improvements and modifications of all firmware versions. They also explain how to update the firmware. They are available along with the firmware on the same web page.

Remote commands:

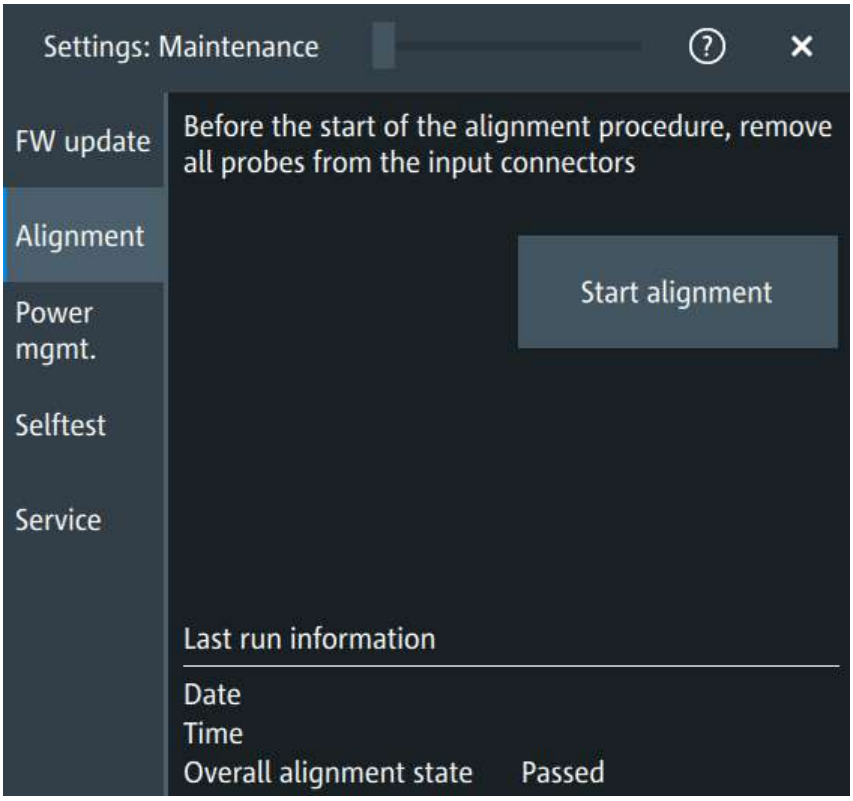
- `SYSTem:FW:FILepath` on page 954
- `SYSTem:FW:STARt` on page 955

### 5.7.2 Alignment

When data from several input channels is displayed at the same time, it can be necessary to align the data. Alignment is done vertically or horizontally to synchronize the time bases or amplitudes and positions. Perform an alignment, for example, when strong temperature changes occur (> 5°).

#### 5.7.2.1 Alignment settings

Access: "Menu" > "Settings" > "Maintenance" > "Alignment"



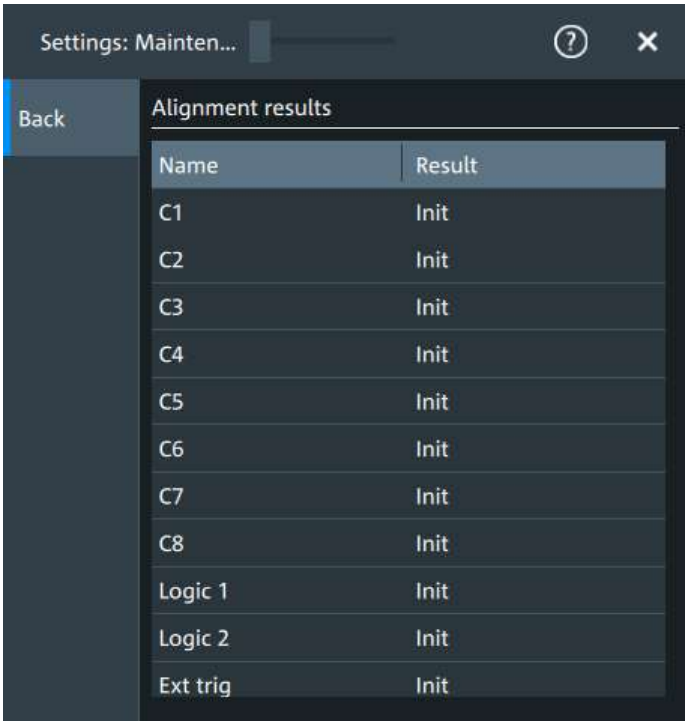

**Start alignment**
Starts the self-alignment procedure for all channels.

**Date, Time, Overall alignment state**
Show the date, time and the summary result of the self-alignment process: not aligned, passed or failed. Detailed results are provided on the "Alignment results" tab.

Remote command:
`CALibration:DATE?` on page 988
`CALibration:TIME?` on page 988
`CALibration:RESult?` on page 988

**Show results**
Opens a dialog to display the alignment results.

For each channel, the results of the individual alignment steps are shown for all technical channel component. In case you require support, you may be asked to provide this information.




### 5.7.2.2 Performing a self-alignment

The self-alignment aligns the data from several input channels vertically and horizontally to synchronize the timebases, amplitudes and positions. The self-alignment process includes a basic hardware check.

Recommendation on performing the self-alignment:

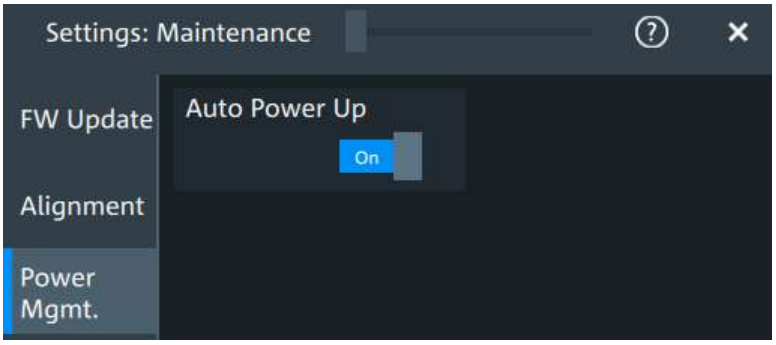
- When putting the instrument into operation for the first time
- After a firmware update
- Once a week
- When major temperature changes occur (> 5°)

1. Warm up the instrument before you start the self-alignment. The minimum warm-up time is indicated in the specifications document.
2. Remove the probes from the input connectors.
3. Open "Menu" > "Settings" > "Maintenance".
4. In the "Alignment" tab, tap "Start alignment".

   The alignment is performed, the process can take several minutes. A message box informs you about the running process, wait until this message box closes.
   The overall pass/fail result is shown in the "Overall alignment state" field.

### 5.7.3 Power management

Access: "Menu" > "Settings" > "Maintenance" > "Power mgmt."



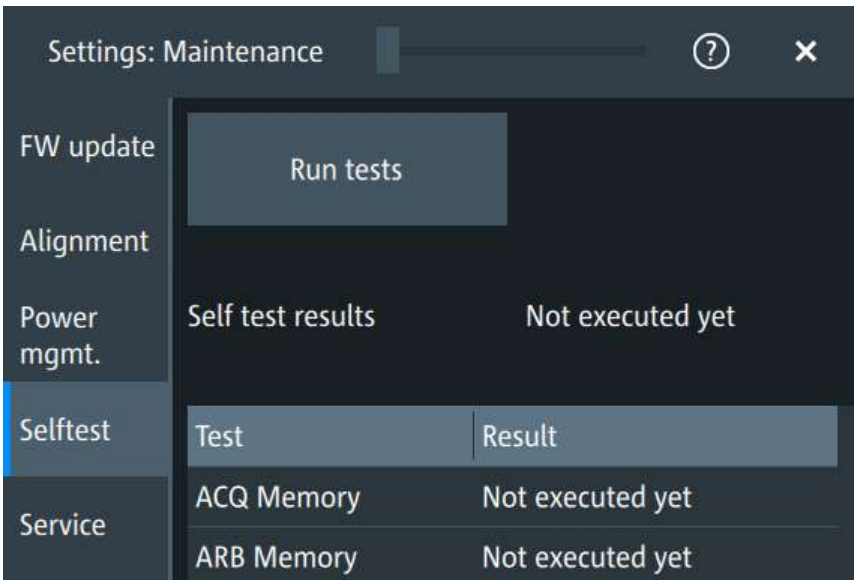

**Auto power up**
If enabled, the instrument powers up automatically when it is connected to the mains voltage, and after a power outage. If disabled, the instrument also powers up after a power outage if it was running when the power outage occurred.

Remote command:
`SYSTem:APUP` on page 989

### 5.7.4 Self-test

The instrument's selftest checks the hardware for correct operation. Perform the self-test if you suspect problems in hardware operation.

Access: "Settings" > "Maintenance"> "Selftest"




The selftest can take several minutes. The summary result is shown in the "Selftest results" field, and the results of the single tests are shown in the "Result" column of the test table.

**Run tests**
Starts the selftest.

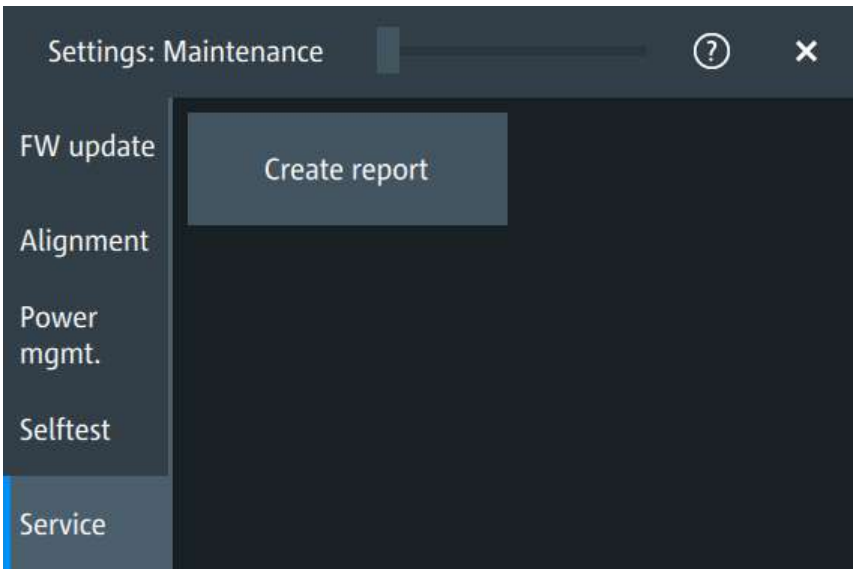
Remote command:
*TST? on page 949

**Selftest results**
Shows the summary result of the self-test: pass or fail.

**Test table**
The table shows the available tests and their particular tests results. This can be helpful in case you need support.

### 5.7.5 Service

Access: "Menu" > "Settings" > "Maintenance" > "Service"




**Create report**
Creates a service report.

The service report is a ZIP file with a complete bug report, all relevant setup information, reporting and log files, alignment files, and the instrument configuration.

If a USB flash drive is connected, the report is saved on the USB flash drive. Otherwise, the report is saved in the user data folder /home/storage/userData.

See also: Section 19.4, "Information for technical support", on page 2060.

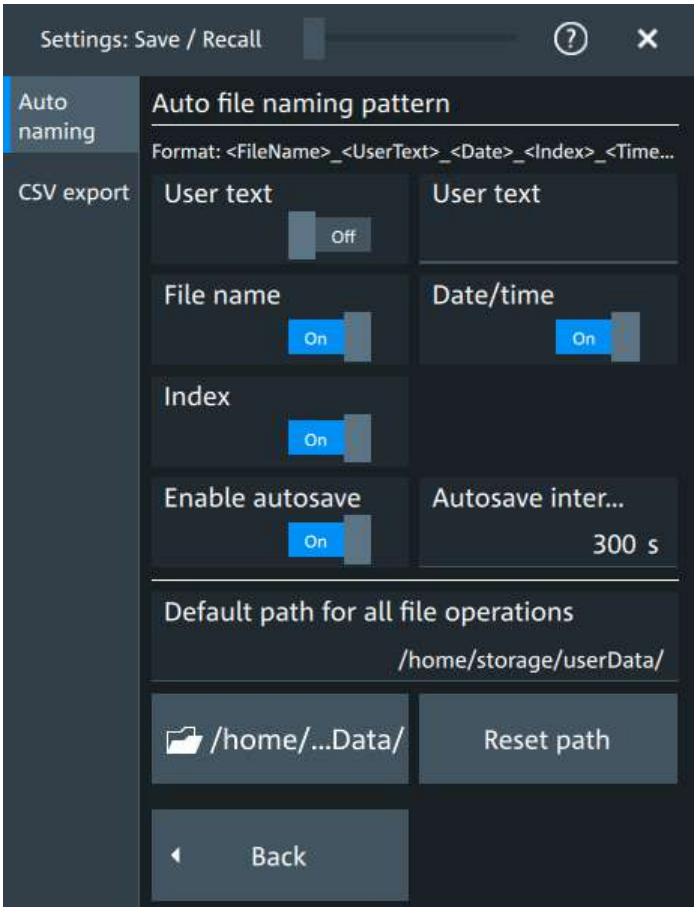
Remote command:
SERVice:REPort on page 989

# 5.8 Save / recall settings

## 5.8.1 Autonaming

Access: "Menu" > "Settings" > "Save/recall" > "Autonaming" tab.

### 5.8.1.1 Autonaming settings




In the "Autonaming" tab, you can define the pattern for automatic filename generation. This name is used as the default filename. The default path is the storage location for all saved files and their subdirectories. The local directory for user data is `/home/storage/userData`, the path to a USB storage device is `/run/media/usb/<MyDriveName>/`.

**User text**
If enabled, inserts the specified user text after the prefix.

Remote command:
`MMEMory:AUTonaming:USERtext` on page 985
`MMEMory:AUTonaming:TEXT` on page 986

**File name base**
If enabled, inserts the default filename. The filename indicates the type of data that is saved, for example, RefCurve.

Remote command:
MMEMory:AUTonaming:PREFix on page 985

**Date/time**
If enabled, the current date and time are inserted in the filename pattern.

Remote command:
MMEMory:AUTonaming:TIME on page 985

**Index**
If enabled, inserts an index.

Remote command:
MMEMory:AUTonaming:INDex on page 985

**Enable autosave**
Enables the automatic saving. A saveset is generated and saved in a given interval that you define with "Autosave interval". If you switch off the instrument, the last saved settings are recovered.

Remote command:
MMEMory:AUSave:ENABle on page 985
MMEMory:AUSave:INTerval on page 985

**Default path for all file operations**
Defines the path that is displayed in the file selection dialog box for loading and storing operations. Locally, all user data is written to `/home/storage/userData`. You can create subfolders in this folder.

If a USB storage device is connected, the path is automatically changed to the USB path `/run/media/usb/<MyDriveName>/`.

To switch the directory quickly, double-tap the path button. Use the symbols on the left of the file explorer box to change the directory.

Remote command:
MMEMory:AUTonaming:DEFaultpath on page 986

**Reset path**
Resets the default file path to the factory default.

Remote command:
MMEMory:AUTonaming:RESPath on page 986
MMEMory:AUTonaming:RESall on page 986

### 5.8.1.2 Defining default file paths and names

When a save or load operation is performed, a default filename and path is provided. You can configure which path is used, and how the filename is generated. In the file selection dialog box, you can change the folder and name as desired.

**To define the default file path**

1. Tap "Menu" > "Settings".
2. Select the "Save/recall" tab.
3. Select the "Autonaming" tab.
4. Double-tap the path button.

   The directory selection dialog box is opened.
5. Select the folder in which the data is stored by default. Use the symbols on the left of the file explorer box to switch to often used directories.
   Alternatively, you can tap the "Default path for all file operations" field and type the path.
6. To restore the factory-set default path, tap "Reset" next to the path field.

**To define the automatic filename pattern**

The automatic filename pattern can consist of the following elements:

<FileName>_<UserText>_<Date>_<Index>_<Time>

The prefix depends on the data type to be stored and cannot be changed by the user. The other elements can be enabled or disabled as required.

1. Tap "Menu" > "Settings".
2. Select the "Save/recall" tab.
3. Select the "Autonaming" tab.
4. If you want to exclude the "File name base", "Date/time" or an "Index" (serial number), disable the corresponding option.
5. To insert a user-defined text after the prefix, enable "User text". Enter the text in the edit field.

   The specified elements are used to generate the default filename for the next storage operation.

### 5.8.2 CSV export

Access: "Menu" > "Settings" > "Save/recall" > "CSV export" tab.

Currently, there are no active settings in this dialog.

# 6 Acquisition and waveform setup

## 6.1 Horizontal setup

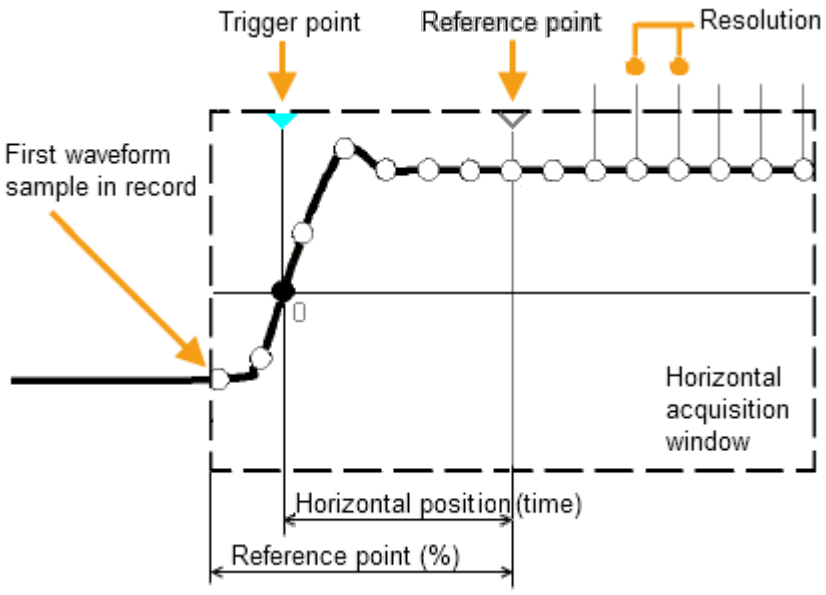
The "Horizontal" dialog provides the timebase configuration for channel and spectrum waveforms.

### 6.1.1 About the horizontal system

Horizontal settings, also known as timebase settings, adjust the waveforms in horizontal direction.

Typically, the trigger is the determining point of the waveform record. In many scenarios, you want to analyze the waveform some time before or after the trigger. To adjust the horizontal acquisition window to the waveform section of interest, you can use the following parameters:

- The **horizontal position** defines the time distance of the trigger point (the zero point of the diagram) to the reference point. Changing the horizontal position, you can move the trigger point, even outside the screen.
- The **reference point** is the rescaling center of the time scale on the screen. If you modify the time scale, the reference point remains fixed on the screen, and the scale is stretched or compressed to both sides of the reference point.




Unlike vertical settings, which are waveform-specific, the horizontal settings apply to all active waveforms.

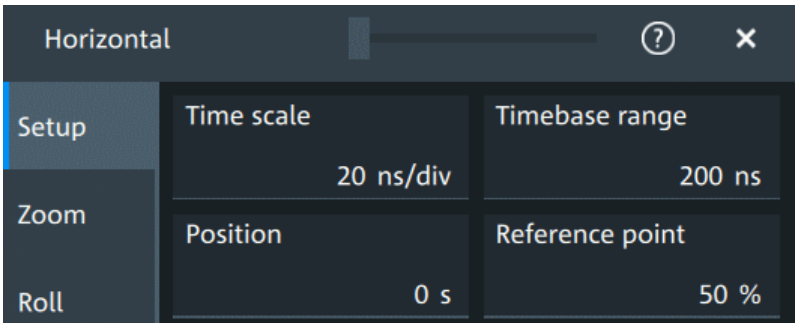
Timebase settings are interdependent:

*Timebase range = Time scale * Number of divisions*

The number of divisions is 10, which is the only constant parameter.

### 6.1.2 Horizontal Setup settings

Access: "Menu" > "Horizontal" > "Setup" tab, or tap the "Horizontal" label above the diagram.




**Time scale**
Sets the horizontal scale, the time per division, for all waveforms in the time domain, for example, channel and math waveforms.

Increase the scale to see a longer time interval of the waveform. Decrease the scale to see it in more detail. The scale has a point that remains fixed on the screen when the scale value is changing: the reference point.

Remote command:
`TIMebase:SCALe` on page 990

**Timebase range**
Sets the time of one acquisition, which is the time across the 10 divisions of the diagram: *Acquisition time = Time scale * 10 divisions*.

Changing the acquisition time changes the time scale, too.

For long timebases, the acquisition progress is shown in the acquisition label.

Remote command:
`TIMebase:RANGe` on page 991

**Position**
Defines the time distance between the reference point and the trigger point, which is the zero point of the diagram. The horizontal position is also known as trigger offset.

If you want to see a section of the waveform some time before or after the trigger, enter this time as horizontal position. The requested waveform section is shown around the reference point. Use positive values to see waveform sections after the trigger - the waveform and the diagram origin move to the left.

Remote command:
`TIMebase:HORizontal:POSition` on page 991

**Reference point**
Sets the position of the reference point in % of the screen. It defines which part of the waveform is shown.

The reference point marks the rescaling center of the time scale on the screen. If you modify the time scale, the reference point remains fixed on the screen, and the scale is stretched or compressed to both sides of the reference point. If the "Position" is 0, the trigger point is on the reference point.

The reference point is not marked in the diagram.

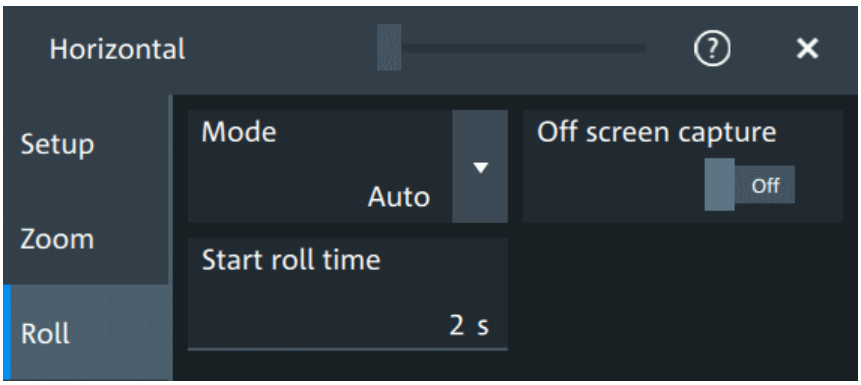
Remote command:
`TIMebase:REFerence` on page 992

### 6.1.3 Zoom settings

The zoom settings are described in Section 9.1, "Zoom", on page 222.

### 6.1.4 Roll mode

Access: "Menu" > "Horizontal" > "Roll" tab.




In this dialog, you can define the behavior of the roll mode.

**Roll mode and acquisition setup**

The settings "SR mode" and "RL mode" can affect the display of the waveform display in roll mode if these settings are set to "Manual". The resulting acquisition time is centered on the screen center.

If the acquisition time (*record length / sample rate*) is longer than the timebase range, the roll mode starts invisibly, outside the right screen edge. The data appears with some delay.

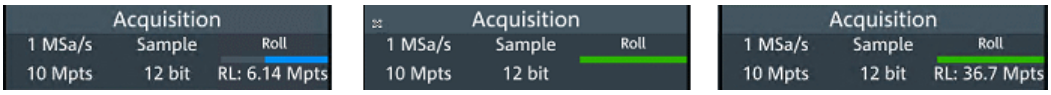
If the acquisition time is shorter than the timebase range, there is a gap from the right screen edge to the beginning of the data. If "SR mode" and "RL mode" are set to "Auto", the instrument optimizes the acquisition settings.

### Roll mode settings

Progress bar....123
Mode....123
Start roll time....123
Off screen capture....124

### Progress bar

The "Roll" progress bar in the "Acquisition" label shows how the record length is filled. The bar turns green when all samples are acquired, and the record is filled.




Left = incomplete roll waveform
Middle = roll completed with "Off screen capture" = Off
Right = roll completed with "Off screen capture" = On

### Mode

Selects, if the roll mode is started automatically by the instrument or if it is turned off.

In roll mode, the instrument shows the waveforms immediately, without waiting for the complete acquisition of the waveform record. If the timebase is slow - at long time scale values - the roll mode saves time waiting for the waveform display. The instrument displays the newly acquired waveform points at the right edge of the display and moves the waveform to the left.

The roll mode has the following restrictions:

- Roll mode disables persistence.
- Event actions are not possible.

The instrument activates the roll mode automatically if the following conditions are fulfilled:

- "Timebase range" exceeds the defined "Start roll time".
- "Acquisition mode" is set to "Sample" or "Peak detect".
- Actual sample rate ≤ 0.05 * sample rate of the instrument, i.e. ≤ 250 Msample/s or 125 Msample/s depending on the number of active channels.
- "N-single/avg count" = 1.
- Fast segmentation is disabled.
- Event actions are disabled.
- All serial buses are disabled.
- All mask tests are disabled.

Measurement statistics evaluate the currently displayed waveform. Therefore, "Measure all events" is enabled automatically in roll mode.

See also: "Roll mode and acquisition setup" on page 122.

Remote command:

`TIMebase:ROLL:ENABle` on page 992
`TIMebase:ROLL:STATe?` on page 993

### Start roll time

Sets the minimum acquisition time for the automatic start of the roll mode.

The instrument activates the roll mode automatically if the "Timebase range" exceeds the value of the "Start roll time".

Remote command:
`TIMebase:ROLL:MTIMe` on page 992

**Off screen capture**
If enabled, the analyzable waveform in roll mode is extended. You can run the roll mode, stop the acquisition after some time, and analyze the data that is on the display and in the unvisible area on the left.

If "Off screen capture" = Off, the analyzable record length is the value that is set in the "Acquisition" > "Setup" tab. The record length determines the length of the displayed waveform.

If "Off screen capture" = On, the whole acquisition memory can be used. After the defined record length is filled, more data is recorded at the left of the completed record length until you stop the acquisition or the acquisition memory is filled.

The current number of recorded data points is shown in the progress bar when the bar is green, or returned by the remote command `ACQuire:ROLLmode:POINts?`. See also: "Roll mode and acquisition setup" on page 122.

The "Roll" progress bar in the "Acquisition" label shows how the record length is filled.

After stopping the acquisition, you can analyze the complete roll waveform, including the invisible part:
- Increase the "Time scale" to see a longer part of the waveform.
- Scroll the waveform: tap and drag the waveform to the right.

If you want to save the waveform, disable "Off screen capture". When the needed data is displayed, save the waveform. Thus, one record length is saved. With active "Off screen capture", the amount of data cannot be limited.

Remote command:
`ACQuire:ROLLmode:OSCapture` on page 993
`ACQuire:ROLLmode:POINts?` on page 993

## 6.2 Acquisition

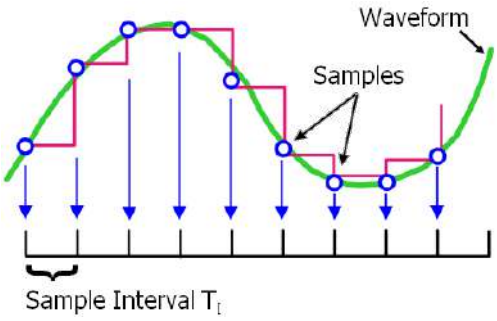
Access: "Menu" > "Acquisition".

Acquisition settings define the processing of the captured samples in the instrument.

### 6.2.1 About the acquisition system

**Sampling and processing**

The A/D converter samples the continuous signal under test at specific points in time and captures digital values. The converter is working at a constant rate specified in GHz.




The captured values are processed according to the acquisition settings. The result is a waveform record that contains **waveform samples** and is stored in the **waveform memory**. The waveform samples are displayed on the screen and build up the waveform.

The number of waveform samples in one waveform record is called **record length**. The rate of recording waveform samples - the number of waveform samples per second - is the **sample rate**. The higher the sample rate, the better the resolution is and the more details of the waveform are visible.

*Sample rate = 1 / Resolution*

The sample rate can be the same as the constant rate of the A/D converter, or higher, or lower. To get a higher sample rate, interpolation is used. Several interpolation methods are available. Other processing methods reduce the sample rate, or build the resulting waveform from several consecutive acquisitions of the signal. These methods are called acquisition modes.

### Minimum sample rate and aliasing

A sufficient resolution is essential for correct reconstruction of the waveform. If the signal is undersampled, aliasing occurs - a false waveform is displayed. To avoid aliasing and accurately reconstruct a signal, the sample rate must be at least 3 to 5 times the fastest frequency component of the signal. A higher sample rate increases signal fidelity, increases the chance to capture glitches and other signal anomalies, and improves the zoom-in capabilities.

### Interleaving

The MXO 3 achieves its highest sample rate of 5 Gsample/s by interleaving channels in pairs. Interleaving assumes that only one of the paired channels is used. If the second channel of a pair is used (on display, or as trigger source, math source, or measurement source), the interleaving mode is disabled. Without interleaving, the channels work with 2.5 Gsample/s and reduced bandwidth.

The interleaved channels are:

- C1 and C2
- C3 and C4
- C5 and C6 (on 8-channel oscilloscope)
- C7 and C8 (on 8-channel oscilloscope)

### 6.2.2 Acquisition setup

Access: "Menu" > "Acquisition" > "Setup" tab.

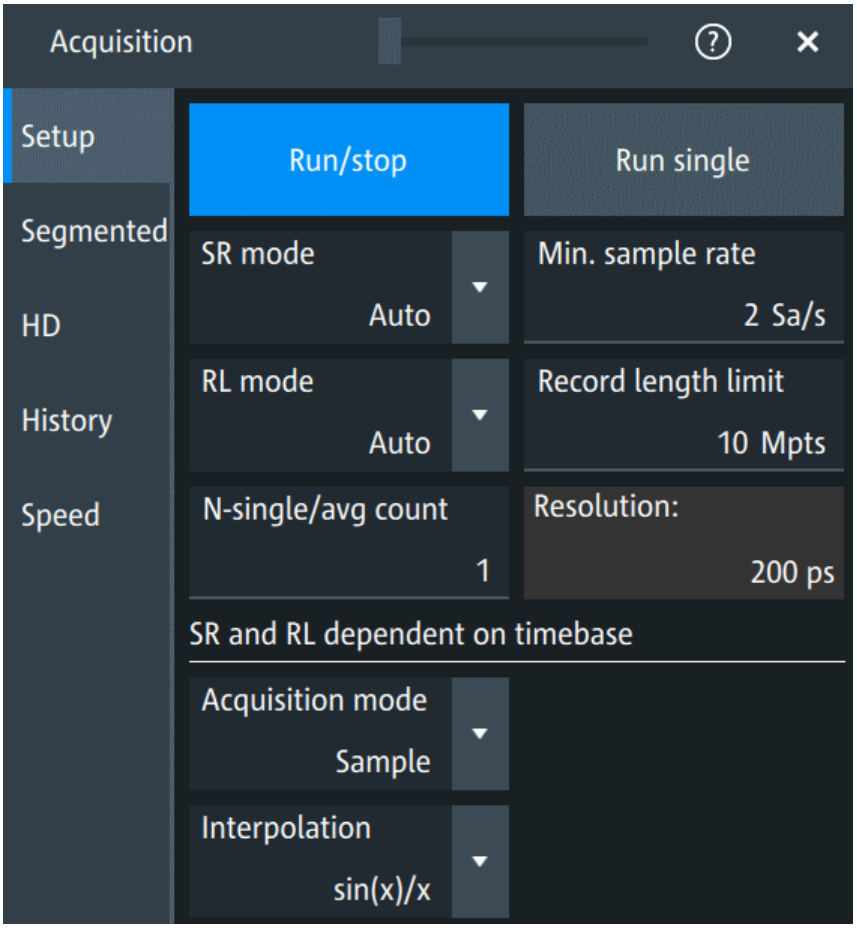



*Figure 6-1: Acquisition settings: automatic sample rate and record length*

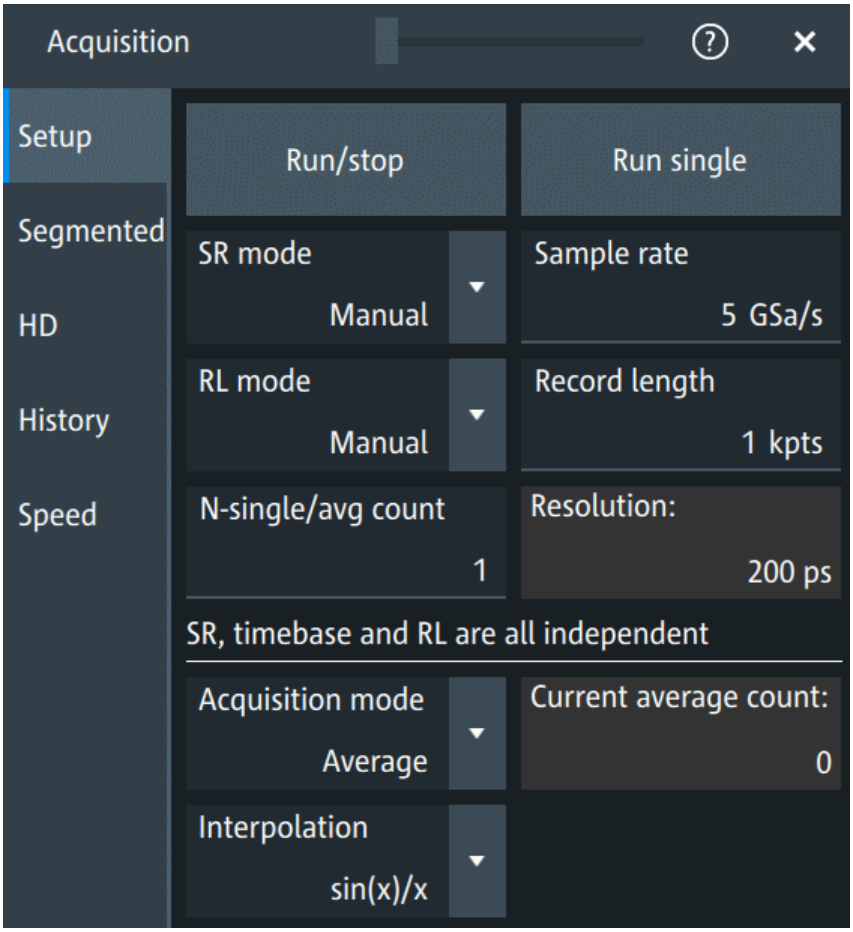



*Figure 6-2: Acquisition settings: manual sample rate and record length*

**Run/stop**
Starts and stops the continuous acquisition.

The [Run / Stop] key lights green when the acquisition is running. When the acquisition is stopped, the key lights red.

Remote command:
`RUN` on page 989
`STOP` on page 990

**Run single**
Starts a defined number of acquisitions.

The [Single] key lights green when the acquisition is running. When the acquisition is stopped, the key lights red.

To set the number of acquisitions, set "N-single/avg count" in the "Acquisition" setup. The acquisition progress is shown in the acquisition label, see "Progress information" on page 80.

Remote command:
`SINGle` on page 990

**SR mode**
Defines how the sample rate is set.

The sample rate considers the samples of the ADC, and the processing of the captured samples including interpolation.

"Auto" Sample rate is determined automatically and changes due to instrument internal adjustments due to other setting changes. You can set a minimum sample rate with Min. sample rate.

"Manual" The sample rate is manually defined with Sample rate.

Remote command:
`ACQuire:SRATe:MODE` on page 998

**Min. sample rate**
Sets the minimum value of the sample rate for calculation in automatic sample rate mode. It is the sample rate of the processed waveform record including interpolation.

Remote command:
`ACQuire:SRATe:MINimum` on page 998

**Sample rate**
Sets the number of waveform points per second in manual sample rate mode. It is the sample rate of the processed waveform record including interpolation.

The sample rate in the dialog differs from the sample rate that is shown in the acquisition label above the diagram. The value in the acquisition label shows the sample rate of the acquisition itself, before interpolation. It is the ADC sample rate or an integer decimated value of the ADC sample rate, which allows you to estimate whether the signal was sufficiently sampled in the acquisition.

If interpolation is used to reach the defined value, it is indicated by "Sample rate (IT)" in the dialog.

Remote command:
`ACQuire:SRATe[:VALue]` on page 998

**RL mode**
Selects the mode of the waveform record length adjustment.

The record length is the number of waveform samples that are stored in one waveform record after processing, including interpolation. It determines the length of the displayed waveform.

"Auto" Record length is determined automatically and changes due to instrument internal adjustments due to other setting changes. You can set a maximum record length with Record length limit.

"Manual" The waveform record length is manually defined with Record length.

Remote command:
`ACQuire:POINts:MODE` on page 996

**Record length limit**
Sets the maximum value of the record length for calculation in automatic record length mode.

Remote command:
ACQuire:POINts:MAXimum on page 996

**Record length**
Sets the record length in manual record length mode.

If the record length value is too high in combination with other settings and activated features, the memory is not sufficient to process the data. In this case, a warning information is shown in the dialog.

Remote command:
ACQuire:POINts[:VALue] on page 995
ACQuire:POMemory? on page 997

**N-single/avg count**
The acquisition and average count has several effects:
- It sets the number of waveforms acquired with "Run single".
- It defines the number of waveforms used to calculate the average waveform. Thus, the instrument acquires sufficient waveforms to calculate the correct average if "Average" is enabled for waveform arithmetic. The higher the value is, the better the noise is reduced.
- It sets the number of acquisitions to be acquired in a fast-segmentation acquisition series, and in the history.
- It is the "Finished" criteria for a mask test.

The setting is read-only if "Segmented" > "Acquire maximum" is "On". See also "Acquire maximum" on page 131.

Remote command:
ACQuire:COUNt on page 994

**Resolution**
Shows the current resolution.

The resolution is the time between two waveform samples in the waveform record. It considers the processing of the captured samples including interpolation. A fine resolution with low values produces a more precise waveform record. The resolution is the reciprocal of the sample rate.

If interpolation is used to reach the defined value, it is indicated by "Resolution (IT)" in the dialog.

Remote command:
ACQuire:RESolution? on page 997

**Acquisition mode**
Sets how the waveform is built from the captured samples.

| | |
|---|---|
| "Sample" | Usually, most signals are displayed optimally with this acquisition mode but very short glitches can remain undiscovered by this method. |
| "Peak detect" | The minimum and the maximum of n samples are recorded as waveform points, the other samples are discarded. Thus the instrument can detect fast signal peaks at slow time scale settings that would be missed with other acquisition modes. |

"Envelope" Each acquisition is done in peak detection mode, and the minimum and maximum values in a sample interval over some consecutive acquisitions build the envelope. The resulting diagram shows two envelope waveforms: the minimums (floor) and maximums (roof).

"Average" The average is calculated from the data of the current acquisition and a specific number of consecutive acquisitions before. The method reduces random noise. It requires a stable, triggered and repetitive signal.
The number of acquisitions for average calculation is defined with N-single/avg count.
If the waveform is clipped, the instrument shows a distorted average waveform to indicate the clipping. Adjust the vertical scale to avoid the clipping. The acquisition progress is shown in the acquisition label.

Remote command:
`ACQuire:TYPE` on page 999

**Current average count**
Shows the current number of acquired waveforms that contribute to the average, for Acquisition mode = "Average".

Remote command:
`ACQuire:AVERage?` on page 994

**Interpolation**
Selects the interpolation method.

If the defined sample rate ("Sample rate") is higher than the ADC sample rate, interpolation adds points between the captured samples of the waveform by various mathematic methods. All captured and interpolated points are used in measurements, and they are included in exports. If interpolation of samples is active, it is indicated by "Sample rate (IT)" and "Resolution (IT)".

"Linear" Two adjacent ADC sample points are connected by a straight line, the interpolated points are located on the line. You see a polygonal waveform similar to the real signal, and also the ADC sample points as vertexes.

"sin (x)/x" Two adjacent ADC sample points are connected by a sin(x)/x curve, and also the adjoining sample points are considered by this curve. The interpolated points are located on the resulting curve. This interpolation method is precise and shows the best signal curve.

"Sample/Hold" The ADC sample points are displayed like a histogram. For each sample interval, the voltage is taken from the sample point and considered as constant, and the intervals are connected with vertical lines. Thus, you see the discrete values of the ADC - the measured samples.

Visual interpolation of the waveforms and zoom waveforms on the display is independent of the calculated sample interpolation, and it is not indicated.

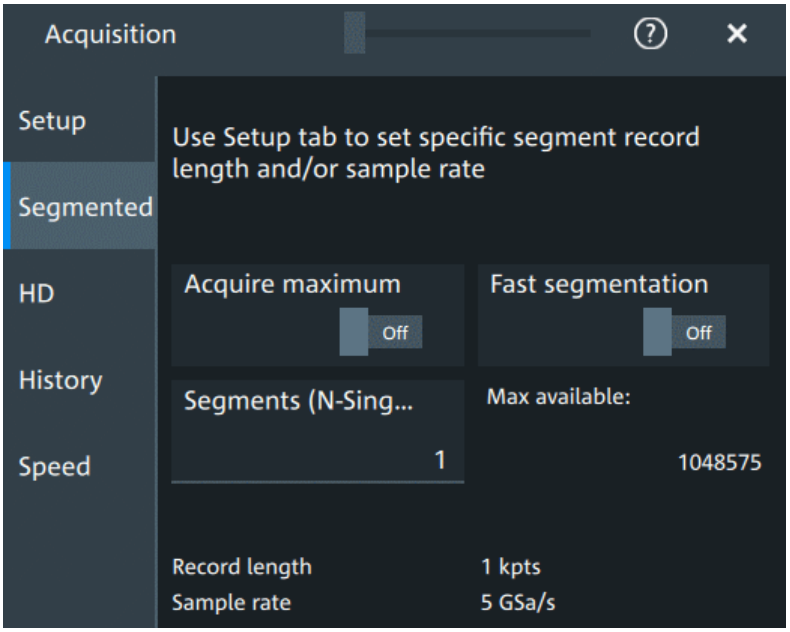
Remote command:
`ACQuire:INTerpolate` on page 995

### 6.2.3 Segmentation settings

Access: "Menu" > "Acquisition" > "Segmented" tab.

You can acquire a limited number of segments, or the maximum number.

In the R&S ScopeStudio software, acquisition settings and functions that need an acquisition take effect if the channel source is a simulated waveform, created by the generator. If the channel source is loaded from a file, the acquisition-related settings can be imported with the data from the oscilloscope, either as a saveset, or included in a session. Changing imported acquisition-related settings has no effect.




**Acquire maximum**
The setting takes effect for N-single acquisitions ("Run single"). During continuous acquisition ("Run / stop"), the history is used completely.

If "On", "Segments (N-single)" is set to the maximum number of available segments. Thus, the history is filled completely with an N-single acquisition. "Segments (N-single)" is read-only, you cannot change the value. The number of available segments depends on current the number of active channels, sample rate, record length, and other settings.

If "Off", set the number of segments in "Segments (N-single)", which is the same setting as "N-single/avg count" on the "Setup" tab. Thus you can acquire a fast segmentation acquisition series and history of defined length with "Run single".

You can stop the running acquisition before the series is completed.

The number of acquired waveforms is shown in the "History" dialog, when the history is active.

Remote command:
`ACQuire:SEGMented:MAX` on page 1028

**Fast segmentation**
If fast segmentation is enabled, the acquisitions are performed as fast as possible, without processing and displaying the waveforms. When acquisition has been stopped, the data is processed and the latest waveform is displayed. Older waveforms are stored in segments. You can display and analyze the segments using the history.

Remote command:
`ACQuire:SEGMented:STATe` on page 1028

**Segments (N-single)**
See "N-single/avg count" on page 129.

**Max available**
Shows the maximum number of segments that can be captured with current sample rate and record length settings.

## 6.2.4 High definition mode

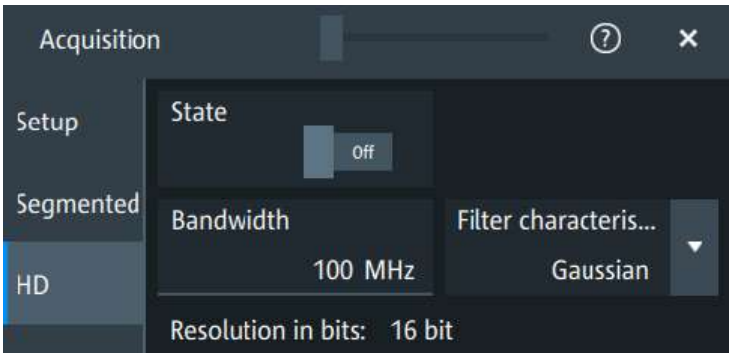
The high definition mode increases the numeric resolution of the waveform signal by using digital filtering, leading to reduced noise. The higher vertical resolution reduces quantization noise and acquires waveforms of higher accuracy with finer details of the signal to be seen. The high definition is also applied to the digital trigger, thus the MXO 3 can trigger with the same high resolution with which they can display signals.

The maximum numeric resolution in high definition mode is 18 bit. The actual value depends on the selected bandwidth. Increasing the bandwidth reduces the resulting digital resolution. For dependencies and details, refer to the MXO 3 specifications.

The number of vertical resolution bits defines the number of vertical levels that the acquisition samples are mapped to (quantization). For example, 16 bits of resolution represent 65536 voltage quantization levels, while 8 bits of resolution represent only 256 voltage levels. The waveform values are recorded with a word length of 16 bit, except for peak detect decimation.

### 6.2.4.1 High definition settings

Access: "Menu" > "Acquisition" > "HD" tab.



High definition is a special acquisition mode of the oscilloscope. This mode has only two settings - the filter bandwidth and the filter type.

**State**
Enables high definition mode, which increases the numeric resolution of the waveform signal.

Remote command:
`HDEFinition:STATe` on page 1030

**Bandwidth**
Sets the filter bandwidth for the high definition mode.

Increasing the bandwidth reduces the resulting digital resolution. For dependencies and details, refer to MXO 3 specifications.

Remote command:
`HDEFinition:BWIDth` on page 1029

**Filter characteristics**
Selects the filter that is used to reduce the bandwidth:

| | |
|---|---|
| "Gaussian" | Gaussian filter |
| "Bessel" | Bessel-Thomson filter, which has maximally flat group delay. It preserves the waveform shape of signals. Bessel filters are preferred in applications requiring high fidelity and timing accuracy. |
| "Brickwall" | Rectangular filter with full transmission in the pass band and complete attenuation in the stop band. |

Remote command:
`HDEFinition:BFTYpe` on page 1029

**Resolution in bits**
Displays the resulting vertical resolution in high definition mode. The higher the filter bandwidth, the lower the resolution.

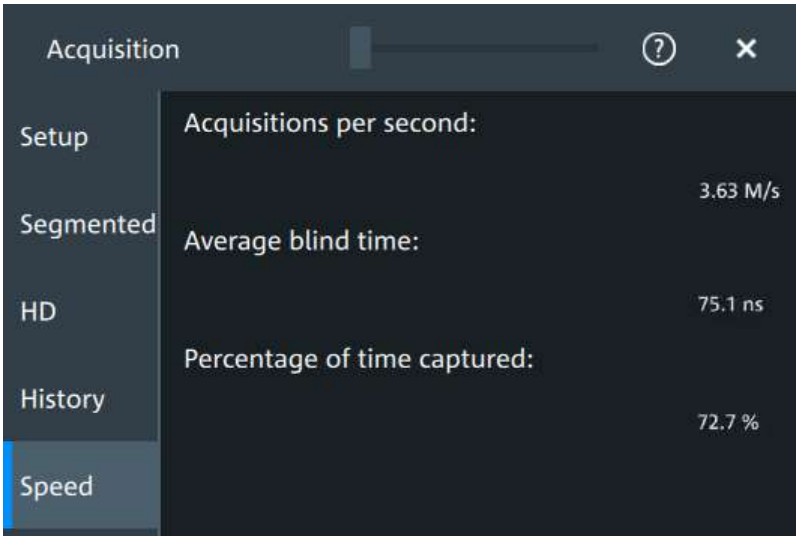
Remote command:
`HDEFinition:RESolution?` on page 1029

### 6.2.5 History settings

The history settings are described in Section 9.3.2, "History setup", on page 240.

### 6.2.6 Speed

Access: "Menu" > "Acquisition" > "Speed" tab.




The "Speed" dialog shows information on the current acquisition performance values of the MXO 3.

You can see the number of "Acquisitions per second", "Average blind time" and the "Percentage of time captured".

## 6.3 Vertical setup

The controls and parameters of the vertical system adjust the vertical scale and position of the waveform, and the waveform display. The probe settings also belong to the vertical setup.

The signal icons at the bottom of the display show the most important actual settings for each channel.

The signal icon also indicates if the incoming data is clipped before processing, i.e. the input range of the ADC is exceeded.

There are several ways to adjust vertical settings:

- Use the keys and knobs in the Vertical functional block of the front panel to select the channel, to scale the waveform, and to set the position or offset.
  See Section 3.2.4, "Keys and controls", on page 45.
- Use the "Vertical" dialog to adjust all vertical settings. See:
  - Section 6.3.2, "Vertical Setup settings", on page 135
  - Section 6.3.3, "Bandwidth settings", on page 138
  - Section 6.4, "Probes", on page 140
  - Section 6.3.5, "Other vertical settings", on page 139
- On the touchscreen, you can adjust the vertical settings directly on the screen:
  - Drag one finger vertically on the screen to change the position of the selected channel waveform.

– Spread or pinch two fingers in vertical direction to change the vertical scale of the selected waveform.

### 6.3.1 About the vertical system

The controls and parameters of the vertical system are used to scale and position the waveform vertically.

**Vertical scale and resolution**

Vertical scale and vertical position directly affect the resolution of the waveform amplitude. The vertical scale corresponds to the ADC input range. To get the full resolution of the ADC, set up the waveforms to cover most of the height of the diagram.

With an MXO 3, you work with multiple diagrams, and each diagram obtains the full vertical resolution, no matter where the diagram is placed. Therefore, use a separate diagram for each waveform instead of the traditional setup that arranges the waveforms side by side in one diagram.

**Bandwidth**

For analog applications, the highest signal frequency determines the required oscilloscope bandwidth. The oscilloscope bandwidth should be higher than the maximum frequency included in the analog test signal to measure the amplitude with very little measurement error. To avoid aliasing, the oscilloscope bandwidth should be at least 3 times higher than the maximum frequency included in the signal.

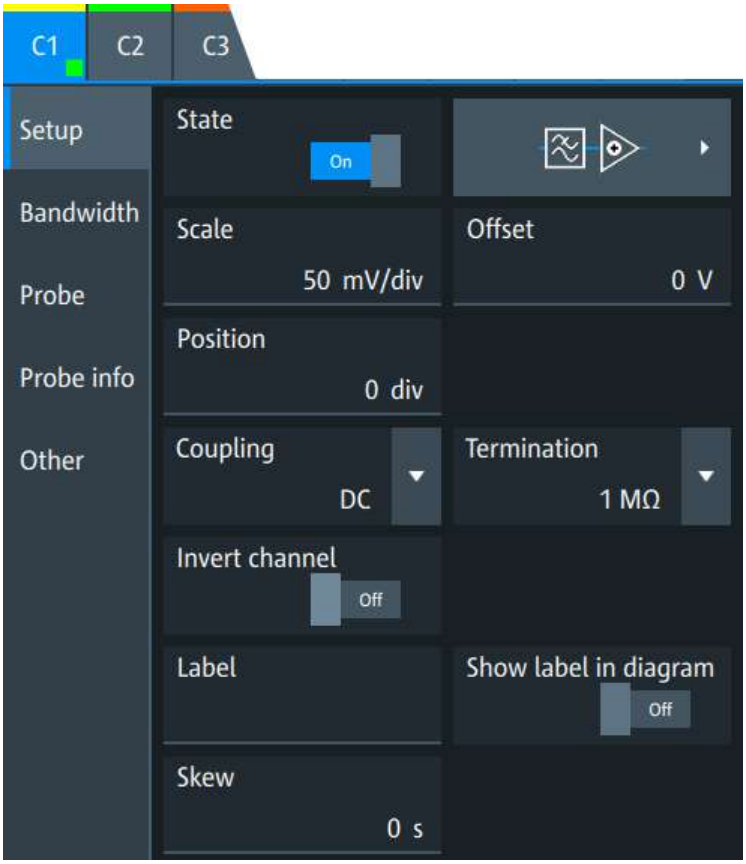
Most test signals are more complex than a simple sine wave and include several spectral components. A digital signal, for example, is built up of several odd harmonics. For digital signals, the oscilloscope bandwidth should be at least 5 times higher than the clock frequency to be measured.

The oscilloscope is not a stand-alone system. You need a probe to measure the signal of interest, and the probe has a limited bandwidth, too. The combination of oscilloscope and probe creates a system bandwidth. To reduce the effect of the probe on the system bandwidth, the probe bandwidth should exceed the bandwidth of the oscilloscope, the recommended factor is 1.5 x oscilloscope bandwidth.

### 6.3.2 Vertical Setup settings

Access: "Menu" > "Vertical" > "Setup" tab

The "Setup" tab provides all basic vertical settings. The channels are listed in horizontal subtabs. Make sure to select the correct channel tab before you enter the settings.




**State**
Switches the selected channel signal on or off.

The signal icon opens on the signal bar. The waveform of the last acquisition is displayed in the diagram.

Remote command:
`CHANnel<ch>:STATe` on page 999

**Signal flow (graphical button)**
Opens a dialog that shows the signal flow as currently configured. In addition, the main settings are shown for information.

**Scale**
Sets the vertical scale, which defines the displayed amplitude of the selected waveform.

The unit depends on the connected probe.

The current value is shown in the signal icon. Vertical scale directly affects the resolution of the waveform amplitude. To get the best resolution of the ADC, set the waveforms to cover most of the height of the diagram.

Remote command:
`CHANnel<ch>:SCALe` on page 1000

**Offset**
Sets the offset voltage, which corrects an offset-affected signal. The vertical center of the selected channel is shifted by the offset value and the signal is repositioned within the diagram.

The waveform marker indicates the offset on the vertical scale.

To set the offset automatically, use [Autoset]. The current value is shown in the signal icon.

Use the offset to measure small AC voltages that are overlaid by higher DC voltages. Unlike AC coupling, the DC part of the signal is not lost with offset setting.

If an active probe is connected, the offset limit is defined by the probe. Refer to the documentation of the probe for allowed values.

If a Rohde & Schwarz differential probe is connected, the "Offset" is the common-mode offset. Set the common-mode offset to compensate for a common DC voltage applied to both input sockets (referenced to the ground socket). Thus, you can measure on differential signals with high common mode levels. You can measure the common mode input voltage using the R&S ProbeMeter.

Remote command:
`CHANnel<ch>:OFFSet` on page 1000

**Position**
Moves the selected signal up or down in the diagram. While the offset sets a voltage, position is a graphical setting given in divisions. The visual effect is the same as for offset.

Remote command:
`CHANnel<ch>:POSition` on page 1001

**Coupling**
Sets the filter for the input signal. The coupling determines what part of the signal is used for waveform analysis and triggering.

The selected coupling is shown in the signal icon.

"DC" Passes both DC and AC components of the signal.

"AC" Connection through DC capacitor, removes DC and very low-frequency components. AC coupling is useful if the DC component of a signal is of no interest. The waveform is centered on zero volts. AC coupling is available with termination 1 MΩ.

If AC coupling is set, the attenuation of passive probes has no effect, and voltage is applied to the instrument with factor 1:1. Observe the voltage limits, otherwise you can damage the instrument.

Remote command:
`CHANnel<ch>:COUPling` on page 1001

**Termination**
Selects the input impedance of the channel input according to the connected probe.

Connection with 50 Ω termination is used to connect, for example, active probes. Connection with 1 MΩ termination is used to connect standard passive probes.

Remote command:
CHANnel<ch>:COUPling on page 1001

**Invert channel**
Turns the inversion of the signal amplitude on or off. To invert means to reflect the voltage values of all signal components against the ground level.

Inversion is indicated in the signal icon by a line above the channel name.

If the inverted channel is the trigger source, the instrument triggers on the inverted signal. You can use inversion, for example, to switch the polarity of a differential signal without changing the probe connections.

Remote command:
CHANnel<ch>:INVert on page 1002

**Label**
Defines a label text. The label is shown on the signal icon. Characters and label length are not resticted.

Remote command:
DISPlay:SIGNal:LABel on page 1002

**Show label in diagram**
Displays a waveform label at the waveform in the diagram, with the content defined in "Settings" > "Appearance" > "Grid". See Section 5.3.2, "Grid", on page 99 for details on the content settings.

**Skew**
Sets a skew value to compensate for the delay of the measurement setup or from the circuit specifics that the instrument cannot compensate automatically. It affects only the selected input channel.

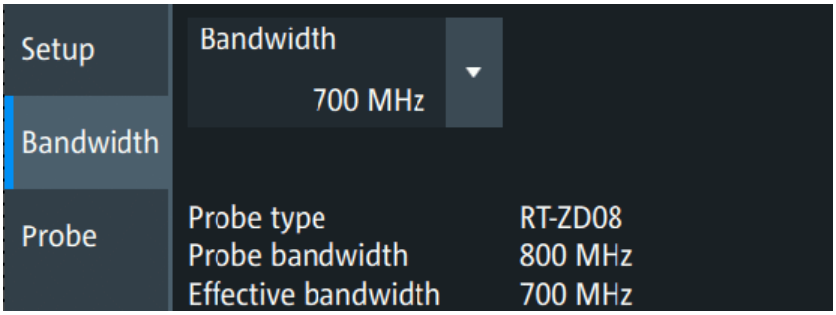
Delay differences between channels are caused by the different length of cables, probes, and other sources. Correct deskew values are important for accurate triggering. Signals that are routed over lines with different lengths have a different propagation delay. This delay can lead to a non-synchronous waveform display.

Remote command:
CHANnel<ch>:SKEW:TIME on page 1002

### 6.3.3 Bandwidth settings

Access: "Menu" > "Vertical" > "Bandwidth" tab

The "Bandwidth" tab provides all settings that affect the bandwidth of the measurement system. The channels are listed in horizontal subtabs. Make sure to select the correct channel tab before you enter the settings.




**Bandwidth**
Sets the bandwidth limit. The specified bandwidth indicates the range of frequencies that the instrument can acquire and display accurately with less than 3 dB attenuation. Frequencies above the limit are removed from the signal, and noise is reduced.

The selected bandwidth is shown on the signal icon.

For basic information, see also: "Bandwidth" on page 135.

| | |
|---|---|
| "Full" | At full bandwidth, all frequencies in the instrument's frequency range are acquired and displayed. Full bandwidth is used for most applications. However, at full bandwidth, the displayed bandwidth can be less than the instrument bandwidth depending on the connected probe, the number of active channels and other settings. |
| "xx MHz" | Frequencies above the selected limit are removed to reduce noise. |

Remote command:
`CHANnel<ch>:BANDwidth` on page 1003

**Probe type, Probe bandwidth**
Shows the type of the connected probe and its bandwidth. The probe is recognized automatically, or selected in the "Probe" tab. See Section 6.4.1, "Common probe settings", on page 141.

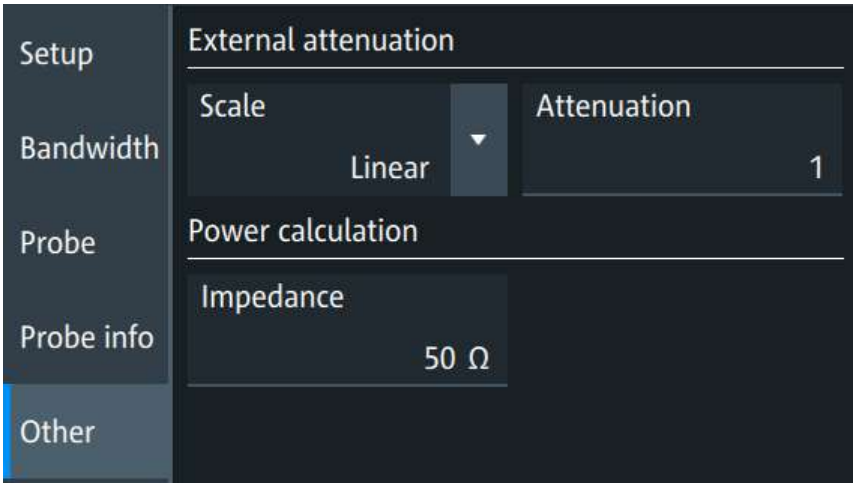
**Effective bandwidth**
Shows the effective bandwidth of probe and oscilloscope system.

## 6.3.4 Probe settings

See Section 6.4, "Probes", on page 140.

## 6.3.5 Other vertical settings

Access: "Menu" > "Vertical" > "Other" tab




**External Attenuation: Scale, Attenuation**
Consider a voltage divider that is part of the DUT before the measuring point. The external attenuation is included in the measurement, and the instrument shows the results that would be measured before the divider. External attenuation can be used with all probes.

"Scale" Select linear or logarithmic attenuation scale.

"Attenuation" Enter the attenuation of the voltage divider according to the selected scale. The conversion from linear to logarithmic values depends on the "Vertical unit" of the probe:
For voltage-based unit (V and A):
*attenuation (dB) = 20 * $log_{10}$(attenuation factor)*
For power-based unit (W):
*attenuation (dB) = 10 * $log_{10}$(attenuation factor)*

Remote command:
`CHANnel<ch>:EATScale` on page 1003
`CHANnel<ch>:EATTenuation` on page 1003

**Impedance**
Sets the impedance of the connected probe for power calculations and measurements.

The value is used, for example, for scale calculation in the spectrum diagram.

Remote command:
`CHANnel<ch>:IMPedance` on page 1004

## 6.4 Probes

With MXO oscilloscopes, you can use various probe types. Mostly these probes are passive and active voltage probes. The instrument can detect many probes and read out the probe-specific parameters, for example, bandwidth and attenuation.

You find all settings that are relevant for the connected probe in the "Vertical" > "Probe" tab. The settings on the "Probe" tab change according to the type of the attached

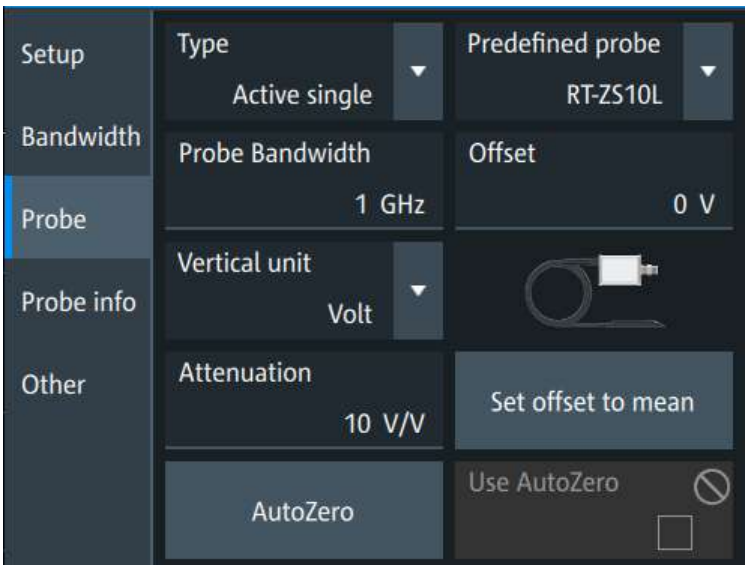
probe. Probes with Rohde & Schwarz probe interface (probe box), and also many other passive voltage probes, are recognized by the instrument. The oscilloscope reads out the main characteristics of the probe and displays them. Other probes cannot be detected, but their characteristics are known to the instrument. These known probes are called "Predefined probes". Probes that are not recognized automatically and not predefined are unknown probes, they require manual setting of measurement unit and attenuation.

The complete characteristic of the connected probe is shown on the "Vertical" > "Probe Info" tab.

### 6.4.1 Common probe settings

Access: "Menu" > "Vertical" > "Probe" tab

Most the settings in the "Probe" tab are available for all probes. For some probe types, additional settings are needed. These settings are described in the sections for the individual probe types.




An external attenuation can be set on the "Other" tab, see Section 6.3.5, "Other vertical settings", on page 139.

The common probe settings are:

Predefined probe, Type and Name of the probe ..... 141
Probe bandwidth ..... 142
Offset ..... 142
Vertical unit ..... 142
Attenuation ..... 142
Set offset to mean ..... 142
AutoZero, Use AutoZero ..... 143

**Predefined probe, Type and Name of the probe**
The model name and type of a detected probe are shown in the dialog.

If the instrument cannot recognize the probe, "None" is indicated in the "Type" and "Predefined probe" lists. Select the used probe "Type", and then the used model in "Predefined probe". The corresponding bandwidth, and attenuation or gain are shown.

If the probe is not detected and not listed as predefined probe, it is an unknown probe. To configure these probes, set "Type" to "User defined". Then select the attenuation or gain in "Predefined probe". You can also adjust the Vertical unit and Attenuation separately.

Remote command:
`PROBe<ch>:SETup:NAME?` on page 1008
`PROBe<ch>:SETup:TYPE?` on page 1009
`PROBe<ch>:SETup:STATe?` on page 1009
`PROBe<ch>:SETup:ATTenuation:DEFProbe` on page 1006

**Probe bandwidth**
Shows the bandwidth of the connected probe. For probes that are not detected or predefined, set the bandwidth manually.

Remote command:
`PROBe<ch>:SETup:BANDwidth?` on page 1007

**Offset**
Channel offset that is also set on "Vertical" > "Setup" tab. See "Offset" on page 137.

**Vertical unit**
Shows the unit of the connected probe if the probe is detected or predefined. For unknown probes, select the unit that the probe can measure.

Remote command:
`PROBe<ch>:SETup:ATTenuation:UNIT` on page 1007

**Attenuation**
Shows the attenuation of the connected probe if the probe is detected or predefined. If the probe is unknown to the instrument, set the correct attenuation of the probe in "Manual attenuation".

The vertical scaling and measured values are multiplied by this factor so that the displayed values are equal to the actual measured signal values.

Make sure to set the attenuation factor on the instrument according to the probe being used. Otherwise, the measurement results do not reflect the actual voltage level, and you might misjudge the actual risk.

Remote command:
`PROBe<ch>:SETup:ATTenuation[:AUTO]?` on page 1005
`PROBe<ch>:SETup:ATTenuation:MANual` on page 1005

**Set offset to mean**
Compensates automatically for a DC component of the input signal using the result of a background mean measurement.

The result is shown in "Offset". The function is probe-independent and supports quick and convenient measurements of input signals with different DC offsets. It detects offset values even when the signal is out of the current measurement range. It also sets the zero level to the determined DC offset in the middle of the screen and thus prevents clipping of the waveform.

Remote command:
`PROBe<ch>:SETup:OFFSet:TOMean` on page 1007

**AutoZero, Use AutoZero**
Voltage probes can have a zero error. The zero error is the voltage appearing at the probe output when nothing is connected to the probe. If a zero error occurs, it results in an external offset and the waveform is not displayed around 0 V.

To correct the zero error of voltage probes, short the signal pin and the ground pin together. Then select "AutoZero". While the alignment is running, the instrument switches to DC coupling to display the waveform correctly.

To correct the zero error, enable "Use AutoZero".

"AutoZero" uses the current scale and attenuation settings during alignment. If you change these settings, e.g. the vertical scale, repeat "AutoZero".

If a current probe is connected, the function demagnetizes the probe's sensor head and sets the waveform to zero position.

Remote command:
`PROBe<ch>:SETup:OFFSet:AZERo` on page 1008
`PROBe<ch>:SETup:OFFSet:USEautozero` on page 1008

### 6.4.2 Setup for passive probes

Passive probes are the most widely used probes for voltage measurements with oscilloscopes. If a passive probe is connected, the probe attenuation is read out and shown in the "Probe" tab.

Passive probes require compensation, see Section 6.4.9, "Adjusting passive probes", on page 162.

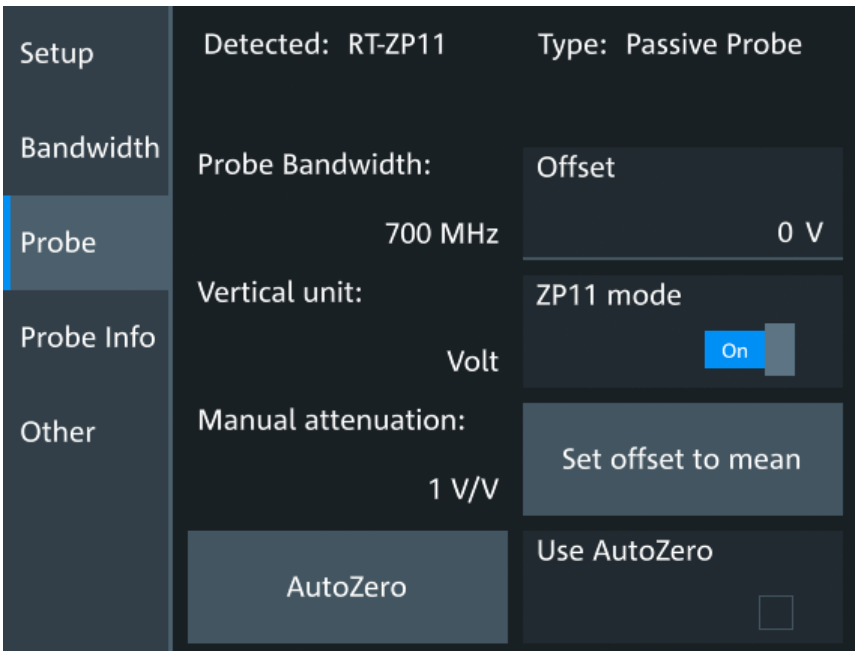



*Figure 6-3: Passive probe R&S RT-ZP11, detected by the oscilloscope*

The settings for passive probes are common settings, which are available for all probe types. See Section 6.4.1, "Common probe settings", on page 141 for description of the settings.

For R&S RT-ZP11, a special setting is available.

**ZP11 mode**
Enable if R&S RT-ZP11 is connected to adjust the probe bandwidth to 700 MHz.

If you use 500 MHz passive probe, e.g. R&S RT-ZP10, disable the setting.

## 6.4.3 Setup for active voltage probes

Active voltage probes with Rohde & Schwarz probe interface have an integrated data memory that contains identification data and individual probe correction parameters. The MXO 3 can detect these probes and read out the data. Furthermore, the Rohde & Schwarz probe interface provides special features: the micro button and the ProbeMeter.

Active voltage probes that are offered by Rohde & Schwarz but not equipped with a Rohde & Schwarz probe interface are known to the MXO 3 as predefined probes.

### 6.4.3.1 Settings for the Rohde & Schwarz probe interface

The Rohde & Schwarz probe interface provides special features: the micro button and the ProbeMeter.

**MicroButton**
The micro button is located on the probe head. Pressing this button, you initiate an action on the instrument directly from the probe. The button is disabled during internal automatic processes, for example, during self-alignment, autoset, and level detection.

Select the action that you want to start from the probe.

"Run continuous"
Starts or stops the acquisition (same as Run / Stop key).

"Run single" Starts a defined number of acquisitions (same as [Single] key).

"Autoset" Starts the autoset procedure (same as Autoset key).

"Auto zero" Starts an auto zero measurement.

"Offset to mean"
Performs an automatic compensation for a DC component of the input signal using the result of a background mean measurement.

"Save image to file"
Saves the current display as image according to the image settings.

"No action" Select this option to prevent unwanted actions due to unintended usage of the micro button.

"Find trigger level" Sets the trigger level automatically to *0.5 * (MaxPeak – MinPeak)*.

"Probe mode" Only available if a R&S RT-ZM modular probe is connected. Sets the measurement mode of the modular probe.

"Probe setup" Opens the probe setup on the screen.

Remote command:
`PROBe<ch>:SETup:MODE` on page 1010

**ProbeMeter**
Activates the integrated R&S ProbeMeter on probes with Rohde & Schwarz probe interface.

The R&S ProbeMeter is a voltmeter. It measures DC voltages between the probe tip and ground connection or between the probe tips with very high precision. The measurement runs continuously and in parallel to the measurements of the oscilloscope.

If a single-ended or power rail probe is connected, the ProbeMeter measures DC voltages between the probe tip and ground connection and enables ground-referenced measurements of voltages.

If a differential probe is connected, you can select if the ProbeMeter measures differential and common mode voltages, or single-ended voltages. See "Display" on page 147.

Remote command:
`PROBe<ch>:PMETer:STATe` on page 1012
`PROBe<ch>:PMETer:RESults:SINGle?` on page 1012
`PROBe<ch>:PMETer:RESults:COMMon?` on page 1013
`PROBe<ch>:PMETer:RESults:DIFFerential?` on page 1013
`PROBe<ch>:PMETer:RESults:NEGative?` on page 1013
`PROBe<ch>:PMETer:RESults:POSitive?` on page 1014

#### 6.4.3.2 Setup for R&S RT-ZD differential probes

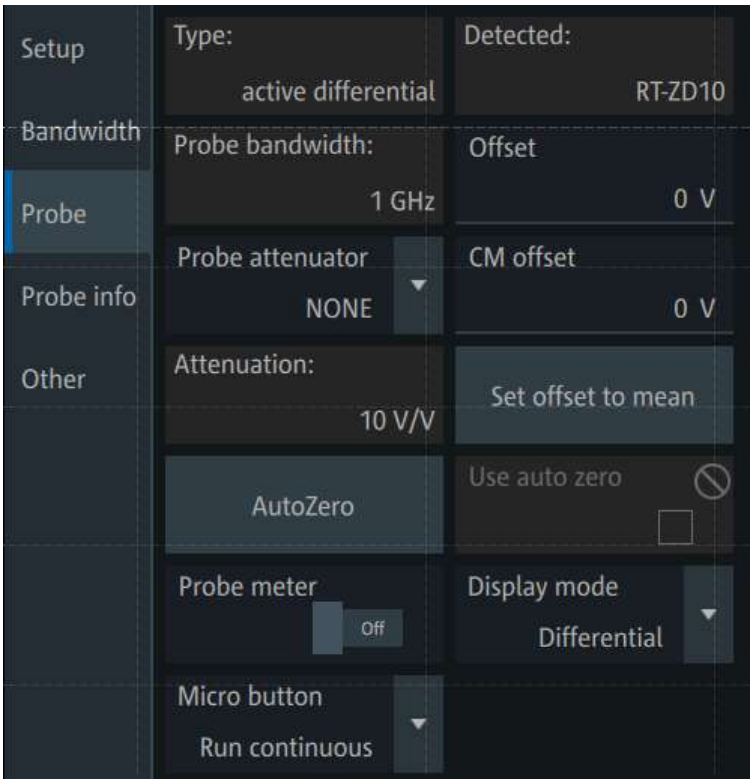



*Figure 6-4: Probe setup for active differential probe R&S RT-ZD30*

Most settings are common settings, which are available for all probe types. See Section 6.4.1, "Common probe settings", on page 141 for description of these settings.

R&S RT-ZD differential probes have the Rohde & Schwarz probe interface and support its functions. For details, see Section 6.4.3.1, "Settings for the Rohde & Schwarz probe interface", on page 144.

The "Offset" is the differential offset, see "Offset" on page 137.

Specific settings for R&S RT-ZD probes are the following:

**CM offset**
Sets the common-mode offset to compensate for a common DC voltage that is applied to both input sockets (referenced to the ground socket). The setting is available for Rohde & Schwarz differential probes and for modular probes in CM measurement mode.

Offset compensation is particularly helpful for measurements on differential signals with high common mode levels, for example, current measurements using a shunt resistor. You can measure the common mode input voltage using the R&S ProbeMeter.

Remote command:
`PROBe<ch>:SETup:CMOFfset` on page 1011

**Probe attenuator**
If you use an external attenuator together with one of the differential active probes R&S RT-ZD10/20/30, select the attenuator used to include its attenuation in the measurements.

Remote command:
`PROBe<ch>:SETup:EPATtenuator` on page 1012

**Display**
Selects the voltage to be measured by the ProbeMeter of differential active probes:

- "Differential / Common Mode":
  Differential voltage is the voltage between the positive and negative signal sockets. Common mode voltage is the mean voltage between the signal sockets and the ground socket. It measures the voltage level relative to ground, for example, to check the operating voltage window.
- "Single Ended Pos/Neg":
  Measures the voltage between the positive/negative signal socket and the ground. The ProbeMeter always measures the common mode and differential voltages. Single-ended voltages are calculated values:
  $V_p = V_{cm} + 0.5 * V_{in}$ and $V_n = V_{cm} - 0.5 * V_{in}$

Remote command:
`PROBe<ch>:SETup:DISPlaydiff` on page 1011

#### 6.4.3.3 Setup for R&S RT-ZPR power rail probes

R&S RT-ZPR power rail probes are designed for power integrity measurements. They can measure small signals in the millivolt range with large DC-offset components.

R&S RT-ZPR power rail probes have the Rohde & Schwarz probe interface and support the ProbeMeter, but not the micro button. For details, see Section 6.4.3.1, "Settings for the Rohde & Schwarz probe interface", on page 144.

R&S RT-ZPR probes require 50 Ω input termination.

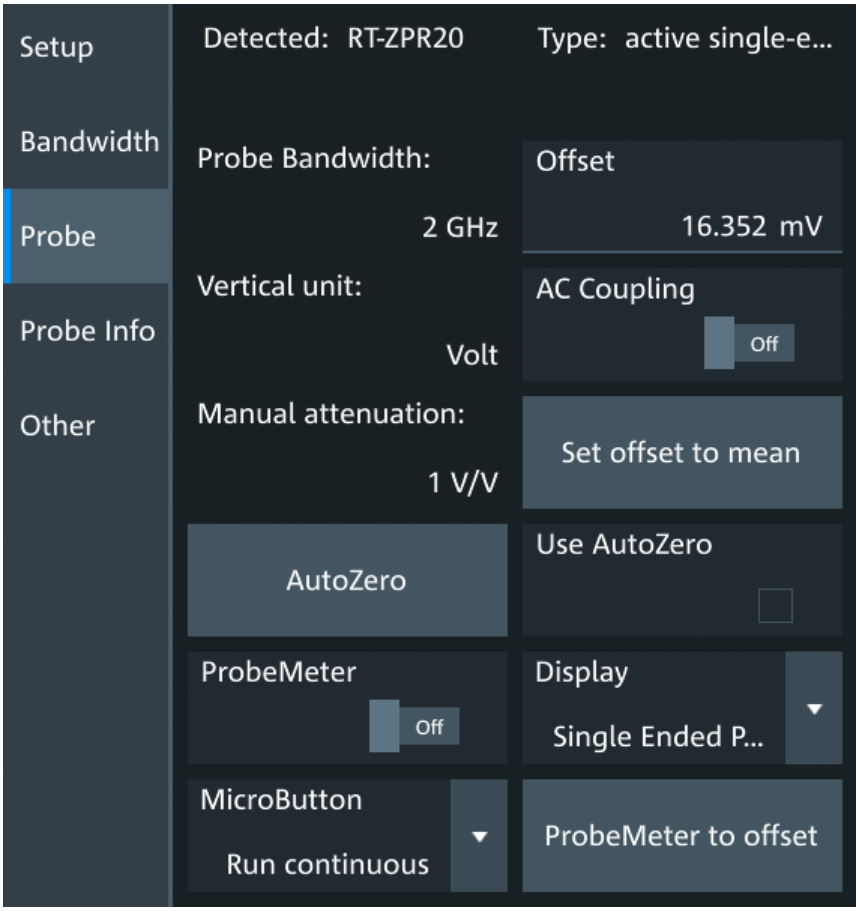



*Figure 6-5: Probe setup for power rail probe R&S RT-ZPR20*

Most settings are common settings, which are available for all probe types. See Section 6.4.1, "Common probe settings", on page 141 for description of these settings.

The "Offset" is the channel offset, see "Offset" on page 137.

Specific settings for R&S RT-ZPR probe are the following:

**AC Coupling**
Enables AC coupling in R&S RT-ZPR power rail probes, which removes DC and very low-frequency components. The R&S RT-ZPR probe requires 50 Ω input termination, for which the channel AC coupling is not available. The probe setting allows AC coupling also at 50 Ω inputs.

Remote command:
`PROBe<ch>:SETup:ACCoupling` on page 1011

**Set offset to ProbeMeter value, ProbeMeter to offset**
Sets the measured R&S ProbeMeter value as offset. Thus, the value is considered in measurements.

The function is available if the ProbeMeter is active.

Remote command:
PROBe<ch>:SETup:ADVanced:PMToffset on page 1015

#### 6.4.3.4 Setup for R&S RT-ZHD high-voltage differential probes

R&S RT-ZHD high-voltage differential probes are designed to measure safely high-voltage floating circuits using a grounded oscilloscope. They extend the measurement capability of oscilloscopes to measure electronic power converters, inverters, motor speed controls, switch mode power supplies and many other applications.

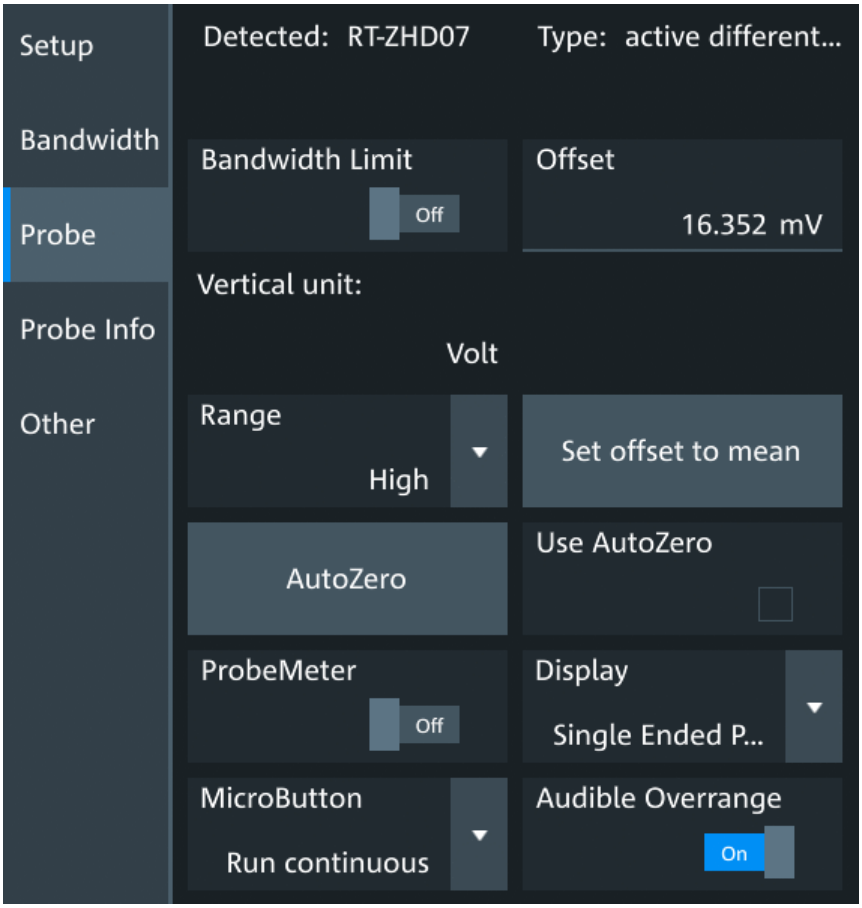



*Figure 6-6: Setup for high-voltage differential probe R&S RT-ZHD07*

Most settings are common settings, which are available for all probe types. See Section 6.4.1, "Common probe settings", on page 141 for description of these settings.

Bandwidth and attenuation values are indicated on the probe control box. The "Offset" is the differential offset, see "Offset" on page 137.

R&S RT-ZHD high-voltage differential probes have the Rohde & Schwarz probe interface and support its functions. For details, see Section 6.4.3.1, "Settings for the Rohde & Schwarz probe interface", on page 144.

Specific settings for R&S RT-ZHD probes are the following:

**Bandwidth limit**
Activates the lowpass filter in the probe control box. The filter frequency depends on the probe type and is indicated on the probe control box.

You can set the filter on the probe control box or at the oscilloscope.

Remote command:
`PROBe<ch>:SETup:ADVanced:FILTer` on page 1014

**Range**
Sets the voltage range of an R&S RT-ZHD probe. You can set the range on the probe control box or at the oscilloscope.

"Auto" The voltage range is set only at the oscilloscope with "Scale".

"Low" Sets the lower voltage range of the connected probe. The range values depend on the probe type and are indicated on the probe control box.

"High" Sets the higher voltage range of the connected probe. The range values depend on the probe type and are indicated on the probe control box.

Remote command:
`PROBe<ch>:SETup:ADVanced:RANGe` on page 1015

**Audible overrange**
Activates the acoustic overrange warning in the probe control box. You can also activate the sound directly on the probe control box.

Remote command:
`PROBe<ch>:SETup:ADVanced:AUDioverload` on page 1014

### 6.4.4 Setup for current probes

The setup and adjustment of current probes depends on the output connector of the probe: BNC or Rohde & Schwarz probe box.

#### Current probes R&S RT-ZCxx

The current probes **R&S RT-ZCxx** have BNC connectors. They are known to the MXO 3 as predefined probes, see "Predefined probe, Type and Name of the probe" on page 141. Demagnetizing and zero adjustment is done on the probe, see the probe's User Manual for details. Make sure to demagnetize and adjust the probe before taking measurements.

#### Current probes R&S RT-ZCxxB

Current probes **R&S RT-ZCxxB** have a Rohde & Schwarz probe interface; they are powered and remotely controlled by the oscilloscope.

When the probe is connected, demagnetization is performed automatically.

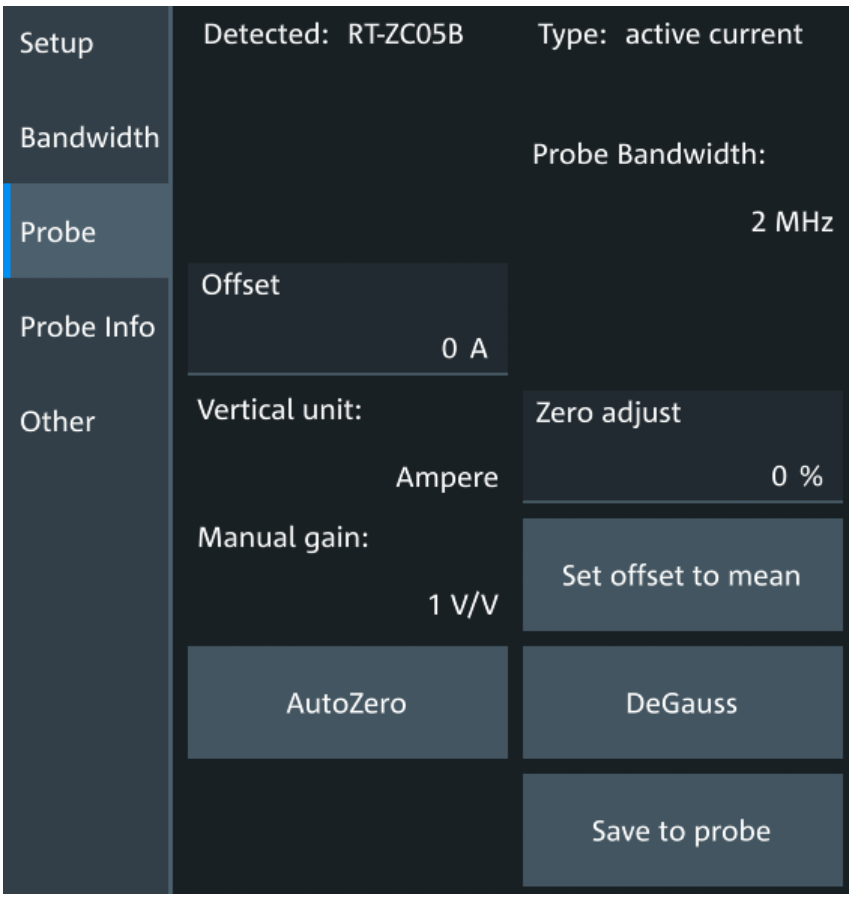



*Figure 6-7: Setup for current probe R&S RT-ZC05B*

Most settings are common settings, which are available for all probe types. See Section 6.4.1, "Common probe settings", on page 141 for description of these settings.

Current probes are adjusted by the following functions.

**Gain, Manual gain**
Shows the gain of the connected probe if the probe is detected or predefined. For unknown current probes, set the correct gain of the probe.

The vertical scaling and measured values are multiplied by this factor so that the displayed values are equal to the actual measured signal values.

Remote command:
`PROBe<ch>:SETup:GAIN:AUTO?` on page 1019
`PROBe<ch>:SETup:GAIN:MANual` on page 1019

**DeGauss**
Demagnetizes the core if it has been magnetized by switching the power on and off, or by an excessive input. Always carry out demagnetizing before measurement.

The demagnetizing process takes about one second. During demagnetizing, a demagnetizing waveform is displayed.

Demagnetizing is done automatically when R&S RT-ZCxxB is connected to the oscilloscope, or when "AutoZero" is performed.

Remote command:
PROBe<ch>:SETup:DEGauss on page 1020

**Zero adjust**
Zero adjust corrects the effect of an offset caused by temperature drift, and compensates for the remanence. The setting is only available if DC coupling is set.

To set the waveform to zero level by the instrument, use "AutoZero". The detected value is displayed.

Alternatively, you can adjust the value manually until the waveform is set to zero level. Make sure to demagnetize the probe before zero adjustment.

The value is given in percent of the maximum range, which is internally defined. The actual setup range depends on the temperature drift, the measured current and other variables, and it can change over time. If you measure high currents, the probe core magnetizes, which impairs the measurement results. Therefore, repeat "AutoZero" before the measurement.

Remote command:
PROBe<ch>:SETup:OFFSet:ZADJust on page 1020

**Save to probe**
Saves the zero adjust value in the probe box. If you connect the probe to another channel or to another Rohde & Schwarz oscilloscope, the value is read out again, and you can use the probe without further adjustment.

Remote command:
PROBe<ch>:SETup:OFFSet:STPRobe on page 1020

### 6.4.5 Modular probes

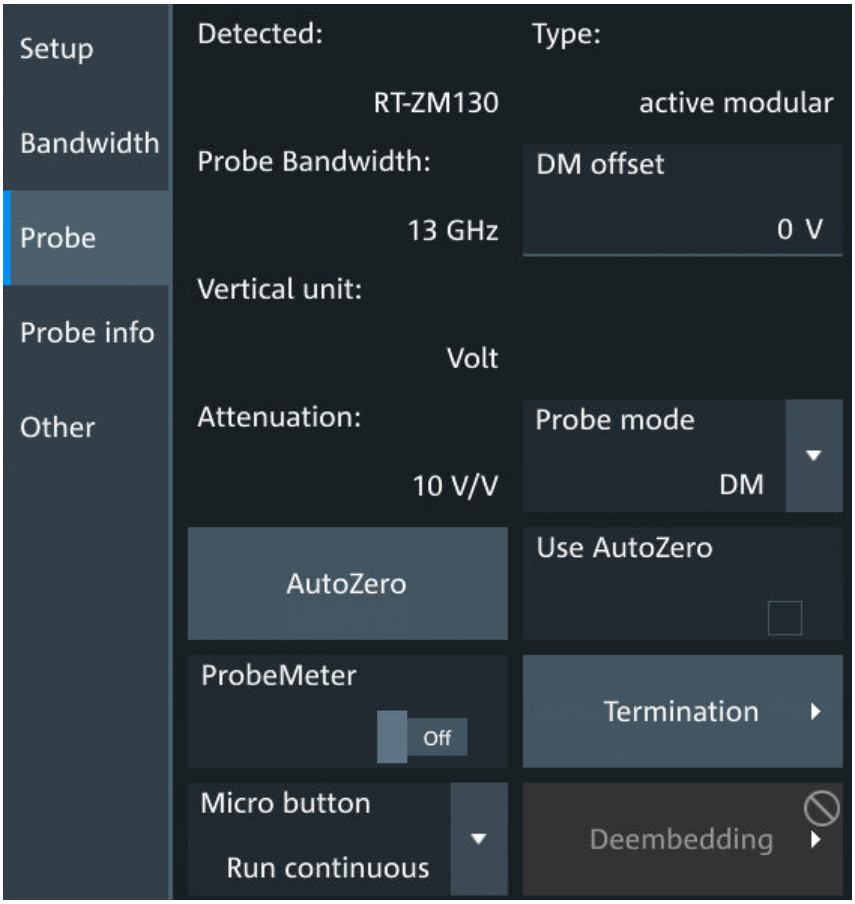
The probes of the R&S RT-ZM family are modular probes. They have a probe head and a probe amplifier connected by a cable, and various probe tip modules and tip cables for different applications. R&S RT-ZM probes are equipped with Rohde & Schwarz probe interface, and provide special features: R&S ProbeMeter, micro button, and a wide offset compensation range.

The following common probe settings are available:

- "Probe bandwidth" on page 142
- "Vertical unit" on page 142
- "Attenuation" on page 142
- "AutoZero, Use AutoZero" on page 143

The R&S RT-ZM probes support the following features of the Rohde & Schwarz probe interface:

- "MicroButton" on page 145
- "ProbeMeter" on page 145




The specific functions of modular probes are described in the following sections:

- Setup parameters of modular probes.....153
- Termination voltage with R&S RT-ZMA40 SMA module.....154

### 6.4.5.1 Setup parameters of modular probes

Access: "Menu" > "Vertical" > "Probe" tab, and R&S RT-ZM probe is connected to the channel

**Probe mode**
Sets the measurement mode of modular probes.

The modular probes of the R&S RT-ZM family have a multimode function. You can switch between single-ended, differential and common mode measurements without reconnecting or resoldering the probe. You can set the probe mode in the dialog, and you can assign the probe mode setting to the micro button.

If you use the R&S RT-ZMA30 browser module, only DM measurements are possible because this module has no ground connector.

The measurement modes are:

"DM" Differential-mode input voltage ($V_{dm}$), the voltage between the positive and negative input terminal.

$$V_{dm} = V_p - V_n$$

"CM" Common-mode input voltage ($V_{cm}$), the mean voltage between the positive and negative input terminal vs. ground.

$$V_{cm} = \frac{V_p + V_n}{2}$$

"P" Positive single-ended input voltage ($V_p$). The voltage between the positive input terminal and the ground.

"N" Negative single-ended input voltage ($V_N$). The voltage between the negative input terminal and the ground.

Remote command:
`PROBe<ch>:SETup:PRMode` on page 1016

**DM offset, CM offset, P offset, N offset**
Compensate offset voltages. The dialog shows the offset of the selected probe mode.

The offset of the selected probe mode is used as the channel offset and considered automatically for correction. For example, in CM mode, the common mode offset is used as the channel offset.

"DM Offset" Compensates a DC voltage applied between the positive ($V_p$) and the negative ($V_n$) input terminal at the probe tip.

"CM Offset" Compensates a DC voltage applied to both input terminals referenced to ground.

"P Offset" Compensates a DC voltage applied to the positive input terminal ($V_p$) referenced to ground.

"N Offset" Compensates a DC voltage applied to the negative input terminal ($V_n$) referenced to ground.

Remote command:
`PROBe<ch>:SETup:DMOFfset` on page 1017
`PROBe<ch>:SETup:CMOFfset` on page 1011
`PROBe<ch>:SETup:NOFFset` on page 1017
`PROBe<ch>:SETup:POFFset` on page 1017

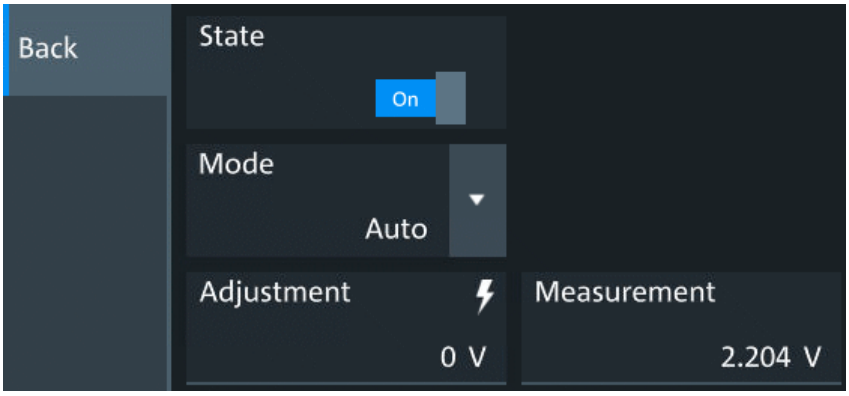
### 6.4.5.2 Termination voltage with R&S RT-ZMA40 SMA module

Access: "Menu" > "Vertical" > "Probe" tab > "Termination"

Termination voltage is relevant if you use the R&S RT-ZMA40 SMA module. The SMA module applies a termination voltage (±4 V) to the DUT to enable measurements against a common mode DC voltage instead of ground. This measurement is required for many digital signal standards.

The termination voltage can be controlled at the oscilloscope. Therefore, connect the $V_T$ terminal of the R&S RT-ZM probe amplifier to the $V_T$ terminal of the R&S RT-ZMA40

SMA module using the red DC lead (see R&S RT-ZM User Manual). The required termination voltage is measured and adjusted automatically, but can also be set manually.




**State**
Activates control of the termination voltage.

Remote command:
`PROBe<ch>:SETup:TERM:STATe` on page 1019

**Mode**
Selects the voltage that is used for termination.

In auto mode, the instrument uses the measured common mode voltage for termination. In manual mode, you can enter the voltage to be used for termination. Use the manual mode if you know the common mode voltage of the DUT.

Remote command:
`PROBe<ch>:SETup:TERM:MODE` on page 1018

**Adjustment**
Sets the voltage to be used for termination correction in manual termination mode.

Remote command:
`PROBe<ch>:SETup:TERM:ADJust` on page 1018

**Measurement**
Shows the measured common mode voltage.

Remote command:
`PROBe<ch>:SETup:TERM:MEASure?` on page 1018

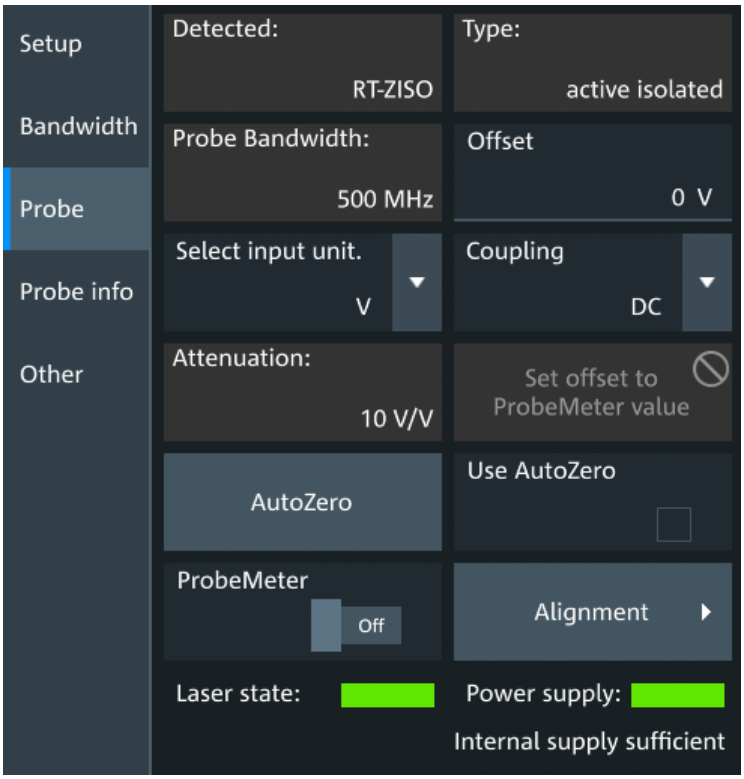
### 6.4.6 Setup for optical isolated probe systems

The R&S RT-ZISO is an optical isolated probe system. It consists of the probe head with exchangeable tip module, probe receiver, and scope connector cable. The probe head is permanently connected to the probe receiver by an optical cable, which is an optical isolated cable with multifiber system. To connect the probe receiver to the MXO 3, the scope connector cable is used. This cable provides the required supply voltage and transmits the measurement data and control signals simultaneously.

The power consumption of the probes is monitored by the power monitor. See Section 6.4.8.1, "Power monitor", on page 161 for details.

### 6.4.6.1 Setup parameters of optical isolated probes

Access: "Menu" > "Vertical" > "Probe" tab > select channel with connected R&S RT-ZISO probe.




Most settings are common settings, which are available for all probe types. See Section 6.4.1, "Common probe settings", on page 141 for description of these settings.

The dialog also shows some vertical settings

- "Offset". See "Offset" on page 137 for details.
- "Coupling". See "Coupling" on page 137 for details.

Set the vertical scale in the "Setup" tab. The probe adjusts its attenuation automatically.

**Select input unit**
Sets the unit of the R&S RT-ZISO signal.

Remote command:
`PROBe<ch>:SETup:ADVanced:UNIT` on page 1021

**Set offset to ProbeMeter value, ProbeMeter to offset**
Sets the measured R&S ProbeMeter value as offset. Thus, the value is considered in measurements.

The function is available if the ProbeMeter is active.

Remote command:
`PROBe<ch>:SETup:ADVanced:PMToffset` on page 1015

**Alignment**
Opens the "Alignment" dialog. See Section 6.4.6.2, "Alignment", on page 157.

**Laser state**
The color indicates the current status of the laser:

- Green: the laser is working.
- Yellow: the laser needs service, but is still working.
- Red: defective laser, send it to your Rohde & Schwarz service center.

Remote command:
`PROBe<ch>:SETup:LASer:STATe?` on page 1021

**Power supply**
Indicates wether the power supply of the R&S RT-ZISO is sufficient.

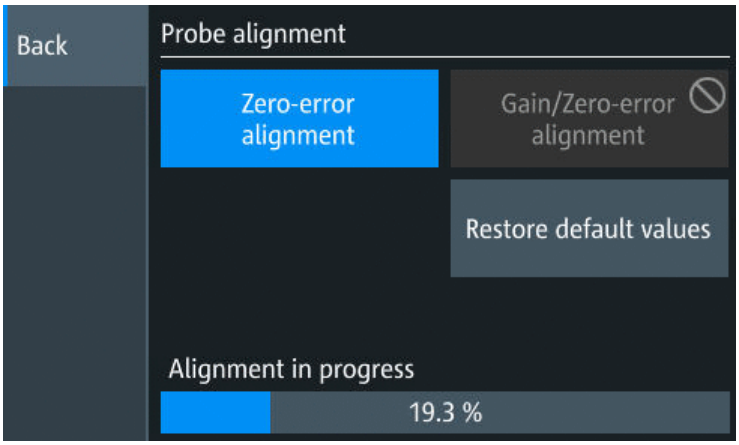
Details on power consumption are given in the power monitor. See Section 6.4.8.1, "Power monitor", on page 161.

### 6.4.6.2 Alignment

Access: "Menu" > "Vertical" > "Probe" tab > select channel with connected R&S RT-ZISO probe > "Alignment".

The dialog provides functions to correct the zero point and gain errors.

Align the probe before first use, and when the ambient conditions have been changed, for example, when strong temperature changes occur (> 5°).




The progress of the alignment process is shown in a progress bar below the buttons.

**Zero-error alignment**
Performs a self-alignment and corrects the zero point error of the R&S RT-ZISO probe.

Tip modules are not affected by the alignment. Attaching or removing the tip module does not change the process or result.

The alignment can take about a minute, the progress is shown in the dialog. Wait until the process has been finished. After successful alignment, the correction data is automatically written to the non-volatile flash of the probe.

Remote command:
`PROBe<ch>:SETup:ALIGnment:ZERO:EXECute` on page 1022

**Gain/Zero-error alignment**
Corrects the zero point error and the gain error of the R&S RT-ZISO probe.

Tip modules are not affected by the alignment. Attaching or removing the tip module does not change the process or result.

The alignment can take about a minute, the progress is shown in the dialog. Wait until the process has been finished. After successful alignment, the correction data is automatically written to the non-volatile flash of the probe.

Remote command:
`PROBe<ch>:SETup:ALIGnment:GAIN:EXECute` on page 1022

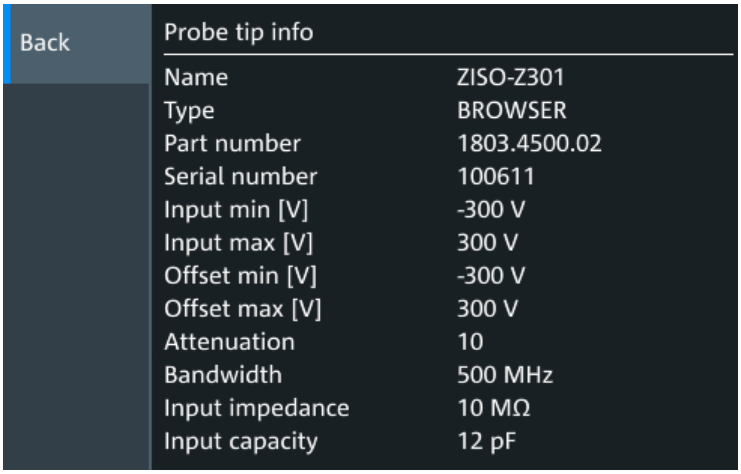
**Restore default values**
Resets the zero point and gain error correction to the factory default values.

Remote command:
`PROBe<ch>:SETup:ADVanced:RDEFaults` on page 1023

### 6.4.6.3 Tip info dialog

Access: "Menu" > "Vertical" > "Probe" > select channel with connected R&S RT-ZISO probe > "Tip info"

The dialog shows the characteristics of the tip modules that are connected to the probes.




Remote commands:

- `PROBe<ch>:SETup:TIPModel:NAME?` on page 1023

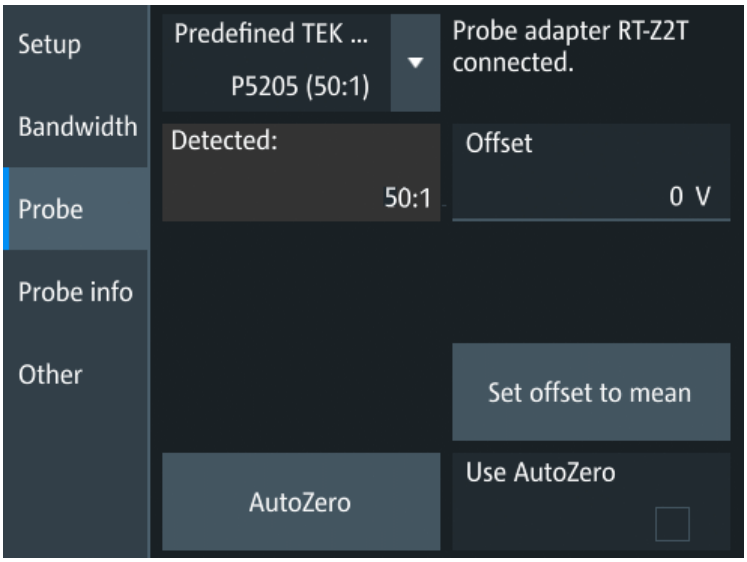
- PROBe<ch>:SETup:TIPModel:INFO? on page 1023

### 6.4.7 Probe adapter R&S RT-Z2T

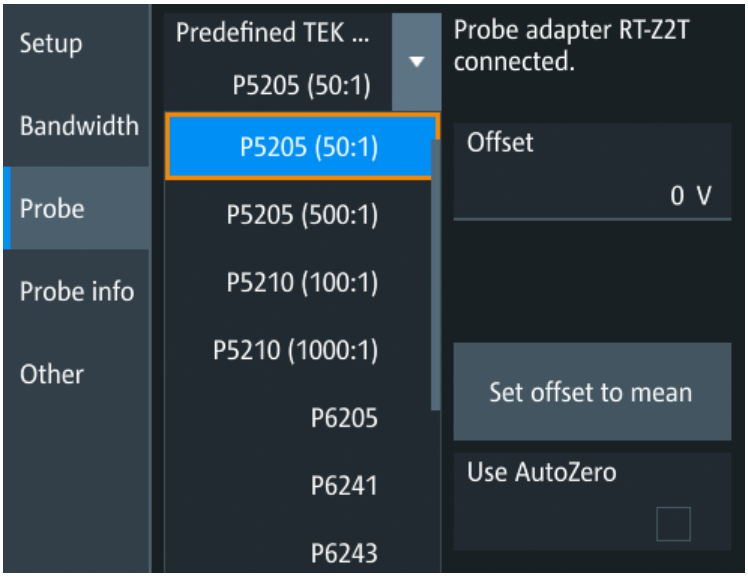
Using the R&S RT-Z2T probe interface adapter, you can connect selected Tektronix active probes with TekProbe BNC™ level II interface.

1. Connect the R&S RT-Z2T adapter to the channel input.
2. Connect the probe to the adapter.

   The instrument identifies the adapter.
3. Select "Menu" > "Vertical" > "Probe" tab.




   The dialog shows that the R&S RT-Z2T probe interface adapter is connected, and supported Tektronix probes are listed as "Predefined probe".
4. Select the "Predefined probe" in the "Probe" tab.



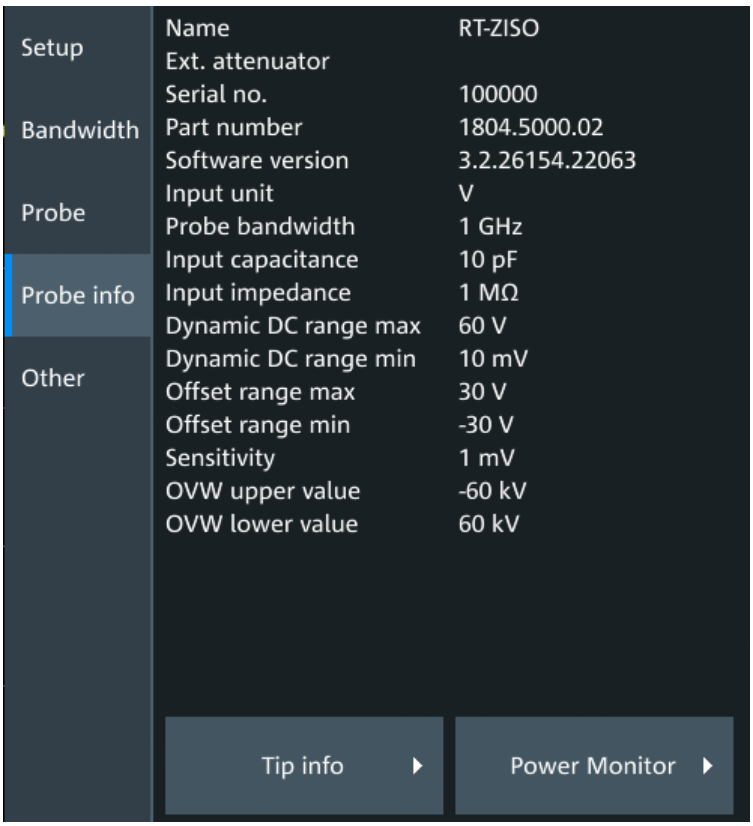

See also: "Predefined probe, Type and Name of the probe" on page 141.

The power consumption of the R&S RT-Z2T probe interface adapter in the "Power monitor": "Vertical" > "Probe Info" > "Power monitor".

### 6.4.8 Probe info

The dialog shows general information on the connected probe, for example, type, serial number, and part number. Below, electrical characteristics are shown, like bandwidth, attenuation, input capacitance and impedance, voltage and DC offset range.

For a specification of the probe parameters, refer to the specifications document.




Remote commands: Section 18.9.6.7, "Probe attributes", on page 1024.

#### 6.4.8.1 Power monitor

The power monitor informs about the connected probes and their power consumption. The maximum available power per rail and per single channel are also shown.

If you use several R&S RT-ZISO probes, the power of the instrument can be insuffcient to power all probes. The power monitor notifies you of insufficient power. To solve the problem, you can remove redundant probes, or use the external power supplies to power the R&S RT-ZISO probes.




#### 6.4.8.2 Tip info

The "TIP info" button and dialog are available if an R&S RT-ZISO probe is connected to the channel. See Section 6.4.6.3, "Tip info dialog", on page 158 for details.

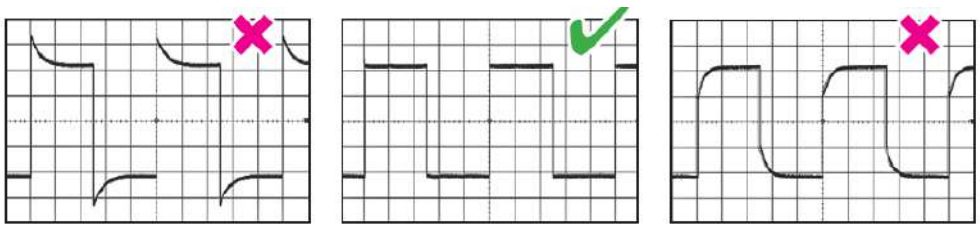
### 6.4.9 Adjusting passive probes

When using a passive probe, you have to compensate it when you connect it to the instrument the first time. Compensation matches the probe cable capacitance to the oscilloscope input capacitance to assure good amplitude accuracy from DC to upper bandwidth limit frequencies. A poorly compensated probe reduces the performance of the probe-oscilloscope system and introduces measurement errors resulting in distorted waveforms and inaccurate results.

Two connector pins are located on the front panel. The ⊥ pin is on ground level. The ⎍ pin supplies a square wave signal with 1 kHz for low frequency probe compensation.

1. Connect the BNC connector of the probe to one of the channel inputs.
2. Connect the probe's ground connector to the ground compensation pin, and the probe tip to the signal pin.
3. Press [Autoset].

   A square wave appears on the display.
4. Adjust the compensation trimmer of the probe to optimum square wave response. For details, refer to the documentation of your probe.



# 7 Trigger

This section describes the trigger capabilities of the MXO 3.


- Basics of triggering....163
- Common trigger settings....165
- Trigger sequence....167
- Trigger types....169
- Trigger mode / holdoff....192
- Hysteresis....194
- Channel filter....196
- Actions on trigger....197
- Zone trigger....199


## 7.1 Basics of triggering

Triggering means to capture the interesting part of the relevant waveforms, and the trigger point is the determining point in the waveform record. Choosing the right trigger type and configuring all trigger settings correctly allows you to detect various incidents in analog, digital, and protocol signals.

**How the oscilloscope triggers**

A trigger occurs if the complete set of trigger conditions is fulfilled. The instrument acquires continuously and keeps the sample points to fill the pre-trigger part of the waveform record. When the trigger occurs, the instrument continues acquisition until the post-trigger part of the waveform record is filled. Then it stops acquiring and waits for the next trigger. When a trigger is recognized, the instrument does not accept another trigger until the acquisition is complete and the holdoff time has expired.

**Trigger setup on the oscilloscope**

A simple trigger setup includes:

- Source of the trigger signal
- Trigger type selection and setup
- Horizontal position of the trigger, see: Section 6.1.1, "About the horizontal system", on page 120
- Trigger mode

The MXO oscilloscopes provide various trigger types for troubleshooting and signal analysis, for example, edge trigger, glitch trigger, interval trigger, pattern trigger, and much more.

For complex tasks like verifying and debugging designs, advanced trigger settings are available:

- Filter to remove high or low frequencies from the trigger signal.
- Hysteresis to avoid unwanted trigger events caused by noise.
- Holdoff to define exactly which trigger event causes the trigger.

- Trigger sequences to combine several event conditions.

**Action on trigger**

A trigger can initiate one or more actions, for example, a trigger out signal, saving a screenshot or saving waveform data. All available actions can be initiated at the same time.

**Trigger sequence**

A trigger sequence joins two or more separate trigger conditions with an optional delay time and an optional reset time or reset condition. Similar setups are also known as multi-step triggers or A/B trigger.

## 7.1.1 Trigger information

Information on the most important trigger settings is shown in the trigger label above the diagram. If you tap the trigger label, the "Trigger" dialog opens.

If you trigger on a single event, the trigger label shows:

- Trigger source
- Trigger type
- Trigger level
- Trigger mode
- Edge or polarity, and important trigger-type specific settings
- Trigger state

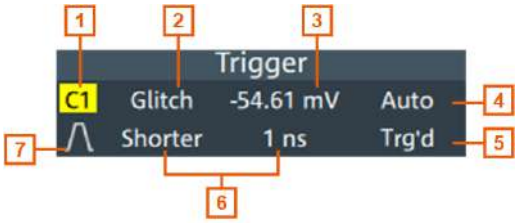



*Figure 7-1: Trigger label on the toolbar*

1 = Trigger source
2 = Trigger type
3 = Trigger level
4 = Trigger mode
5 = Trigger state
6 = Trigger type specific settings
7 = Trigger slope

If you trigger on a sequence, the trigger label shows:

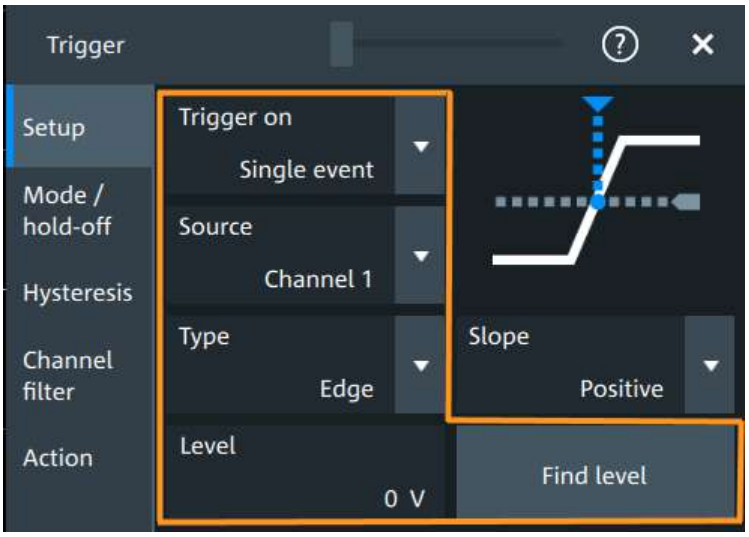
- Sequence type
- Trigger mode
- Trigger state

## 7.2 Common trigger settings

Access: "Menu" > "Trigger" > "Setup" tab

The common trigger settings are the trigger sequence, trigger source, the trigger type, and the trigger level. To set the trigger level automatically, use "Find level".

These common settings are set once for a single event. In a trigger sequence, they are specific for each condition.




Depending on the trigger type, additional settings are needed. They are explained in the trigger-type specific sections, see Section 7.4, "Trigger types", on page 169.

Trigger on ..... 165
Source ..... 165
Type ..... 166
Level ..... 166
Find level ..... 167

**Trigger on**
Selects, if you want to trigger on a single event, or on a series of events.

Remote command:
`TRIGger:MEVents:MODE` on page 1031

**Source**
Selects the source of the trigger signal for the selected trigger event. The trigger source works even if it is not displayed in a diagram.

Available sources depend on the trigger sequence setting. If you trigger on a single event, all inputs can be used as trigger source. If you trigger on a sequence, only analog channels can be set as trigger source for A, B, and R-events.

The trigger source can be:
- Channel <n>: an analog input channel
- Extern: external analog signal connected to the external trigger input. For the external trigger source, the analog edge trigger is available.

- Line: The instrument generates the trigger from the AC power input and synchronizes the signal to the AC power frequency. Use this source if you want to analyze signals related to the power line frequency, such as lighting equipment and power supply devices. For the line trigger source, the edge trigger type is available.
- Digital channels D0 to D15
  Requires option R&S MXO3-B1. Digital channels are only available for triggering on single event. Available trigger types are: edge, width, timeout, state and pattern.
- Serial buses
  If one or more serial protocol options are installed, you can trigger on various conditions in the bus signal. If the hardware trigger is supported for a protocol, triggering on single event is possible. For all protocols, the software trigger is supported in a "A → Serial bus". hardware and software trigger have different trigger settings.

For the state trigger, this source is the "Clock source", the clock signal.

For the setup & hold trigger, this source is the "Data source", the data signal.

Remote command:
`TRIGger:EVENt<ev>:SOURce` on page 1032

**Type**
Selects the trigger type. In a trigger sequence, the trigger type is set for each condition.

The current trigger type is shown on the button and in the trigger label above the diagram.

The following trigger types are available:
- Edge trigger, see page 170
- Glitch trigger, see page 172
- Width trigger, see page 174
- Runt trigger, see page 175
- Window trigger, see page 177
- Timeout trigger, see page 179
- Interval trigger, see page 180
- Slew rate trigger, see page 182
- Setup & Hold, see page 184
- State trigger, see page 186
- Pattern trigger, see page 187

If the external trigger input is used as the trigger source, the analog edge trigger is the only available trigger type.

For digital channels, the edge, width, timeout, state and pattern trigger are available.

Remote command:
`TRIGger:EVENt<ev>:TYPE` on page 1031

**Level**
Sets the voltage level or threshold for the trigger.

You can also drag the trigger level marker on the display, or turn the [Level] knob. To set the trigger level to 50% of the signal amplitude, press the [Level] knob.

For the setup & hold trigger, this level sets the voltage level for the data signal. At this level, the setup and hold time are measured.

For the timeout trigger, the trigger level is the threshold for the high and low signal states.

Runt, window and slew rate triggers require two trigger levels (upper and lower), which are defined as specific settings.

Remote command:
`TRIGger:EVENt<ev>:LEVel<n>[:VALue]` on page 1032
`TRIGger:ANEDge:LEVel` on page 1036 (for external trigger source)

**Find level**
Sets the trigger level automatically to *0.5 * (MaxPeak – MinPeak)*.

In a trigger sequence, "Find level" affects all active events of the sequence (A, B, and R event).

The function is not available for trigger sources "Extern" and "Line".

Remote command:
`TRIGger:FINDlevel` on page 1032

## 7.3 Trigger sequence

With MXO 3, you can trigger on a single trigger event, or on a sequence of events. A trigger sequence consists of at least two event conditions and additional conditions defining when the trigger occurs.

All trigger sequences require that analog input channels C<n> are set as trigger source for the A-trigger. The instrument checks all trigger settings for compatibility and adjusts them if they do not fit.

The following trigger types are only available for triggering on single event:

- Setup & Hold
- State
- Pattern

**A → B → R sequence**

The trigger sequence "A → B → R", for example, consists of two subsequent events: A-trigger and B-trigger with optional B-trigger delay and count. In addition, an optional reset condition R can be configured: timeout or R-trigger condition. A-, B-, and R-triggers are configured in the same way.

After the A-trigger conditions have been met, and an optional delay has passed, the B-trigger with independent conditions is enabled. The instrument waits until one or a specified number of B-trigger conditions occur. If the reset condition is not fulfilled, the latest B-trigger causes the trigger event, and then the sequence starts again. The B-trigger can only cause the trigger event if it occurs after the A-trigger and after the delay time.

If you expect, for example, an irregular B-trigger, you can configure a reset condition to restart the sequence. The reset condition can be a simple timeout, and/or a reset event that is defined in the same way as the A- and B-trigger conditions.

### 7.3.1 Sequence setup

Access: "Menu" > "Trigger" > "Setup" tab > "Trigger on" = "Sequence"

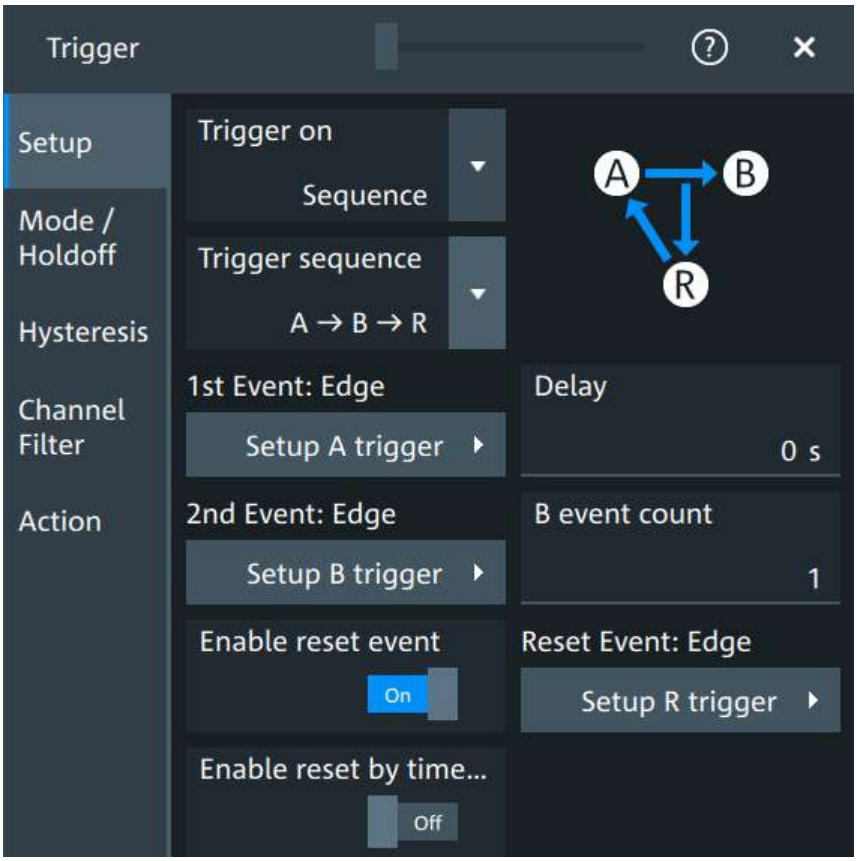



*Figure 7-2: Trigger setup dialog with selected A → B → R sequence*

**Trigger sequence**
Selects the type of the trigger sequence.

"A → B → R" Triggers if all conditions of A- and B-events, as well as additional delay and count, and optional reset timeout and/or R-event conditions are fulfilled.
B-trigger and R-trigger are configured in the same way as the A-trigger. You can configure a delay between the A- and B-trigger, and define a number of fulfilled B-trigger conditions to be ignored. The reset condition R can be a timeout or a trigger condition, or a combination of both.

"A → Zone Trigger" Triggers if the conditions of the A-event and the zone trigger are fulfilled.
See Section 7.9, "Zone trigger", on page 199

"A → Serial bus" Triggers if the conditions of the A-event and the serial bus are fulfilled.
See Section 14.1.6, "Trigger", on page 504.

Remote command:
TRIGger:MEVents:AEVents on page 1033

**Delay**
Sets the time that the instrument waits after an A-trigger until it recognizes B-triggers.

Remote command:
TRIGger:MEVents:SEQuence<se>:DELay on page 1034

**B event count**
Sets the number of B-trigger conditions to be fulfilled after an A-trigger. The last B-trigger causes the trigger event. The waiting time for B-triggers can be restricted with a reset condition: timeout or reset event.

Remote command:
TRIGger:MEVents:SEQuence<se>:COUNt on page 1033

**Enable reset event**
If enabled, the trigger sequence is restarted by the R-trigger condition if the specified number of B-triggers does not occur before the R-trigger conditions are fulfilled.

Remote command:
TRIGger:MEVents:SEQuence<se>:RESet:EVENt on page 1034

**Enable reset by time, Reset timeout**
If reset timeout is enabled, the instrument waits for the "Reset timeout" time for the specified number of B-triggers. If no trigger occurs during that time, the sequence is restarted with the A-trigger.

Remote command:
TRIGger:MEVents:SEQuence<se>:RESet:TIMeout[:ENABle] on page 1035
TRIGger:MEVents:SEQuence<se>:RESet:TIMeout:TIME on page 1034

## 7.4 Trigger types


- Edge trigger....170
- Edge trigger on external trigger source....171
- Glitch trigger....172
- Width trigger....174
- Runt trigger....175
- Window trigger....177
- Timeout trigger....179
- Interval trigger....180
- Slew rate trigger....182
- Setup & Hold....184

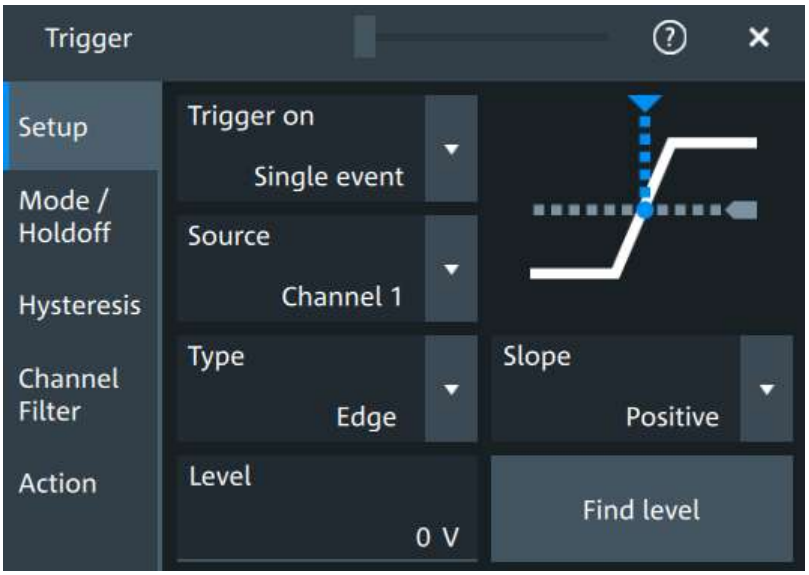
- State trigger.....186
- Pattern trigger.....187
- Line trigger.....192


### 7.4.1 Edge trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Edge"

The edge trigger is the most common trigger type. The trigger occurs when the signal from the trigger source passes the trigger level in the specified direction (slope).

This section describes the trigger-type specific settings. For settings that apply to all trigger types, see Section 7.2, "Common trigger settings", on page 165.




**Level**
See "Level" on page 166.

Remote command:
`TRIGger:EVENt<ev>:LEVel<n>[:VALue]` on page 1032

**Slope**
Sets the edge direction for the trigger.

| | |
|---|---|
| "Positive" | Selects the rising edge, which is a positive voltage change. |
| "Negative" | Selects the falling edge, which is a negative voltage change. |
| "Either" | Selects the rising and falling edge. After starting an acquisition, the instrument triggers on the first identified edge. |

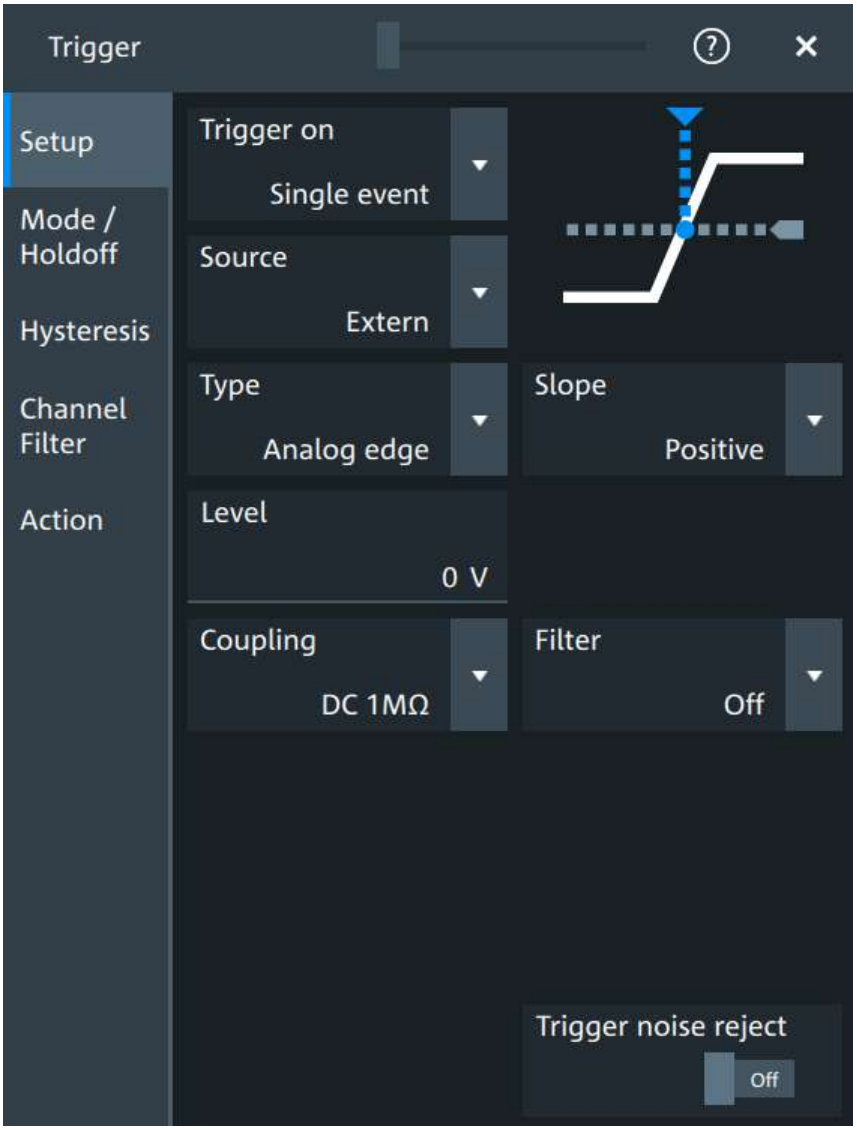
Remote command:
`TRIGger:EVENt<ev>:EDGE:SLOPe` on page 1035

### 7.4.2 Edge trigger on external trigger source

Access: "Menu" > "Trigger" > "Setup" tab > "Source" = "Extern" > "Type = Analog Edge"

If an external trigger signal is connected to the Trigger In connector, and the trigger source is set to "Extern", the analog edge trigger is available. Triggering on an external source is only possible if you trigger on a single event but not for sequences.

This section describes the trigger-type specific settings. For settings that apply to all trigger types, see Section 7.2, "Common trigger settings", on page 165.




The "Slope" and "Level" are the same settings as for the edge trigger, see:

- "Level" on page 166
- "Slope" on page 170

Specific settings for the analog edge trigger are the following:

**Coupling**
Sets the connection of the external trigger signal, i.e. the input impedance and a termination. The coupling determines what part of the signal is used for triggering.

"DC 50 Ω" Connection with 50 Ω termination, passes both DC and AC components of the signal.

"DC 1 MΩ" Connection with 1 MΩ termination, passes both DC and AC components of the signal.

"AC 1 MΩ" Connection with 1 MΩ termination through DC capacitor, removes DC and very low-frequency components. The waveform is centered on zero volts.

Remote command:
`TRIGger:ANEDge:COUPling` on page 1036

**Filter, Cut-off**
Selects the filter mode for the external trigger signal, and sets the cut-off frequency.

"Off" The trigger signal is not filtered.

"LF reject" Frequencies lower than the "Cut-off" frequency are rejected, higher frequencies pass the filter.

"RF reject" Frequencies higher than the "Cut-off" frequency are rejected, lower frequencies pass the filter.

Remote command:
`TRIGger:ANEDge:FILTer` on page 1036
`TRIGger:ANEDge:CUToff:HIGHpass` on page 1036
`TRIGger:ANEDge:CUToff:LOWPass` on page 1037

**Trigger noise reject**
Enables an automatic hysteresis on the trigger level to avoid unwanted trigger events caused by noise.

Remote command:
`TRIGger:ANEDge:NREJect` on page 1037

### 7.4.3 Glitch trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Glitch"

The glitch trigger detects pulses shorter or longer than a specified time. It identifies deviation from the nominal data rate and helps to analyze causes of even rare glitches and their effects on other signals.

This section describes the trigger-type specific settings. For settings that apply to all trigger types, see Section 7.2, "Common trigger settings", on page 165.




**Level**
See "Level" on page 166.

Remote command:
`TRIGger:EVENt<ev>:LEVel<n>[:VALue]` on page 1032

**Polarity**
Sets the polarity of a pulse, which is the direction of the first pulse slope.

| | |
|---|---|
| "Positive" | Selects positive going pulses, the width is defined from the rising to the falling slopes. |
| "Negative" | Selects negative going pulses, the width is defined from the falling to the rising slopes. |
| "Either" | Selects both positive and negative going pulses. |

Remote command:
`TRIGger:EVENt<ev>:GLITch:POLarity` on page 1037
`TRIGger:EVENt<ev>:RUNT:POLarity` on page 1041

**Range**
Selects how the time limit of the runt pulse is defined.

Remote command:
`TRIGger:EVENt<ev>:GLITch:RANGe` on page 1038

**Width**
Sets the length of a glitch. The instrument triggers on pulses shorter or longer than this value, depending on the value set with "Range".

You need to know the expected pulse widths of the circuit to set the glitch width correctly.

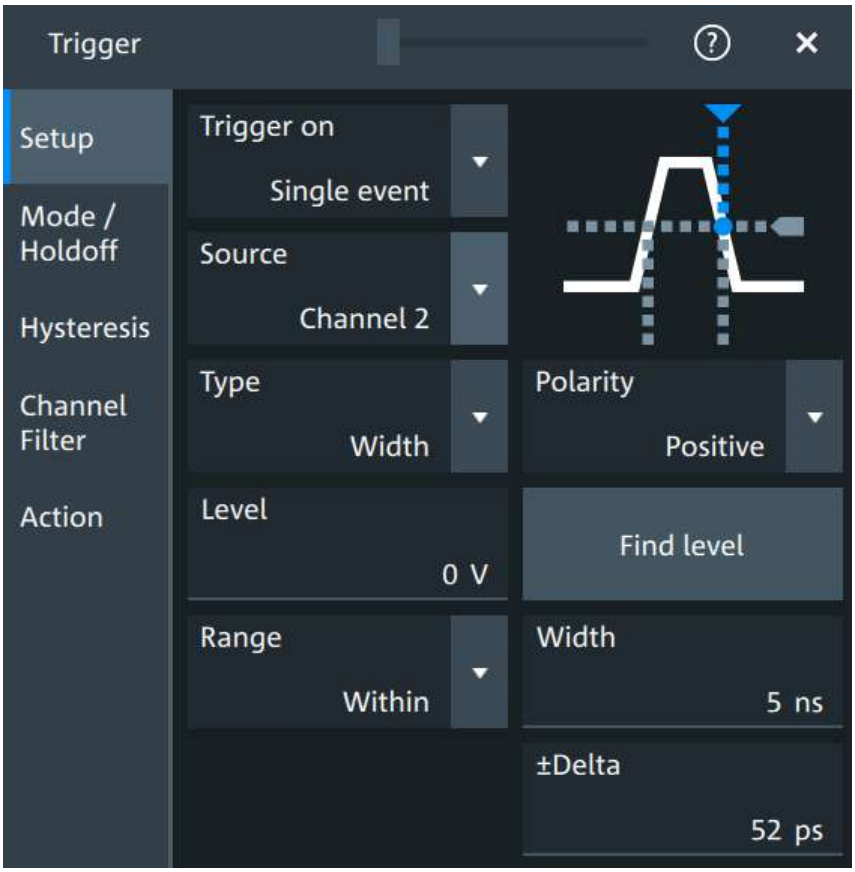
Remote command:
`TRIGger:EVENt<ev>:GLITch:WIDTh` on page 1038

### 7.4.4 Width trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Width"

The width trigger compares the pulse width (duration) with given time limits. It detects pulses with an exact pulse width, pulses shorter or longer than a given time, and also pulses inside or outside the allowable time range. The pulse width is measured at the trigger level.

This section describes the trigger-type specific settings. For settings that apply to all trigger types, see Section 7.2, "Common trigger settings", on page 165.




**Level**
See "Level" on page 166.

Remote command:
`TRIGger:EVENt<ev>:LEVel<n>[:VALue]` on page 1032

**Polarity**
Sets the polarity of a pulse, which is the direction of the first pulse slope.

"Positive" Selects positive going pulses, the width is defined from the rising to the falling slopes.

"Negative" Selects negative going pulses, the width is defined from the falling to the rising slopes.

"Either" Selects both positive and negative going pulses.

Remote command:
`TRIGger:EVENt<ev>:WIDTh:POLarity` on page 1039

**Range**
Selects how the range of a pulse width is defined.

"Longer" Triggers on pulses longer than the given "Width".

"Shorter" Triggers on pulses shorter than the given "Width".

"Within" Triggers on pulses inside a given range. The range of the pulse width is defined by "±Delta" related to "Width".

"Outside" Triggers on pulses outside a given range. The range definition is the same as for "Within" range.

Remote command:
`TRIGger:EVENt<ev>:WIDTh:RANGe` on page 1039

**Width**
For the ranges "Within" and "Outside", the width defines the center of a range which is defined by the limits "±Delta".

For the ranges "Shorter" and "Longer", the width defines the maximum and minimum pulse width, respectively.

Remote command:
`TRIGger:EVENt<ev>:WIDTh:WIDTh` on page 1039

**±Delta**
Defines a range around the width value.

The combination "Range" = "Within" and "±Delta" = 0 triggers on pulses with a pulse width that equals "Width".

The combination "Range" = "Outside" and "±Delta" = 0 means to trigger on pulse widths ≠ "Width".

Remote command:
`TRIGger:EVENt<ev>:WIDTh:DELTa` on page 1039

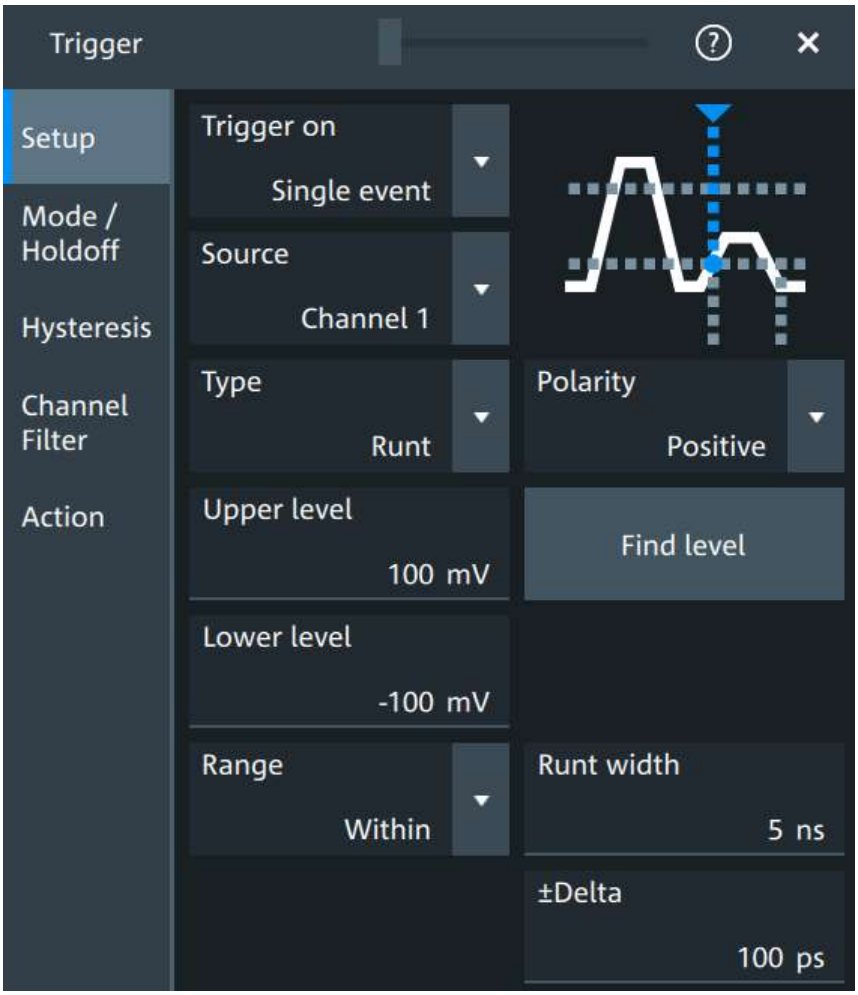
### 7.4.5 Runt trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Runt"

A runt is a pulse lower than normal in amplitude. The amplitude crosses the first threshold twice in succession without crossing the second one. For example, this trig-

ger can detect logic, digital, and analog signals remaining below a specified threshold amplitude because I/O ports are in undefined state.

This section describes the trigger-type specific settings. For settings that apply to all trigger types, see Section 7.2, "Common trigger settings", on page 165.




**Polarity**
Sets the polarity of a pulse, which is the direction of the first pulse slope.

"Positive" Selects positive going pulses, the width is defined from the rising to the falling slopes.

"Negative" Selects negative going pulses, the width is defined from the falling to the rising slopes.

"Either" Selects both positive and negative going pulses.

Remote command:
`TRIGger:EVENt<ev>:GLITch:POLarity` on page 1037
`TRIGger:EVENt<ev>:RUNT:POLarity` on page 1041

**Upper level**
Sets the upper voltage limit.

Remote command:
TRIGger:EVENt<ev>:LEVel<n>:RUNT:UPPer on page 1040

**Lower level**
Sets the lower voltage limit.

Remote command:
TRIGger:EVENt<ev>:LEVel<n>:RUNT:LOWer on page 1040

**Range**
Selects how the time limit of the runt pulse is defined.

| | |
|---|---|
| "Any runt" | Triggers on all runts fulfilling the level condition, without time limitation. |
| "Longer" | Triggers on runts longer than the given "Runt width". |
| "Shorter" | Triggers on runts shorter than the given "Runt width". |
| "Within" | Triggers if the runt length is inside a given time range. The range is defined by "Runt width" and "±Delta". |
| "Outside" | Triggers if the runt length is outside a given time range. The range definition is the same as for "Within" range. |

Remote command:
TRIGger:EVENt<ev>:RUNT:RANGe on page 1041

**Runt width**
For the ranges "Shorter" and "Longer", the runt width defines the maximum and minimum pulse width, respectively.

For the ranges "Within" and "Outside", the runt width defines the center of a range which is defined by "±Delta".

Remote command:
TRIGger:EVENt<ev>:RUNT:WIDTh on page 1042

**±Delta**
Defines a range around the runt width value.

Remote command:
TRIGger:EVENt<ev>:RUNT:DELTa on page 1041

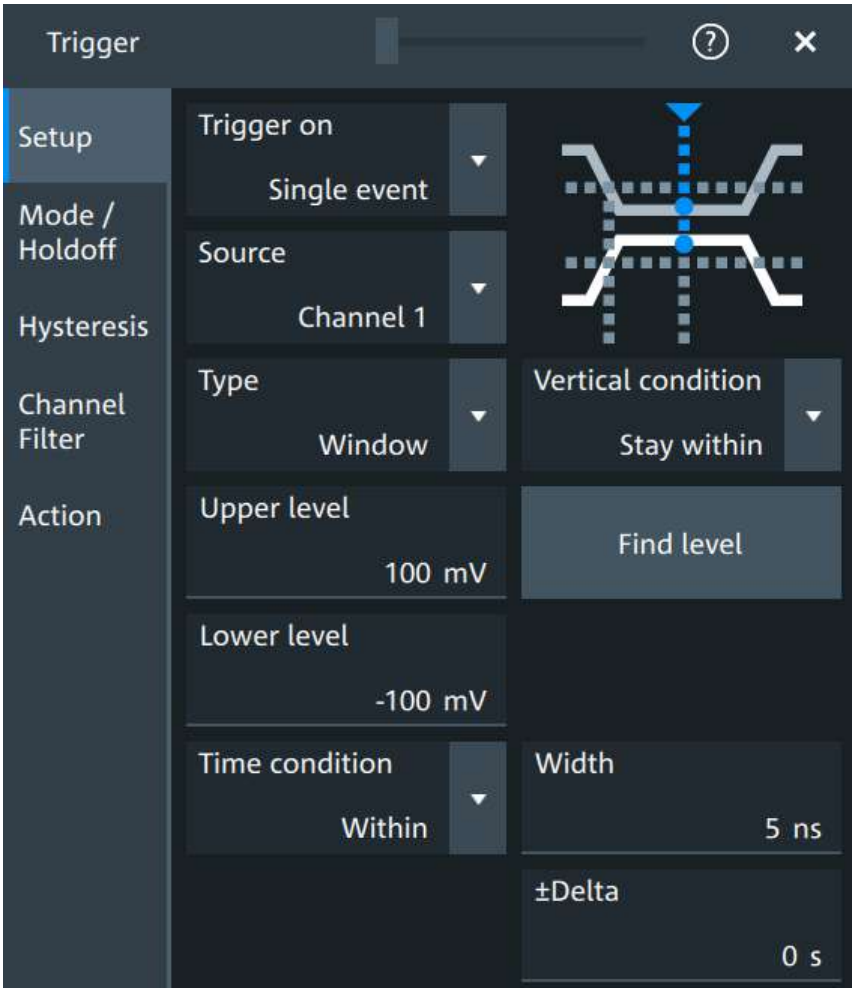
### 7.4.6 Window trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Window"

The window trigger checks the signal run in relation to a "window". The window is formed by the upper and lower voltage levels. The trigger condition is fulfilled, if the waveform enters or leaves the window, or if the waveform stays inside or outside for a time longer or shorter than specified.

With the window trigger, you can display longer transient effects.

This section describes the trigger-type specific settings. For settings that apply to all trigger types, see Section 7.2, "Common trigger settings", on page 165.




**Vertical condition**
Selects how the signal run is compared with the window.

"Enter" Triggers when the signal crosses the upper or lower level and thus enters the window made up of these two levels.

"Exit" Triggers when the signal leaves the window.

"Stay within" Triggers if the signal stays between the upper and lower level for a specified time. The time is defined in various ways by the "Time condition".

"Stay outside" Triggers if the signal stays above the upper level or below the lower level for a specified time. The time is also defined by the "Time condition".

Remote command:
`TRIGger:EVENt<ev>:WINDow:RANGe` on page 1044

**Upper level**
Sets the upper voltage limit.

Remote command:
`TRIGger:EVENt<ev>:LEVel<n>:WINDow:UPPer` on page 1043

**Lower level**
Sets the lower voltage limit.

Remote command:
TRIGger:EVENt<ev>:LEVel<n>:WINDow:LOWer on page 1043

**Time condition**
Available for "Vertical condition" = "Stay within"/"Stay outside".

Selects how the time limit of the window is defined.

| | |
|---|---|
| "Within" | Triggers if the signal stays inside or outside the vertical window limits at least for the time *Width - Delta* and for *Width + Delta* at the most. |
| "Outside" | "Outside" is the opposite definition of "Within". The instrument triggers if the signal stays inside or outside the vertical window limits for a time shorter than *Width - Delta* or longer than *Width + Delta*. |
| "Shorter" | Triggers if the signal crosses vertical limits before the specified "Width" time is reached. |
| "Longer" | Triggers if the signal crosses vertical limits after the specified "Width" time is reached. |

Remote command:
TRIGger:EVENt<ev>:WINDow:TIME on page 1044

**Width**
For the ranges "Within" and "Outside", the width defines the center of a time range which is defined by the limits "±Delta".

For the ranges "Shorter" and "Longer", it defines the maximum and minimum time lapse, respectively.

Remote command:
TRIGger:EVENt<ev>:WINDow:WIDTh on page 1045

**±Delta**
Defines a range around the width value.

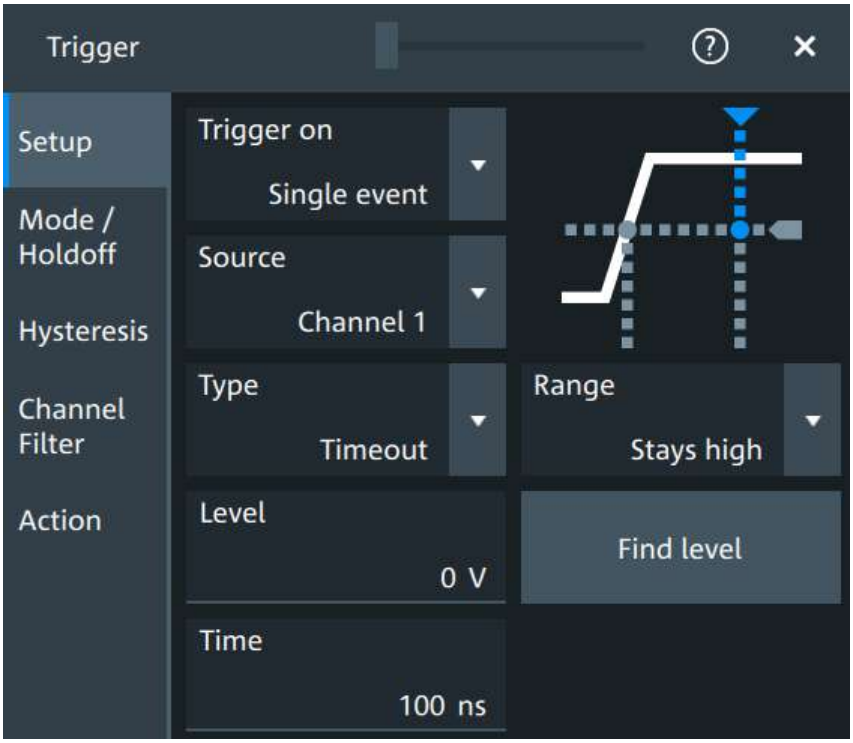
Remote command:
TRIGger:EVENt<ev>:WINDow:DELTa on page 1043

### 7.4.7 Timeout trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Timeout"

The timeout trigger checks if the signal stays above or below the threshold voltage for a specified time lapse. In other words, the trigger occurs if the trigger source does not have the expected transition within the specified time.

This section describes the trigger-type specific settings. For settings that apply to all trigger types, see Section 7.2, "Common trigger settings", on page 165.




**Level**
Sets the threshold for the high and low signal states.

Remote command:
`TRIGger:EVENt<ev>:LEVel<n>[:VALue]` on page 1032

**Range**
Sets the relation of the signal level to the trigger level for the timeout trigger.

"Stays high" The signal level stays above the trigger level.

"Stays low" The signal level stays below the trigger level.

"High or low" The signal level stays above or below the trigger level.

Remote command:
`TRIGger:EVENt<ev>:TIMeout:RANGe` on page 1045

**Time**
Sets the time limit for the timeout at which the instrument triggers.

Remote command:
`TRIGger:EVENt<ev>:TIMeout:TIME` on page 1046

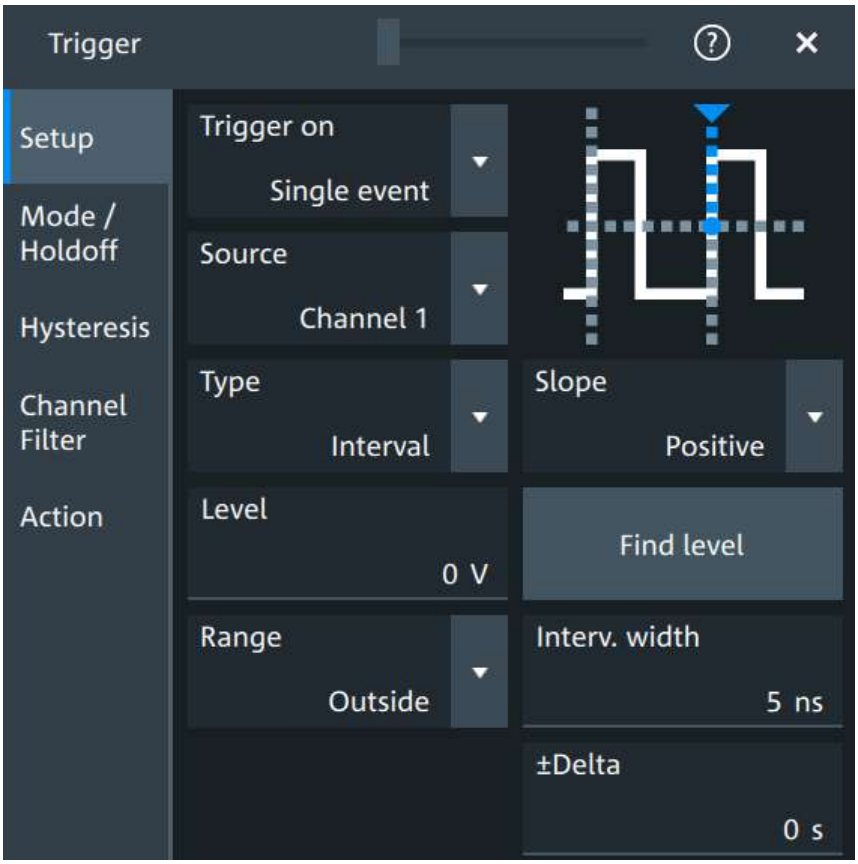
### 7.4.8 Interval trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Interval"

The interval trigger analyzes the time between two pulses.

The interval trigger can analyze either rising or falling edges, but searching for an interval is also possible for both edges at the same time ("Either").

This section describes the trigger-type specific settings. For settings that apply to all trigger types, see Section 7.2, "Common trigger settings", on page 165.




**Level**
See "Level" on page 166.

Remote command:
`TRIGger:EVENt<ev>:LEVel<n>[:VALue]` on page 1032

**Slope**
Sets the edge for the trigger. You can analyze the interval between positive edges or between negative edges.

Remote command:
`TRIGger:EVENt<ev>:INTerval:SLOPe` on page 1047

**Range**
Selects how the range of an interval is defined:

"Within" Triggers on pulse intervals inside a given range. The range is defined by "Interv. width" and "±Delta".

"Outside" Triggers on intervals outside a given range. The range definition is the same as for "Within" range.

"Shorter" Triggers on intervals shorter than the given "Interv. width".

"Longer" Triggers on intervals longer than the given "Interv. width".

Remote command:
`TRIGger:EVENt<ev>:INTerval:RANGe` on page 1047

**Interv. width**
Sets the time between two pulses for the interval trigger.

Remote command:
`TRIGger:EVENt<ev>:INTerval:WIDTh` on page 1047

**±Delta**
Defines a range around the "Interv. width" value.

Remote command:
`TRIGger:EVENt<ev>:INTerval:DELTa` on page 1046

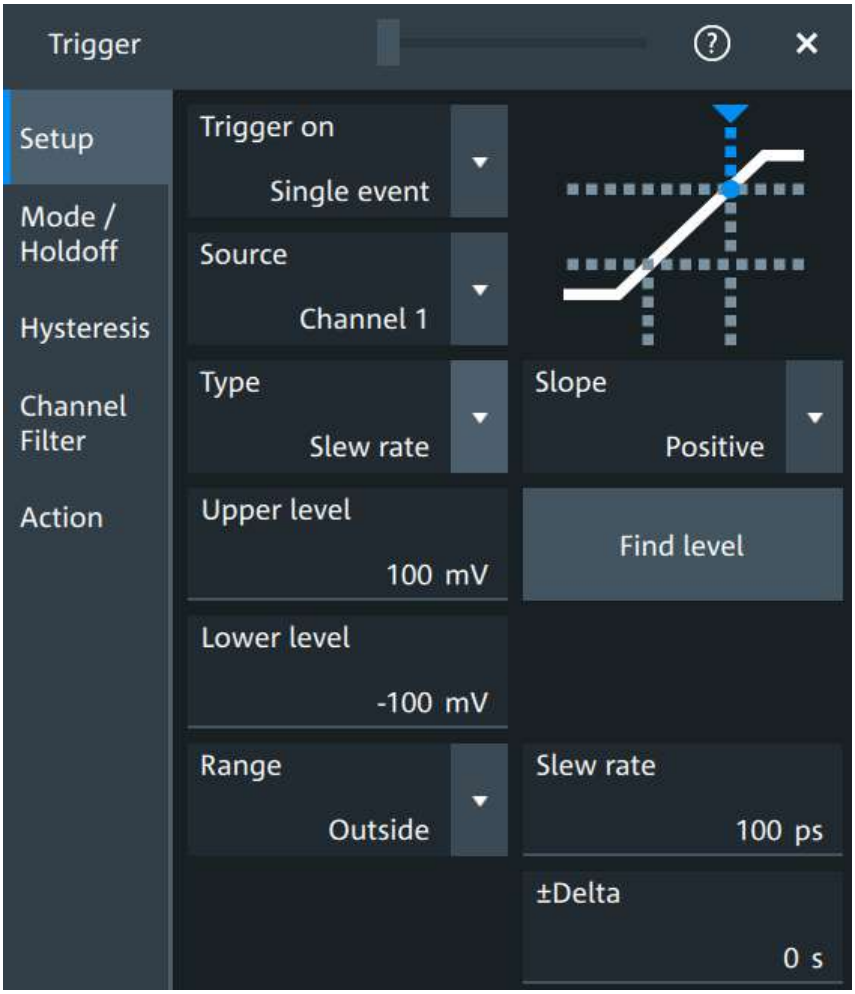
### 7.4.9 Slew rate trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Slew rate"

The slew rate trigger is also known as transition trigger. It triggers if the transition time from the lower to higher voltage level (or vice versa) is shorter or longer as defined, or outside or inside a specified time range.

The slew rate trigger finds slew rates faster than expected or permissible to avoid overshooting and other interfering effects. It also detects slow edges violating the timing in pulse series.

This section describes the trigger-type specific settings. For settings that apply to all trigger types, see Section 7.2, "Common trigger settings", on page 165.




**Slope**
Sets the edge direction for the trigger.

| | |
|---|---|
| "Positive" | Selects the rising edge, which is a positive voltage change. |
| "Negative" | Selects the falling edge, which is a negative voltage change. |
| "Either" | Selects the rising and falling edge. After starting an acquisition, the instrument triggers on the first identified edge. |

Remote command:
`TRIGger:EVENt<ev>:SLEW:SLOPe` on page 1050

**Upper level**
Sets the upper voltage threshold. When the signal crosses this level, the slew rate measurement starts or stops depending on the selected slope.

Remote command:
`TRIGger:EVENt<ev>:LEVel<n>:SLEW:UPPer` on page 1048

**Lower level**
Sets the lower voltage threshold. When the signal crosses this level, the slew rate measurement starts or stops depending on the selected slope.

Remote command:
TRIGger:EVENt<ev>:LEVel<n>:SLEW:LOWer on page 1048

**Range**
Selects how the time limit for the slew rate is defined. The time measurement starts when the signal crosses the first trigger level - the upper or lower limit depending on the selected slope. The measurement stops when the signal crosses the second level.

"Within" Triggers on slew rates inside a given time range. The range is defined by "Slew rate" and "±Delta".

"Outside" Triggers on slew rates outside a given time range. The range definition is the same as for "Within" range.

"Shorter" Triggers on slew rates shorter than the given "Slew rate" limit.

"Longer" Triggers on slew rates longer than the given "Slew rate" limit.

Remote command:
TRIGger:EVENt<ev>:SLEW:RANGe on page 1049

**Slew rate**
For the ranges "Within" and "Outside", the slew rate defines the center of a range which is defined by the limits "±Delta".

For the ranges "Shorter" and "Longer", the slew rate defines the maximum and minimum slew rate limits, respectively.

Remote command:
TRIGger:EVENt<ev>:SLEW:RATE on page 1050

**±Delta**
Defines a time range around the given slew rate.

Remote command:
TRIGger:EVENt<ev>:SLEW:DELTa on page 1049

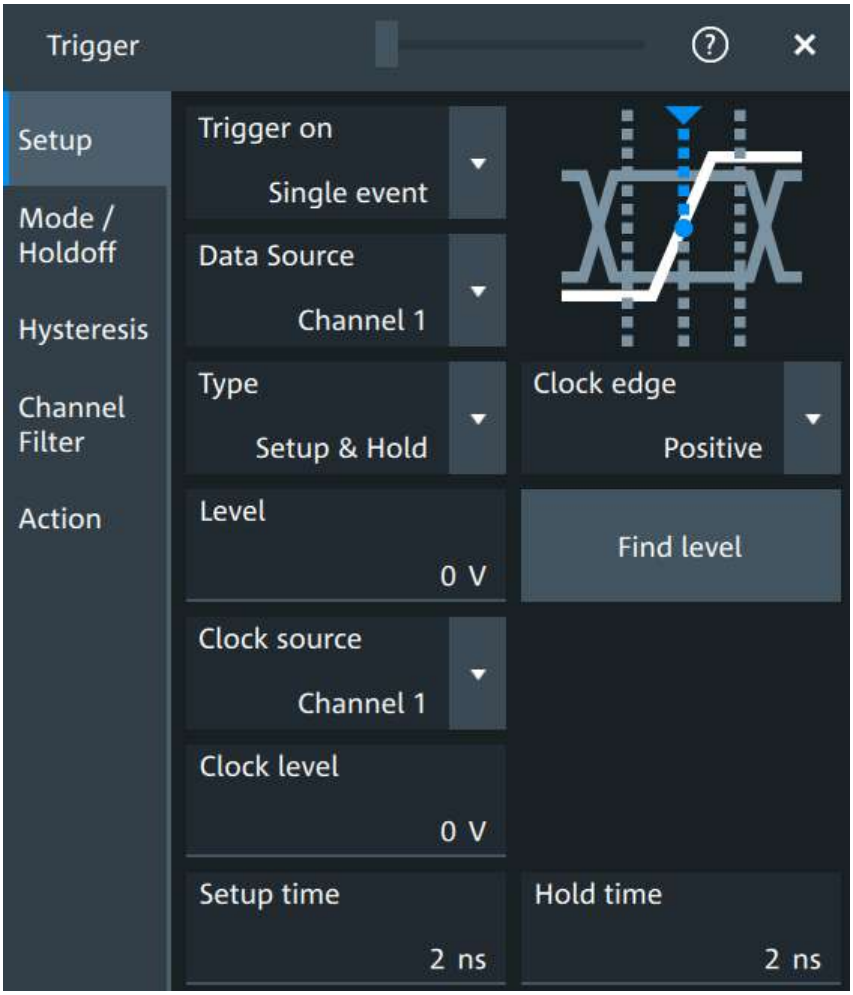
### 7.4.10 Setup & Hold

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Setup & Hold"

The setup & hold trigger analyzes the relative timing between two signals: a data signal and the synchronous clock signal. Many systems require, that the data signal must be steady for some time before and after the clock edge, for example, the data transmission on parallel interfaces. With this trigger type, you can also test the time correlation of sideband and in-band signals.

The setup & hold trigger is also known as Data2Clock trigger.

This section describes the trigger-type specific settings. For settings that apply to all trigger types, see Section 7.2, "Common trigger settings", on page 165.




**Data source**
Selects the input channel of the data signal.

Remote command:
`TRIGger:EVENt<ev>:SOURce` on page 1032

**Level**
Sets the voltage level for the data signal. At this level, the setup and hold time are measured.

Remote command:
`TRIGger:EVENt<ev>:LEVel<n>[:VALue]` on page 1032

**Clock source**
Selects the input channel of the clock signal.

Remote command:
`TRIGger:EVENt<ev>:SETHold:CSOurce[:VALue]` on page 1051

**Clock edge**
Sets the edge of the clock signal. Edge and level define the time reference point.

"Positive" Rising edge, a positive voltage change.

"Negative" Falling edge, a negative voltage change.

"Both" Both the rising and the falling edge.

Remote command:
`TRIGger:EVENt<ev>:SETHold:CSOurce:EDGE` on page 1050

**Clock level**
Sets the voltage level for the clock signal.

Both the clock level and the clock edge define the starting point for calculation of the setup and hold time.

Remote command:
`TRIGger:EVENt<ev>:SETHold:CSOurce:LEVel` on page 1051

**Setup time**
Sets the minimum time **before** the clock edge while the data signal must stay steady above or below the data level.

Remote command:
`TRIGger:EVENt<ev>:SETHold:STIMe` on page 1052

**Hold time**
Sets the minimum time **after** the clock edge while the data signal must stay steady above or below the data level.

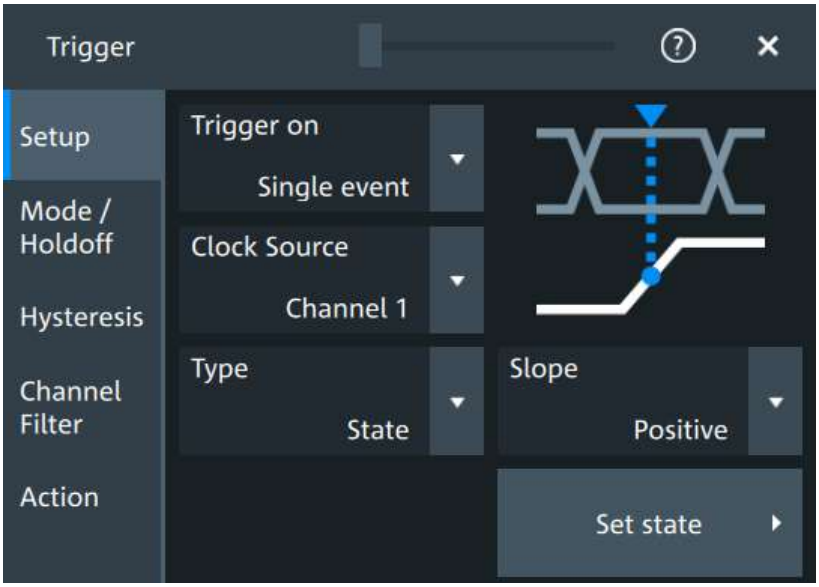
Remote command:
`TRIGger:EVENt<ev>:SETHold:HTIMe` on page 1051

### 7.4.11 State trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Type = State"

The state trigger verifies if the channel states match the defined pattern at the clock edge. The trigger occurs if the logical combination of the input channels is true at the crossing point of the selected clock edge and the trigger level.

This section describes the trigger-type specific settings. For settings that apply to all trigger types, see Section 7.2, "Common trigger settings", on page 165.




1. Select the "Clock source" and the "Slope".
2. Tap "Set state".
   See Section 7.4.12.2, "Settings for state and pattern trigger of analog channels", on page 189 for setting details.

**Slope**
Sets the edge direction for the trigger.

| | |
|---|---|
| "Positive" | Selects the rising edge, which is a positive voltage change. |
| "Negative" | Selects the falling edge, which is a negative voltage change. |
| "Either" | Selects the rising and falling edge. After starting an acquisition, the instrument triggers on the first identified edge. |

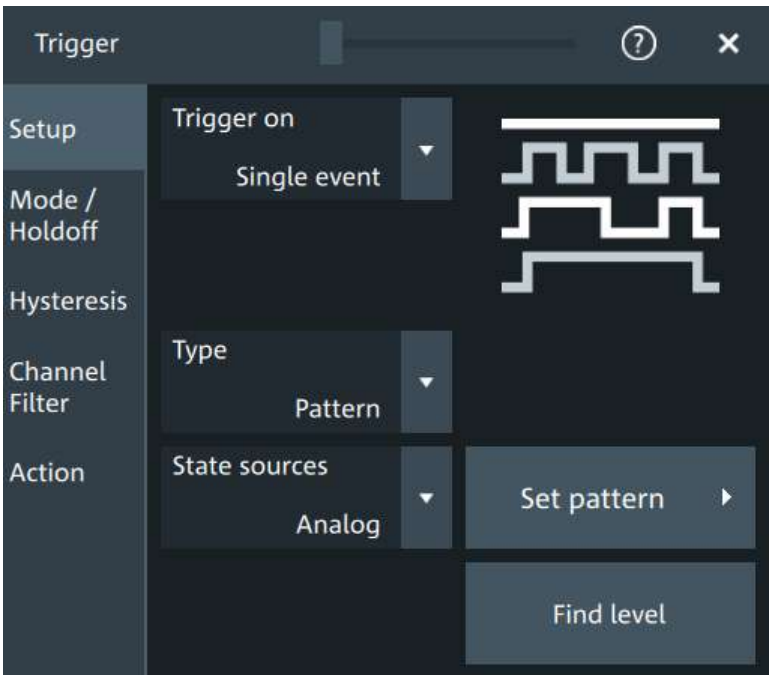
Remote command:
`TRIGger:EVENt<ev>:STATe:SLOPe` on page 1053

## 7.4.12 Pattern trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Pattern"

The pattern trigger works like a logic trigger. It provides logical combinations of the input channels and can be used for verifying the operation of digital logic. If the channel states match the desired pattern, the pattern trigger occurs.

This section describes the trigger-type specific settings. For settings that apply to all trigger types, see Section 7.2, "Common trigger settings", on page 165.




► Tap "Set pattern".
See Section 7.4.12.2, "Settings for state and pattern trigger of analog channels", on page 189 for setting details.

### 7.4.12.1 General settings

**State sources**
Selects if the source of the trigger for the pattern type is an "Analog" or "Digital" channel.

Remote command:
`TRIGger:EVENt<ev>:PATTern:QUALify:SOURces` on page 1055

**Set pattern**
Opens a dialog for setting the pattern by defining the channel stated and logic operator..

For analog channels, see Section 7.4.12.2, "Settings for state and pattern trigger of analog channels", on page 189.

For digital channels, see Section 7.4.12.3, "Settings for state and pattern trigger of digital channels", on page 191.

**Find level**
Sets the trigger level automatically to *0.5 * (MaxPeak – MinPeak)*.

In a trigger sequence, "Find level" affects all active events of the sequence (A, B, and R event).

The function is not available for trigger sources "Extern" and "Line".

Remote command:
TRIGger:FINDlevel on page 1032

### 7.4.12.2 Settings for state and pattern trigger of analog channels

Access for state trigger: "Menu" > "Trigger" > "Setup" tab > "Type = State" > "Set state"

Access for pattern trigger: "Menu" > "Trigger" > "Setup" tab > "Type = Pattern" > "Set pattern"

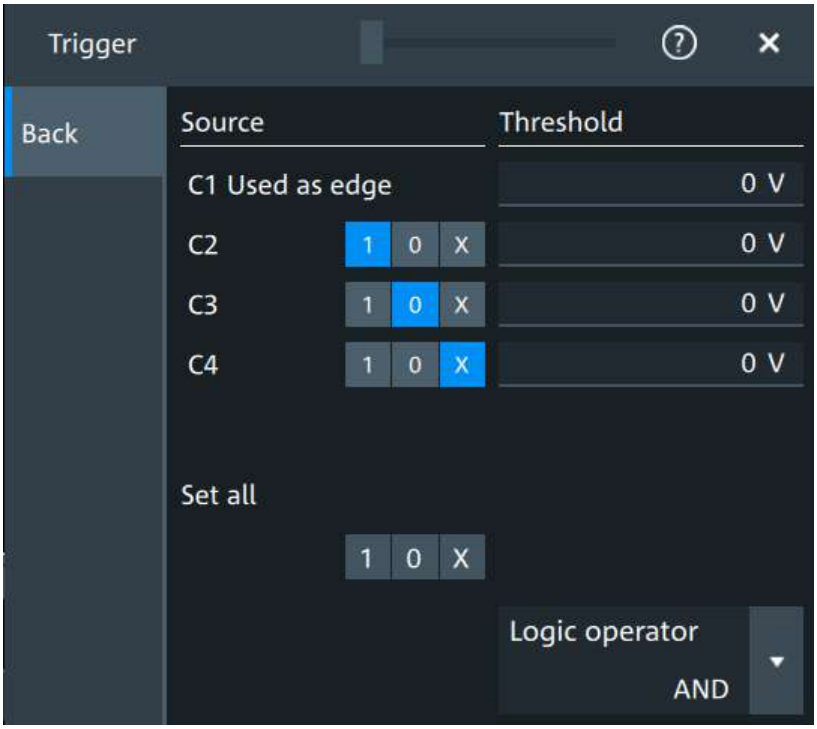



*Figure 7-3: Detailed settings for state trigger*

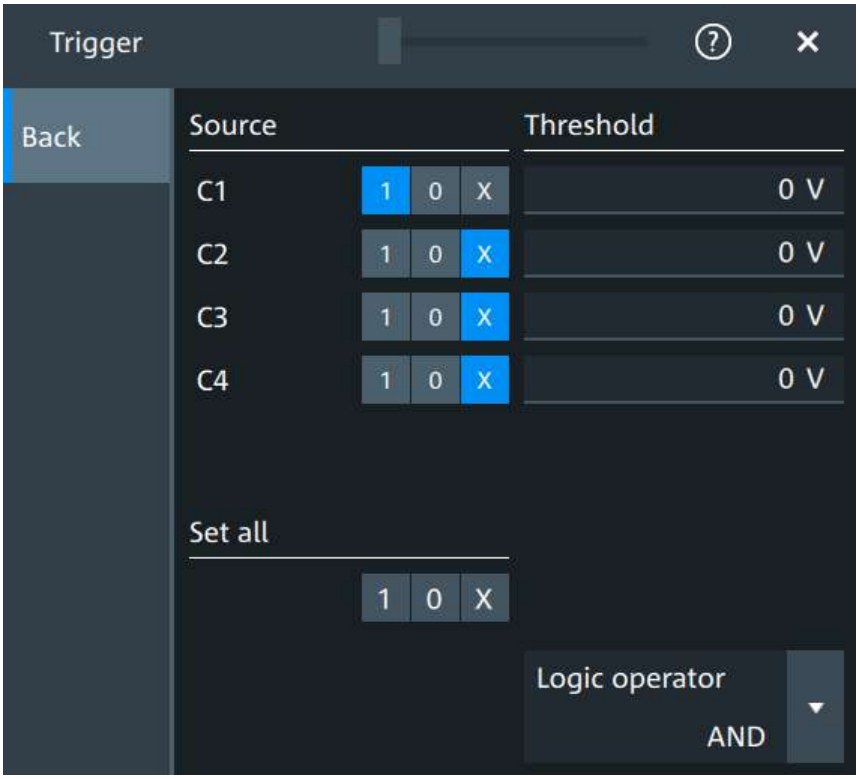



*Figure 7-4: Detailed settings for pattern trigger*

**Source: channel states**
Set the state for each channel. For the state trigger, the clock source is indicated and does not get a state.

Remote command:
`TRIGger:EVENt<ev>:STATe:QUALify:ANALog:CHAN<n>:HLX` on page 1052
`TRIGger:EVENt<ev>:PATTern:QUALify:ANALog:CHAN<n>:HLX` on page 1054

**Threshold**
Set the threshold for each channel.

Remote command:
`TRIGger:EVENt<ev>:LEVel<n>[:VALue]` on page 1032

**Set all**
Sets all channels to the selected state.

**Logic Operator**
Defines the logic combination of the channels and their states.

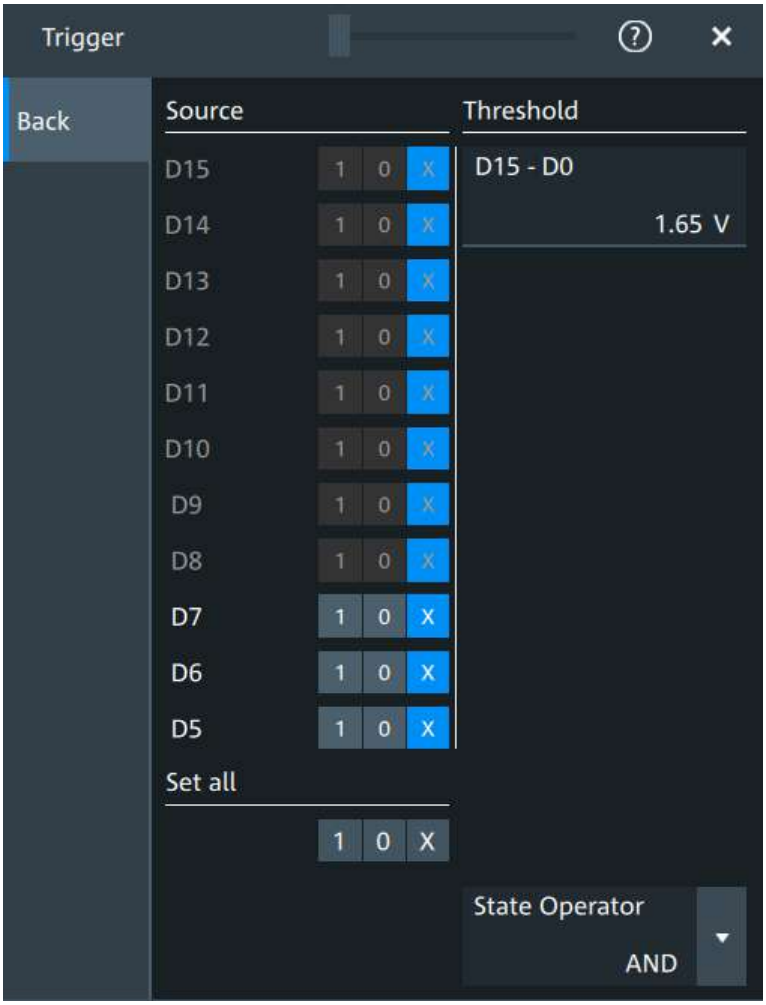
- "AND": logical AND, conjunctive combination
- "OR": logical OR, disjunctive combination

Remote command:
`TRIGger:EVENt<ev>:STATe:QUALify:LOGic` on page 1053
`TRIGger:EVENt<ev>:PATTern:QUALify:LOGic` on page 1055

TRIGger:EVENt<ev>:PATTern:QUALify:DIGital:LOGic on page 1055
TRIGger:EVENt<ev>:STATe:QUALify:DIGital:LOGic on page 1053

### 7.4.12.3 Settings for state and pattern trigger of digital channels

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Pattern" > "Set pattern"




**Source: channel states**
Set the state for each channel.

Remote command:
TRIGger:EVENt<ev>:PATTern:QUALify:DIGital:CHAN<n>:HLX on page 1054
TRIGger:EVENt<ev>:STATe:QUALify:DIGital:CHAN<n>:HLX on page 1054

**Set all**
Sets all channels to the selected state.

**Logic Operator**
Defines the logic combination of the channels and their states.

- "AND": logical AND, conjunctive combination
- "OR": logical OR, disjunctive combination

Remote command:
`TRIGger:EVENt<ev>:STATe:QUALify:LOGic` on page 1053
`TRIGger:EVENt<ev>:PATTern:QUALify:LOGic` on page 1055
`TRIGger:EVENt<ev>:PATTern:QUALify:DIGital:LOGic` on page 1055
`TRIGger:EVENt<ev>:STATe:QUALify:DIGital:LOGic` on page 1053

### 7.4.13 Line trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Source" = "Line"

The line trigger is an edge trigger that triggers on the AC power input and synchronizes the signal to the AC power frequency. It is not a trigger type but rather a special trigger source. Use the line source if you want to analyze signals related to the power line frequency, such as lighting equipment and power supply devices.

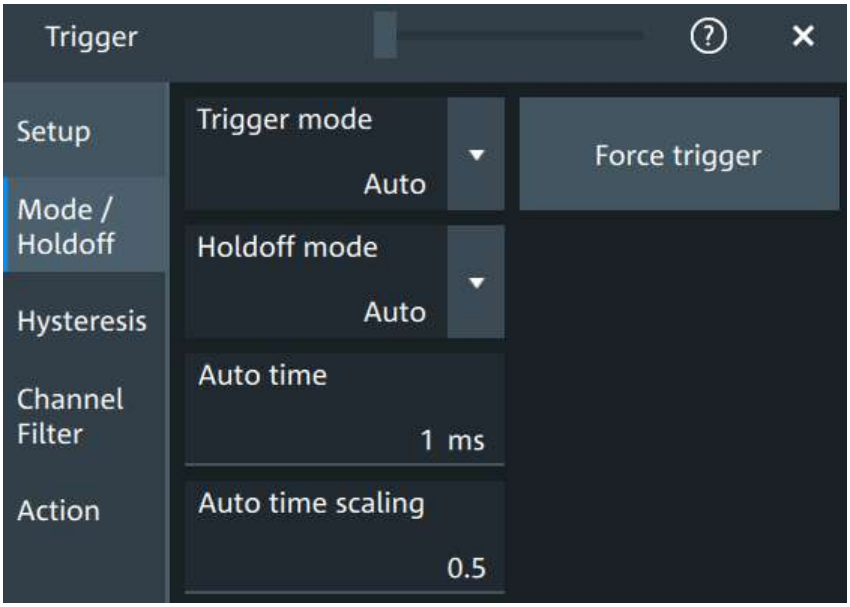
**Slope**
Selects the rising or falling edges of the AC power input.

Remote command:
`TRIGger:EVENt<ev>:EDGE:SLOPe` on page 1035

## 7.5 Trigger mode / holdoff

Access: "Trigger" menu > "Mode / holdoff" tab

Holdoff conditions define a waiting time after the current trigger until the next trigger can be recognized.



**Trigger mode**
Sets the trigger mode which determines the behavior of the instrument if no trigger occurs. The current setting is shown on the trigger label.

In a trigger sequence, the trigger mode affects only the A-trigger.

To toggle quickly between "Auto" and "Normal" mode, use the [Auto Norm] key on the front panel (in "Trigger" section).

"Auto" The instrument triggers repeatedly after a time interval if the trigger conditions are not fulfilled. If a real trigger occurs, it takes precedence. This mode helps to see the waveform even before the trigger conditions are set correctly. The waveform on the screen is not synchronized, and successive waveforms are not triggered at the same point of the waveform. The time interval depends on the timebase settings.

"Normal" The instrument acquires a waveform only if a trigger occurs, that is, if all trigger conditions are fulfilled. If no trigger occurs, no waveform is acquired and the last acquired waveform is displayed. If no waveform was captured before, none is displayed.
When no trigger has been found for longer than one second, a message box appears that shows the time elapsed since the last trigger.

"Free run" The instrument starts acquisition immediately and triggers after a short time interval independent of the timebase settings and faster than in "Auto" mode. Real triggers are ignored. Use this mode if the "Auto" mode is too slow.

Remote command:
`TRIGger:MODE` on page 1056

**Force trigger**
Provokes an immediate single acquisition. Force the trigger if the acquisition is running in normal mode and no valid trigger occurs. Thus, you can confirm that a signal is available and use the waveform display to determine how to trigger on it.

Remote command:
`TRIGger:FORCe` on page 1056

**Holdoff mode**
Selects the method to define the holdoff condition.

The trigger holdoff defines when the next trigger after the current will be recognized. Thus, it affects the next trigger to occur after the current one. Holdoff helps to obtain stable triggering when the oscilloscope is triggering on undesired events.

Holdoff settings are not available if the trigger source is an external trigger input or serial bus, and if you trigger on a sequence of events.

**Example:**

For example, you want to analyze the first pulse in a burst of several pulses. At first, you select a sufficiently slow time base to display the entire burst. Then, you set the holdoff time a little longer than the length of the burst. Now, each trigger corresponds to the first pulse in successive bursts, and you can change the time base to display the waveform in more detail.

The following methods are available:

"Time" Defines the holdoff directly as a time period. The next trigger occurs only after the "Holdoff time" has passed.

"Events" Defines the holdoff as a number of trigger events. The next trigger only occurs when this number of events is reached. The number of triggers to be skipped is defined in "Holdoff events".

"Random" Defines the holdoff as a random time limited by "Minimum time" and "Maximum time". For each acquisition cycle, the instrument selects a new random holdoff time from the specified range.
Random holdoff prevents synchronization to discover effects invisible with synchronized triggering, for example, the features of a pulse train.

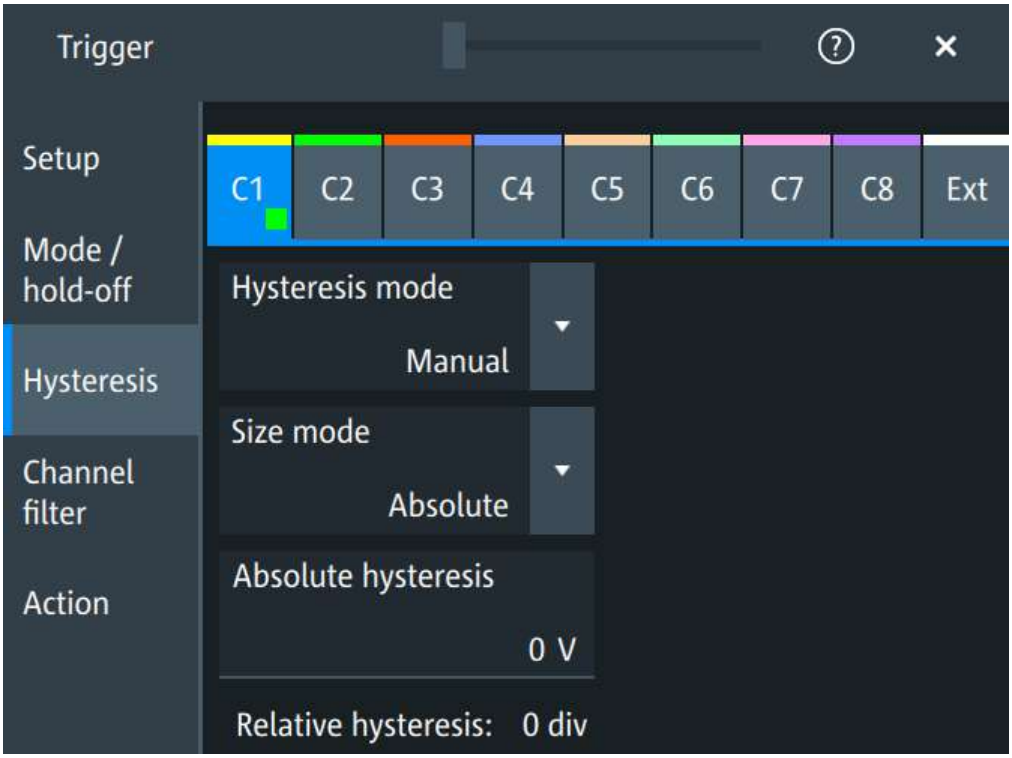
"Auto" The holdoff time is calculated automatically based on the current horizontal scale.
"Auto time scaling" defines the factor that the horizontal scale is multiplied with.
"Auto time" shows the resulting holdoff time: *Auto time = Auto time scaling * Horizontal scale*.

"Off" No holdoff

Remote command:
`TRIGger:HOLDoff:MODE` on page 1057
`TRIGger:HOLDoff:TIME` on page 1059
`TRIGger:HOLDoff:EVENts` on page 1058
`TRIGger:HOLDoff:MAX` on page 1058
`TRIGger:HOLDoff:MIN` on page 1059
`TRIGger:HOLDoff:AUTotime?` on page 1057
`TRIGger:HOLDoff:SCALing` on page 1058

## 7.6 Hysteresis

Access: "Menu">"Trigger" > "Hysteresis"




The rejection of noise by setting a hysteresis avoids unwanted trigger events caused by noise oscillation around the trigger level. You can select the hysteresis mode and value for each channel separately.

**HW hysteresis**
Displays the hysteresis that is set by the instrument in automatic hysteresis mode.

Remote command:
`TRIGger:NOISe<m>:EFFective?` on page 1060

**Hysteresis mode**
Selects how the hysteresis is set.

"Auto" Automatic mode is the recommended mode. The hysteresis is set by the instrument to reject the internal noise of the instrument.

"Manual" The hysteresis is defined directly in absolute or relative values.

Remote command:
`TRIGger:NOISe<m>:MODE` on page 1060

**Size mode**
Selects whether the hysteresis is defined in absolute or relative values. The setting is available only in manual hysteresis mode.

Remote command:
`TRIGger:NOISe<m>:MODE` on page 1060

**Absolute hysteresis**
Defines a range in absolute values around the trigger level. If the signal oscillates inside this range and thus crosses the trigger level, no trigger event occurs.

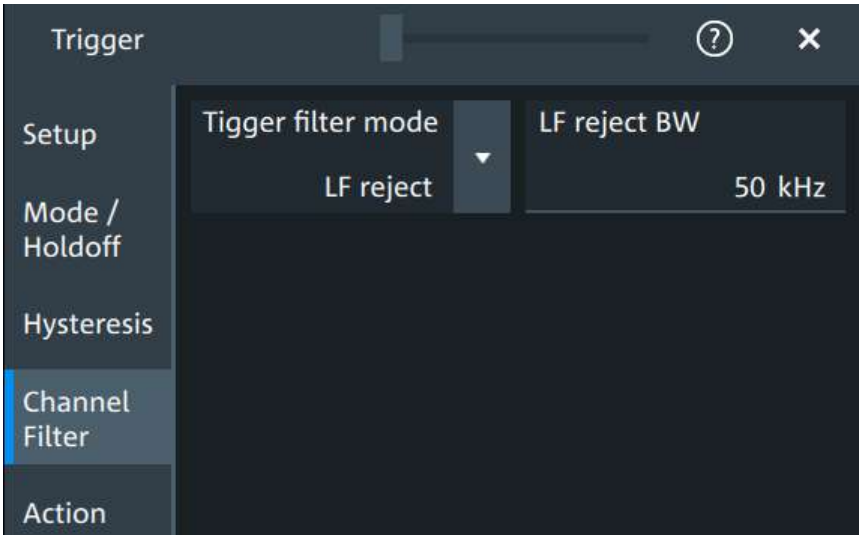
Remote command:
`TRIGger:NOISe<m>:ABSolute` on page 1059

**Relative hysteresis**
Defines a range in divisions around the trigger level in division or as percentage. If the signal oscillates inside this range and thus crosses the trigger level, no trigger event occurs.

Remote command:
`TRIGger:NOISe<m>:PERDivision` on page 1060
`TRIGger:NOISe<m>:RELative` on page 1061

## 7.7 Channel filter

Access: "Menu">"Trigger" > "Channel filter"




**Trigger filter mode**
Selects the filter mode for the trigger channel.

| | |
|---|---|
| "Off" | The trigger signal is not filtered. |
| "LF reject" | Frequencies lower than the "LF reject BW" are rejected, higher frequencies pass the filter. |
| "RF reject" | Frequencies higher than the "RF reject BW" are rejected, lower frequencies pass the filter. |

Remote command:
`TRIGger:FILTermode` on page 1061

**LF reject BW**
Sets the limit frequency limit for the highpass filter of the trigger signal. Frequencies lower than this value are rejected, higher frequencies pass the filter.

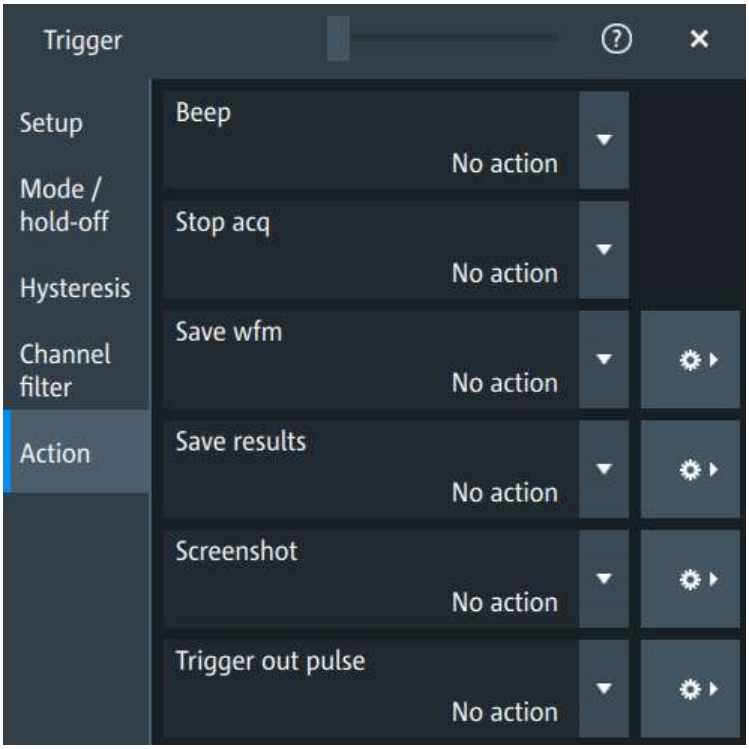
Remote command:
`TRIGger:LFReject` on page 1062

**RF reject BW**
Sets the limit frequency limit for the lowpass filter of the trigger signal. Frequencies higher than this value are rejected, lower frequencies pass the filter.

Remote command:
`TRIGger:RFReject` on page 1062

## 7.8 Actions on trigger

Access: "Menu" > "Trigger" > "Action" tab




The action settings define what happens when a trigger occurs. All available actions can be initiated at the same time. To activate an action, set it to "On trigger".

**Beep**
Generates a beep sound.

Remote command:
`TRIGger:ACTions:BEEP` on page 1063

**Stop acq**
Stops the acquisition when a trigger occurs.

Remote command:
`TRIGger:ACTions:STOP` on page 1065

**Save wfm**
Saves the waveform data to a file according to the folder, filename and extension settings in the "Menu" > "Save/recall" > "Save" > "Waveform" dialog.

To select the waveforms and adjust the file path, tap the settings icon.

See also Section 13.4.2, "Saving waveforms", on page 460.

Remote command:
`TRIGger:ACTions:WFMSave` on page 1065

**Save results**
Saves the results to a file according to the settings in "Menu" > "Save/recall" > "Save" > "Results".

To select the waveforms and adjust the file path, tap the settings icon.

See also Section 13.6, "Results", on page 475.

Remote command:
`TRIGger:ACTions:SRESults` on page 1064

**Screenshot**
Saves a screenshot according to the settings in the "Menu" > "Save/recall" > "Save" > "Screenshot" dialog.

To adjust the screenshot settings, tap the settings icon.

See also Section 13.8, "Screenshots", on page 481.

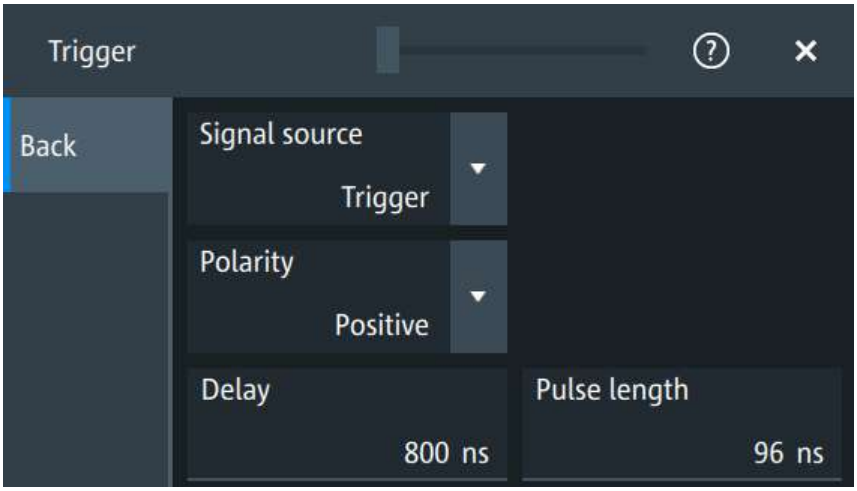
Remote command:
`TRIGger:ACTions:SCReenshot` on page 1064

**Trigger out pulse**
Selects, if a pulse is provided to the [Trigger Out] connector on the rear panel. The trigger-out signal is used to synchronize the measurements of other instruments.

To adjust the outgoing signal, tap the settings icon.

Remote command:
`TRIGger:ACTions:OUT:STATe` on page 1063

**Settings for the trigger out pulse**
In this dialog you can adjust the trigger out pulse.




**Signal source ← Settings for the trigger out pulse**
Defines when the trigger out signal is initiated: at the trigger point, when waiting for the trigger, or when the post-trigger time is finished.

Remote command:
`TRIGger:ACTions:OUT:SOURce` on page 1063

**Polarity ← Settings for the trigger out pulse**
Sets the polarity of the trigger out pulse, which is the direction of the first pulse edge.

Remote command:
`TRIGger:ACTions:OUT:POLarity` on page 1064

**Delay ← Settings for the trigger out pulse**
Defines the delay of the first pulse edge to the trigger point. The minimum delay is 600 ns.

Remote command:
`TRIGger:ACTions:OUT:DELay` on page 1063

**Pulse length ← Settings for the trigger out pulse**
Sets the length of the trigger out pulse.

Remote command:
`TRIGger:ACTions:OUT:PLENgth` on page 1064

## 7.9 Zone trigger

The zone trigger triggers on the intersection or non-intersection of the signal and one or more zones. The zone can be applied to any active input signal like the normal hardware trigger, and also to math waveforms, XY-plots and normal spectrum traces. So, the zone trigger works in the time domain and in the frequency domain.

You can use the zone trigger, for example, to solve the following tasks:

- Triggering on differential math signals.

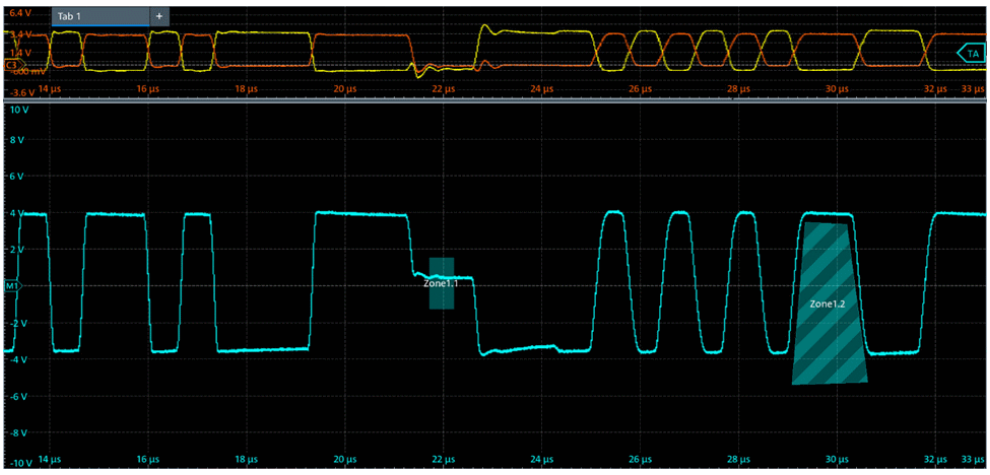



*Figure 7-5: Zone trigger on math*

- Trigger if a peak in the spectrum occurs: define a zone in the spectrum diagram to filter amplitude peaks.

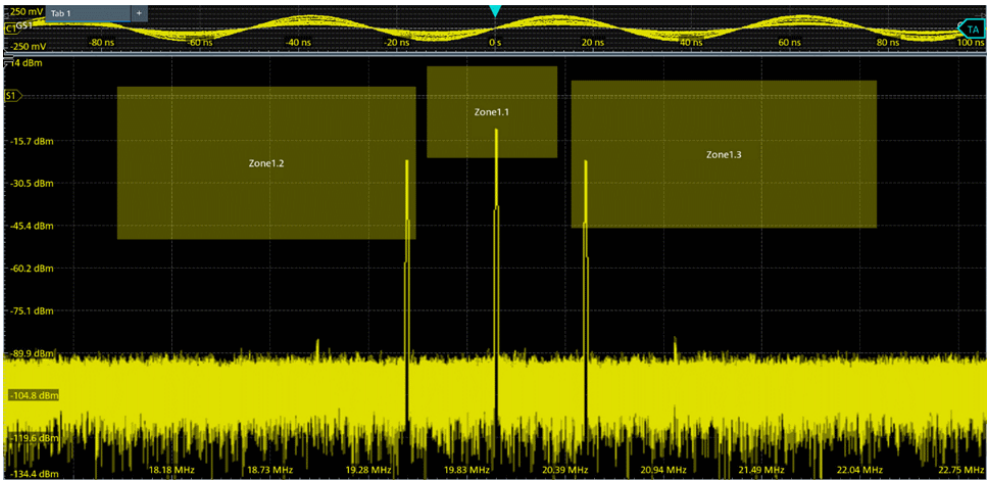



*Figure 7-6: Zone trigger on spectrum*

- Separate rising and falling edges: define a zone around the base or top of the digital signal.

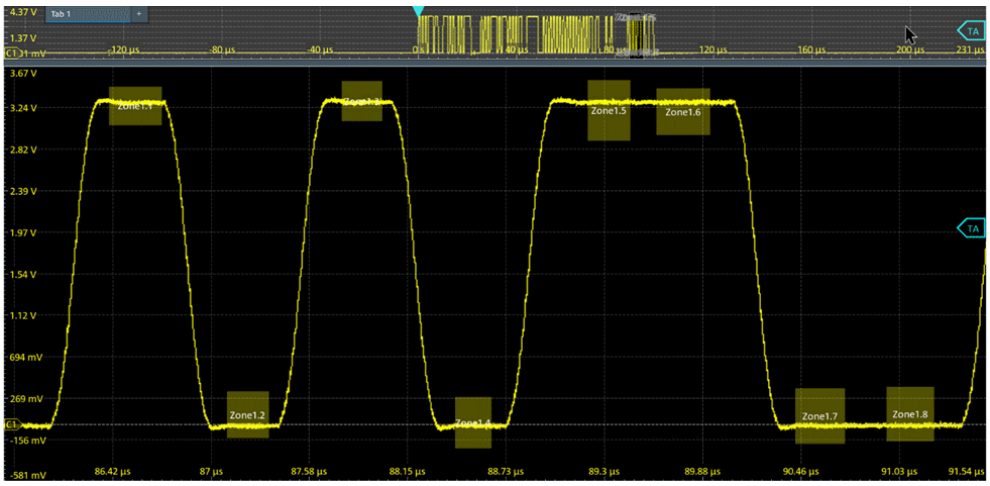



*Figure 7-7: Zone trigger on digital serial pattern*

- Zoom on a trigger event, even when capturing large amounts of time.

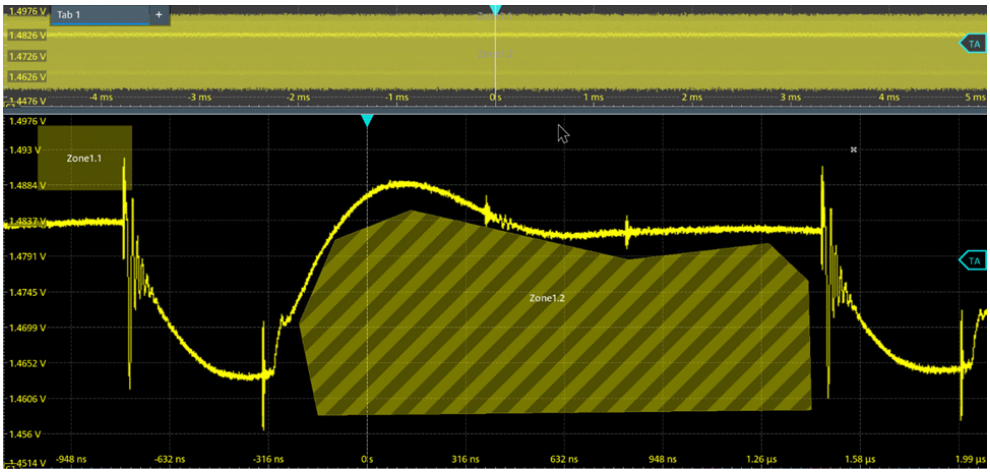



*Figure 7-8: Zoom on a zone trigger in the mV and ns range*

- Filter events in the history after the acquisition has finished.

The zone trigger is the second step of the "A → ZoneTrigger" trigger sequence and acts like a filter on previous acquisitions. The A-trigger condition and then the zone condition must be fulfilled independently.

Note that the trigger mode applies only to the A-trigger. If only the zone trigger conditions are relevant for the trigger, you can set the trigger mode to "Free run". Thus, you can use the zone trigger to trigger on a spectrum trace without triggering on a signal in the time domain, or you can analyze DC signals.

When an acquisition fulfills the complete trigger definition including the zone trigger, the acquisition is kept in the history memory. Otherwise, the acquisition is discarded. Moreover, you can apply the zone trigger to stored history waveforms that were acquired without zone trigger and thus filter the history waveforms on replay.

### Trigger zone definition

With the MXO 3, you can define up to 4 trigger zones. A zone consists of up to 8 areas associated with a single source. In total, 32 areas can be defined.

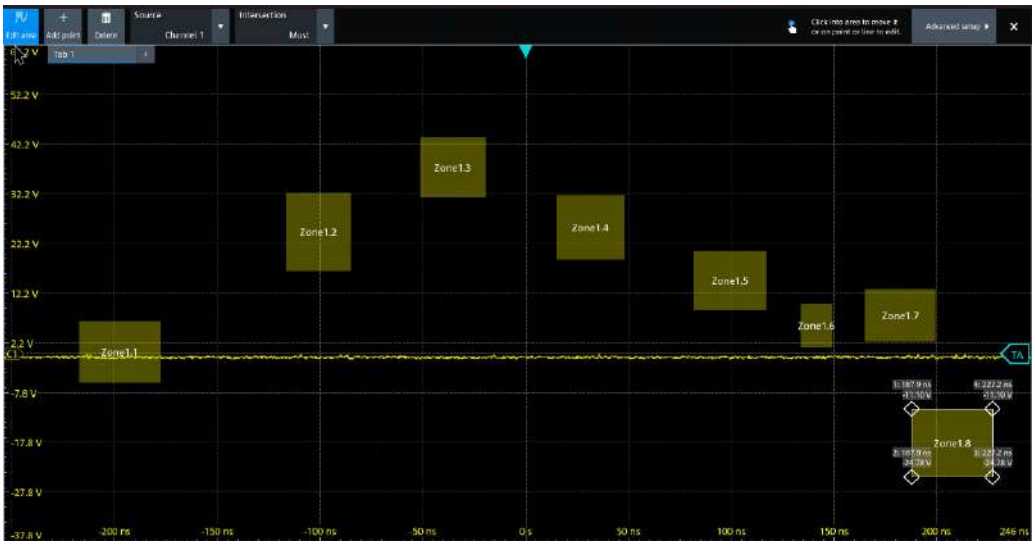



*Figure 7-9: 1 trigger zone with 8 areas*

The zone is applied to the diagram onto which it is drawn and has the color of the source. All areas of the zone have to be in the same diagram. If a zone is shown in a diagram for which it is not applied, for example in a zoom diagram, it is shown in a gray color.

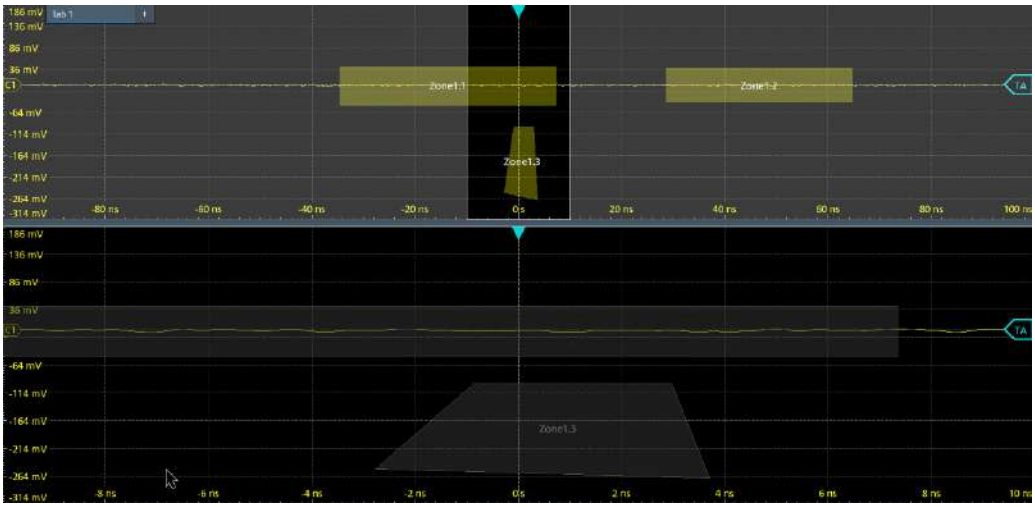



*Figure 7-10: Trigger zone and a zoom diagram*

Trigger zones can be applied to different sources, for example, the source of zone 1 is channel 1, and the source of zone 2 is a math signal. For the zone trigger condition, the zones are combined by Boolean operators: AND, OR, XOR, NOT. Thus, you can trigger on different waveform types simultaneously.

### Trigger area definition

An area is a graphical shape on the screen, defined by pixels. It is a part of the zone for a trigger. The area has the following characteristics:

- An area is a polygon that is applied against a captured waveform. All areas of the same zone have the same source.

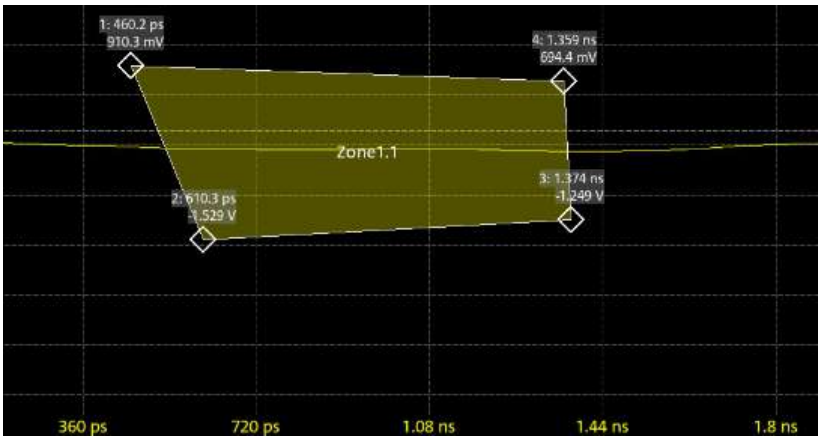



*Figure 7-11: 1 area of a trigger zone*

- An area can be defined from 3 up to 16 sides.

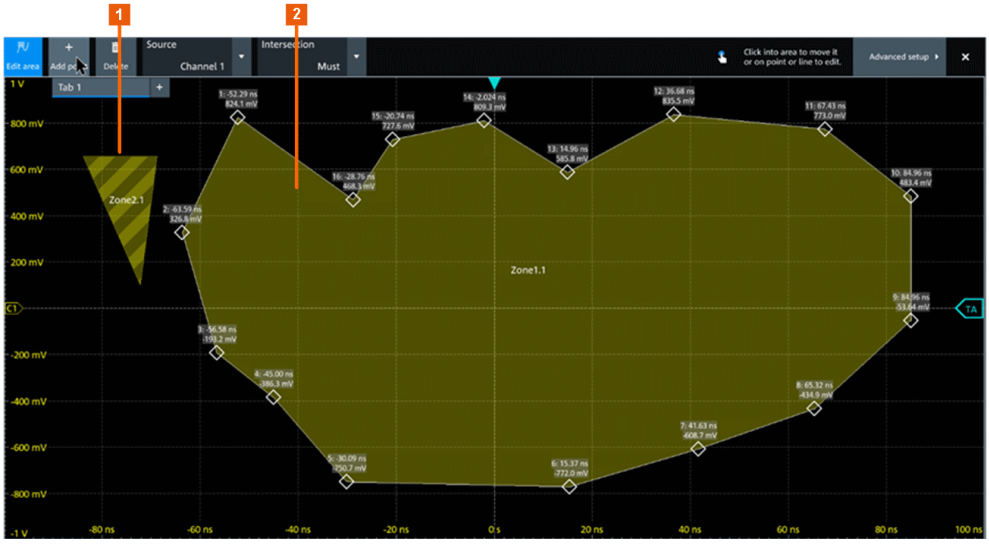



*Figure 7-12: Area sizes*

1 = 3 point area with "Intersection" = "Must not"
2 = 16 point area with "Intersection" = "Must"

- A zone area can have only one range in y-direction at a certain time point. If a shape is not valid, the invalid point is marked red, and an attention symbol is shown at the area name.

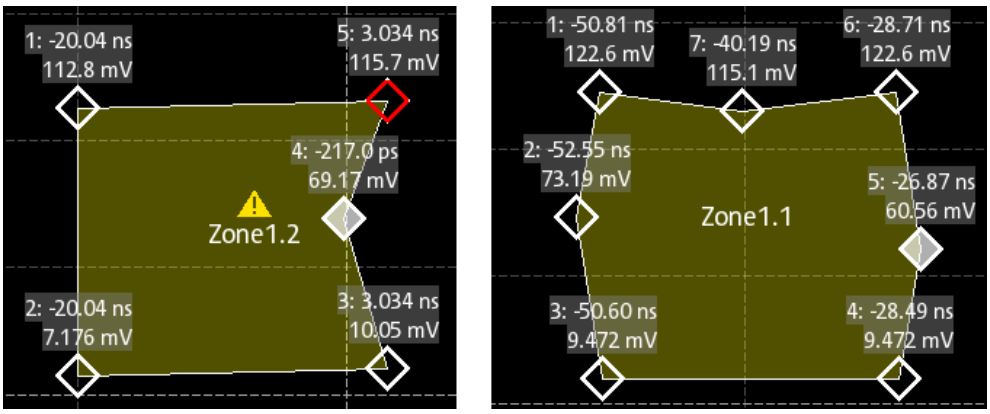



*Figure 7-13: Invalid zone area (left) and valid zone area (right)*

To check the validity in remote control, use `ZONE<m>:AREA<n>:VALid?` and `ZONE<m>:AREA<n>:POINt<o>:VALid?`.

- Next to each point, the number of the point and the horizontal and vertical values are shown.




*Figure 7-14: Area point definition*

1 = Number of the point
2 = Horizontal and vertical values of the point

- An area has a defined intersect condition, defining if the signal must intersect the area to allow the instrument to trigger, or if it must not intersect the area.
On the screen, an area that fulfills a "Must" condition has a solid fill in the color of the source channel. An area that fulfills a "Must not" also has the same color as the source channel, but is striped.

## 7.9.1 Using trigger zones and areas

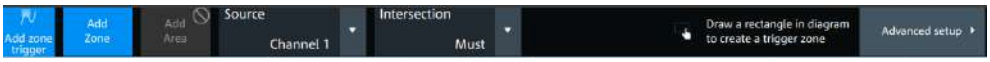
You can define the trigger zones on the display through the toolbar:

**To add trigger zones and areas**

1. Tap the "Add zone trigger" icon on the toolbar.



The trigger zone toolbar opens.
Alternatively, you can use the Zone key on the front panel.



2. Select the "Source" of the zone, the signal to which the zone condition is applied.
3. Select if the signal must violate the zone to cause a trigger ("Intersect" = "Must"), or vice versa ("Intersect" = "Must not").
4. Tap "Add zone".
5. Draw a rectangle in the diagram to create the first area of a new zone trigger.
6. To add more areas to the zone, tap "Add Area".

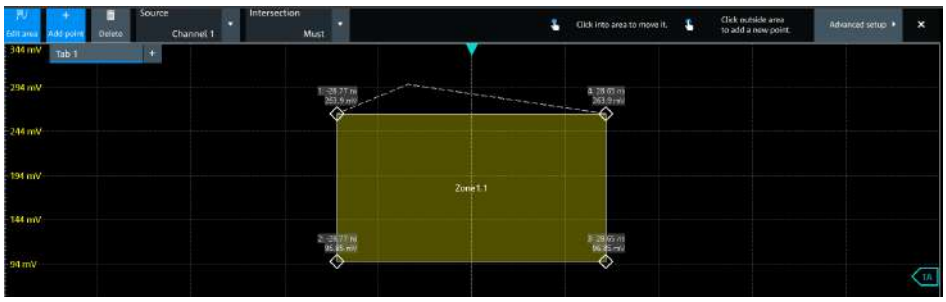



7. Draw a rectangle in the same diagram to add a new trigger area.
8. Select if the signal must violate the area to cause a trigger (must intersect), or vice versa (must not intersect).
9. You can add more areas and further more zones to the zone trigger.

   Make sure to draw them in the correct diagram because zones affect only the diagram where they were defined.

**To edit an existing area**

1. Tap an existing zone area.
2. To change the shape of the zone area, tap the corner points of the zone on the SmartGrid.
3. Move the point to the required coordinates on the SmartGrid.
4. If necessary, add more points to the zone area:
   a) In the toolbar, tap "Add point".




   b) Tap outside of the area to add the new point.
   c) If necessary, add more points.
   d) If necessary to change the coordinates of a point, tap on it and move it.

**To delete an area or point**

1. Tap an existing zone area.
2. In the "Edit area" toolbar, tap the delete icon.
3. Tap on the area that you want to delete or edit.



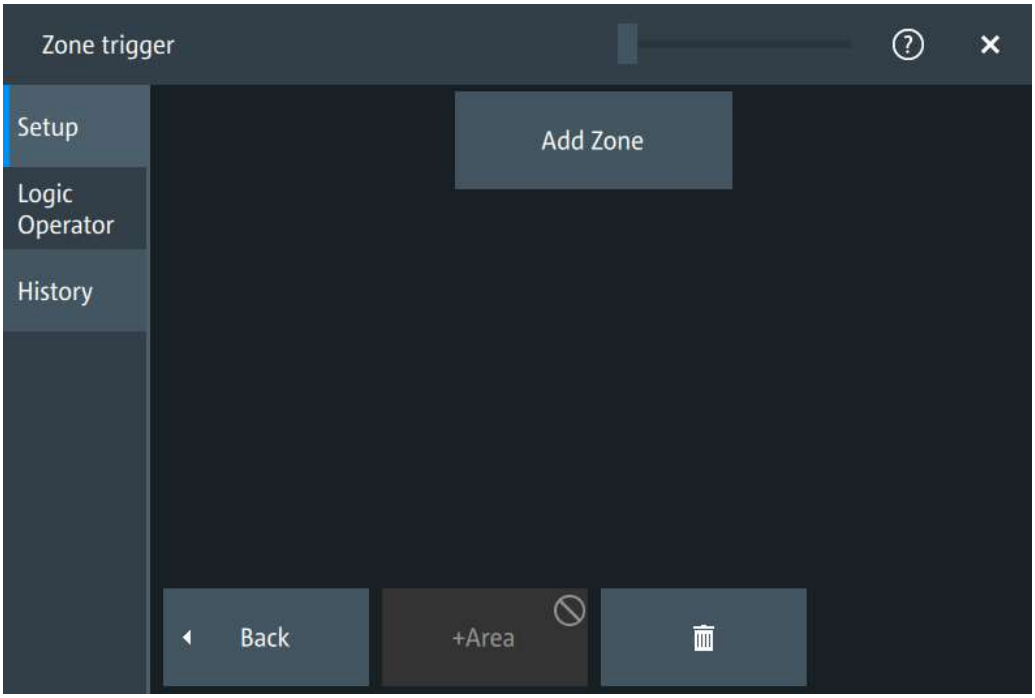

Delete icons appear at the area points and in the middle of the area.

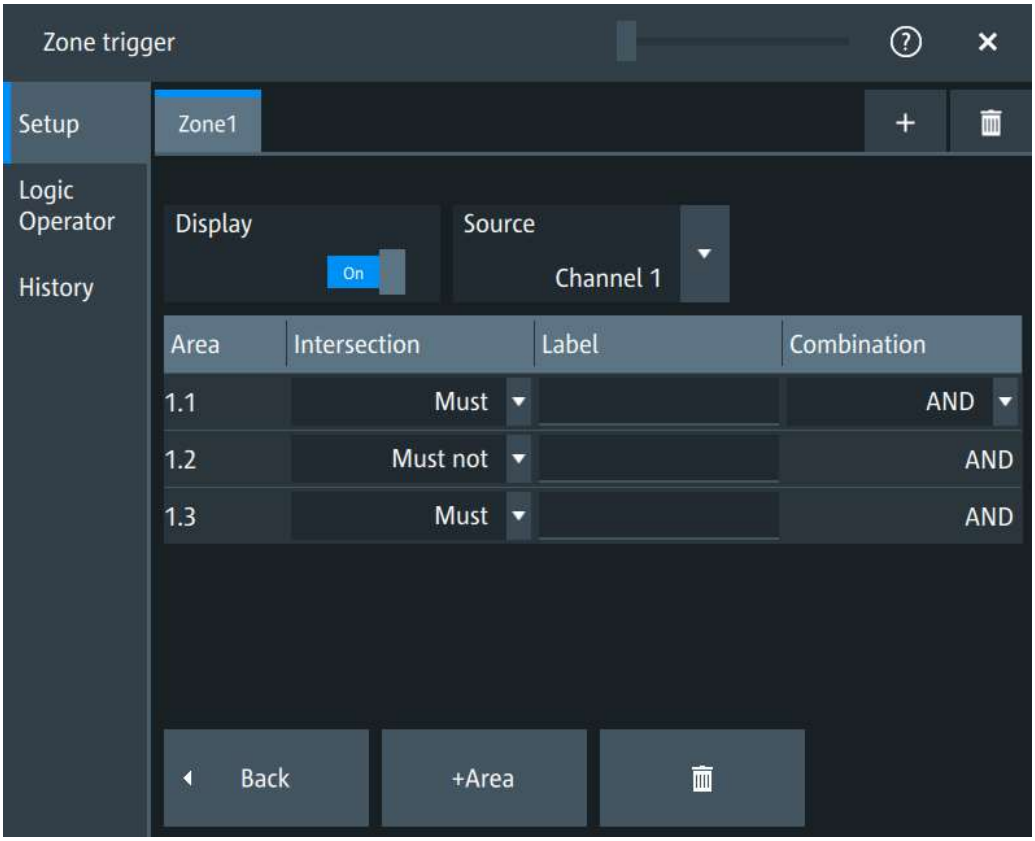
4. To delete a point from the area, tap on the "Delete" icon on top of the point.
5. To delete the whole area, tap the "Delete" icon in the middle of the area next to the area "Label".

## 7.9.2 Zone trigger settings

Access: "Menu" > "Trigger" > "Setup" tab > select sequence with zone trigger > "Setup Z trigger" > "Setup" tab, or: Toolbar > "Add zone trigger" icon > "Advanced setup".

In this dialog, you can add zones and areas to your zone trigger and adjust their shapes.






**Add zone,**
Adds a new trigger zone. You can add up to four zones.

Remote command:
`ZONE<m>:ADD` on page 1066

**Display**
Enables the display of the zone on the screen.

Remote command:
`ZONE<m>[:VISible]` on page 1072

**Source**
Sets the source of the zone trigger. The zone can be applied to any active input signal, math waveform, XY-plot, normal spectrum trace and power waveforms.

Reference waveforms are not supported as a spurce for the zone trigger. Thus, also math waveforms are not supported if they are calculated on reference waveforms.

Remote command:
`ZONE<m>:SOURce` on page 1072

**Area**
Defines the area and its condition for the zone.

"+Area" Adds a new area to the zone. You can add up to 8 areas.

"Intersect" Defines if the signal must intersect the area to allow the instrument to trigger, or if it must not intersect the area.

"Label" Defines a label for the selected area.

"Combination" Sets the logic combination that applies to all areas in a zone. Available are "AND" or "OR".
Example: If three areas are defined and the combination is "OR", then the Boolean state of the zone is Zone1 = Area1 OR Area2 OR Area3..

Remote command:
`ZONE<m>:AREA<n>:ADD` on page 1067
`ZONE<m>:AREA<n>:COUNt?` on page 1067
`ZONE<m>:AREA<n>:INTersect` on page 1067
`ZONE<m>:AREA<n>:LABel` on page 1068
`ZONE<m>:ACOMbination` on page 1066

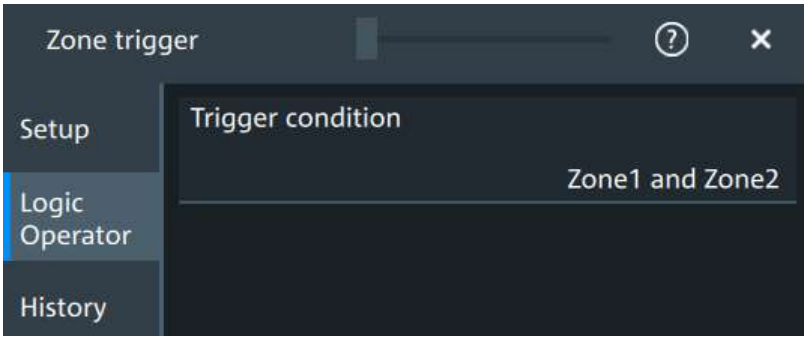
**Delete icon**
Enables the removal of areas of the zone.

You can also delete all current areas with "Delete All".

Remote command:
`ZONE<m>:REMove` on page 1072
`ZONE<m>:AREA<n>:REMove` on page 1070

### 7.9.3 Logic operator settings

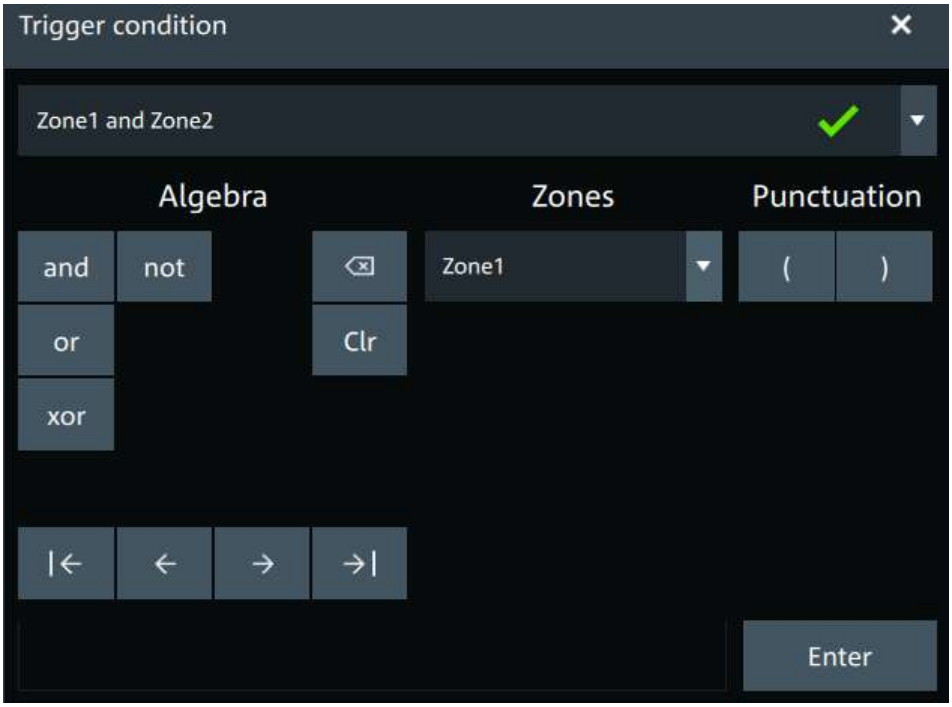
Access: "Menu" > "Trigger" > "Setup" tab > select sequence with zone trigger > "Setup Z trigger" > "Logic operator" tab.




**Trigger condition**
Sets a logical expression for the trigger condition between different defined zones.

If all required zones are defined, you can type the logical expression directly, or use the trigger condition editor.

Double-tap the "Trigger condition" field to open the editor:




The keys of the "Trigger condition" editor are described in Trigger condition keys.

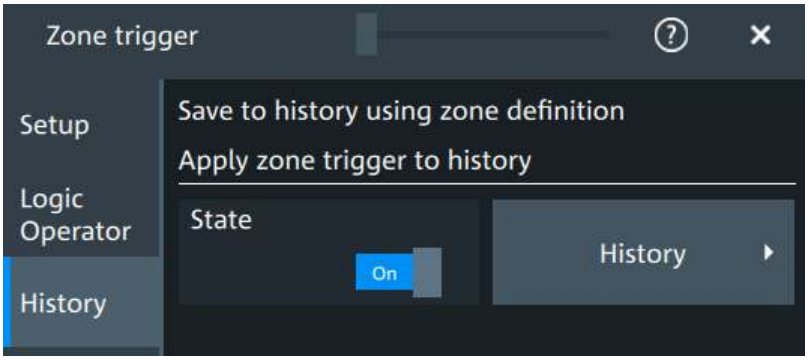
***Table 7-1: Trigger condition keys***

| Icon | Description |
|---|---|
| and | And |
| not | Must not intersect |
| or | Or |
| xor | Exclusive or |
| Zone | Selects the zone |
| Clr | Clear expression in editor |
| ( | Left bracket to enclose operands |
| ) | Right bracket to enclose operands |
| \|← | Moves cursor to beginning |
| ← | Moves cursor 1 step to the left |
| → | Moves cursor 1 step to the right |
| →\| | Moves cursor to end |

Remote command:
`TRIGger:ZONE:EXPRession[:DEFine]` on page 1066

### 7.9.4 History settings

Access: "Menu" > "Trigger" > "Setup" tab > select sequence with zone trigger > "Setup Z trigger" > "History" tab.




When a zone trigger is defined, the history saves only acquisitions that fulfill the complete trigger condition including the zone trigger condition. On the other hand, you can use the current zone trigger definition to filter the history replay.

**Apply zone trigger to history**
If enabled, the instrument applies the zone trigger condition to the acquisitions that are stored in the history memory. Tap "History" to open the history player and analyze the filtered waveforms.

Using "Apply zone trigger to history", you can:
- Acquire waveforms with high speed and filter them afterwards in the history.
- Change the zone trigger condition after acquisition.

Remote command:
`TRIGger:ZONE:HISTory` on page 1071

# 8 Signal configuration

## 8.1 Reference level setup

Some measurements and analysis tasks require reference levels to obtain the measurement points, e.g. time measurements or timing reference. You can define 8 sets of reference levels. During configuration of the analysis task, you select which set of reference levels is used.

Access: "Menu" > "Apps" > "General" tab > "Signal config" > "Reference level" tab

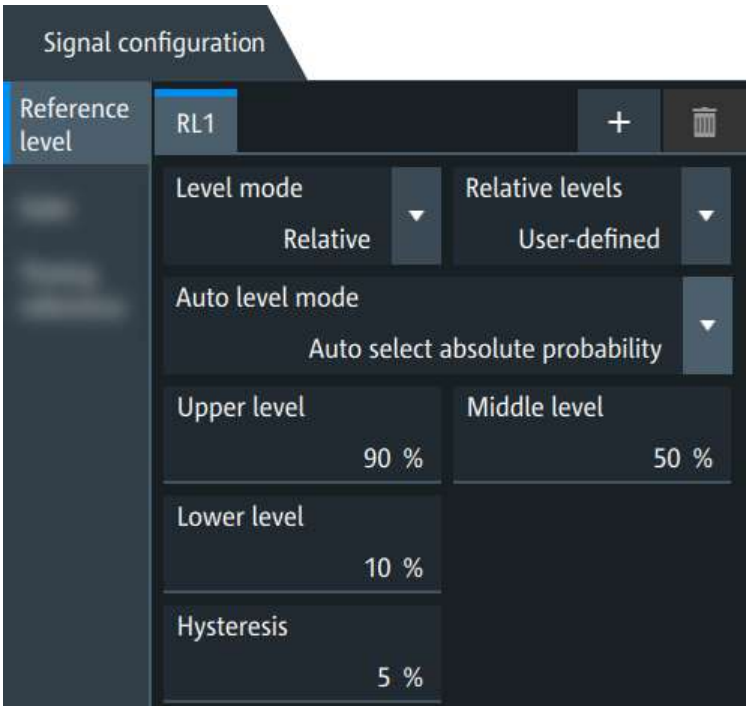



***Figure 8-1: Relative reference level settings***

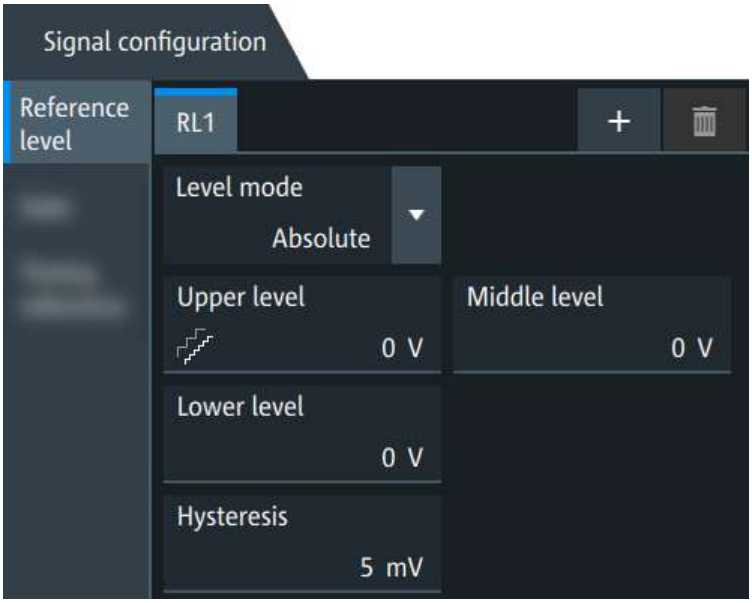



*Figure 8-2: Absolute reference level settings*

**RLx**
Select the correct reference level tab before you enter the settings.

You can add up to 8 reference levels.

Remote command:
`REFLevel<rl>:ENABle` on page 1073

**Level mode**
Defines if the reference level is set in absolute or relative values.

Remote command:
`REFLevel<rl>:LMODe` on page 1073

**Absolute Level mode**
In the absolute mode, the reference values are set in absolute values with the unit of the vertical axis.

**Upper level, Middle level, Lower level ← Absolute Level mode**
Set the lower, middle and upper reference signal levels.

Remote command:
`REFLevel<rl>:ABSolute:LLEVel` on page 1074
`REFLevel<rl>:ABSolute:MLEVel` on page 1075
`REFLevel<rl>:ABSolute:ULEVel` on page 1075

**Relative Level mode**
In the relative mode, the reference values are set as percentages of the signal amplitude.

**Auto level mode ← Relative Level mode**
Defines the computation method for high and low signal levels. The instrument analyzes and measures the signal, and defines the signal levels using the selected method. The reference levels are then defined on the computed signal levels.

See also: Section 10.2.5, "Histogram measurements", on page 289.

| | |
|---|---|
| "Auto select absolute probability" | The most suitable signal levels are used. |
| "Peak probability" | The signal levels with the highest probability values are used. These values are the upper peak value and the lower peak value of the histogram measurement. |
| "Mean probability" | The signal levels with mean probabilities are used. |
| "Absolute peak" | The absolute peak signal levels are used. These values are the maximum and minimum signal values of the amplitude measurement. |

Remote command:
`REFLevel<rl>:AUTO:MODE` on page 1073

**Relative levels ← Relative Level mode**
Selects the lower, middle and upper reference levels, defined as percentages of the signal amplitude.

Available relative levels:

- 5/50/95
- 10/50/90
- 20/50/80
- User defined: Enter "Upper level", "Middle level", and "Lower level".

For example, for "5/50/95" the levels are set to the following values:

- Lower level = 5% of the signal amplitude
- Middle level = 50% of the signal amplitude
- Upper level = 95% of the signal amplitude

Remote command:
`REFLevel<rl>:RELative:MODE` on page 1076

**Upper level, Middle level,Lower level ← Relative Level mode**
Define the reference levels in percent, if "Level mode" is set to "User-defined".

Remote command:
`REFLevel<rl>:RELative:LOWer` on page 1076
`REFLevel<rl>:RELative:MIDDle` on page 1076
`REFLevel<rl>:RELative:UPPer` on page 1077

**Hysteresis**
Defines a hysteresis for the middle reference level. A rise or fall from the middle reference value that does not exceed the hysteresis is rejected as noise.

Remote command:
`REFLevel<rl>:ABSolute:HYSTeresis` on page 1074
`REFLevel<rl>:RELative:HYSTeresis` on page 1075

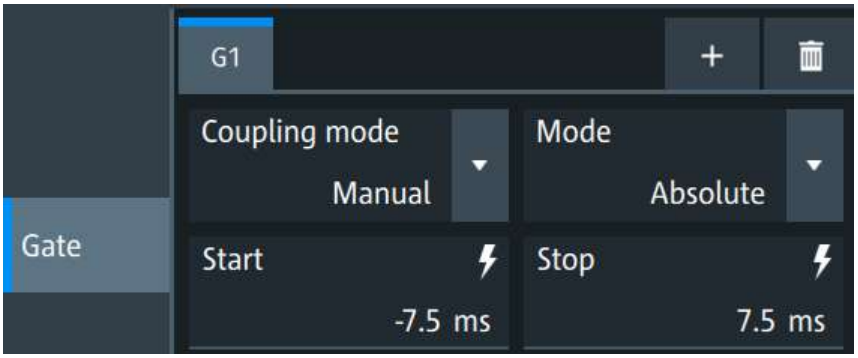
## 8.2 Gate setup

Gate areas limit various measurements to a user-defined range of the waveform.

You can define up to 8 separate gates. You can define absolute or relative limit values, or use an existing zoom, cursor set or spectrum as a gate.

Then, you assign the gates to analysis tasks, for example, to measurements or power analysis.

Access: "Menu" > "Apps" > "General" tab > "Signal config" > "Gate" tab




**Add**
Creates a new gate.

Remote command:
`GATE<g>:ENABle` on page 1078

**Coupling mode**
The gate coupling mode selects how the gate area is defined.

| | |
|---|---|
| "Manual" | Manually define the gate with a user-defined start and stop values. |
| "Cursor" | Cursor coupling is available if a cursor is defined. The gate area is defined by the cursor lines of an active cursor measurement.<br>If several cursor measurements are enabled, select the "Cursor" to be used for gating. The start and stop values of the gate are adjusted to the values of the cursor line positions. The measurement is limited to the part of the waveform between the cursor lines. |
| "Zoom" | Zoom coupling is available if a zoom is defined. The gate area is defined identically to the zoom area - if you change the zoom, the gate changes as well.<br>If several zoom diagrams are defined, select the "Zoom" diagram to be used for gating. The start and stop values of the gate are adjusted to the values of the zoom positions. |
| "Spectrum" | Spectrum coupling is available if a spectrum is enabled.<br>If several spectrums are enabled, select the "Spectrum" diagram to be used for gating. |

Remote command:
GATE<g>:GCOupling on page 1078
GATE<g>:CURSor on page 1079
GATE<g>:ZDIagram on page 1079

**Mode, Start, Stop**
Selects if the gate settings are configured using absolute or relative values.

"Absolute" The gate is defined by absolute start and stop values with "Start" and "Stop".

"Relative" The gate's start and stop values are defined by a percentage of the value range with "Relative start" and "Relative stop".

Lightning symbols in the value fields indicate that the gate is not yet assigned to any measurement.

Remote command:
GATE<g>:MODE on page 1080
GATE<g>:ABSolute:STARt on page 1080
GATE<g>:ABSolute:STOP on page 1080
GATE<g>:RELative:STARt on page 1080
GATE<g>:RELative:STOP on page 1080

## 8.3 Timing reference

The timing reference is the clock that is used for jitter measurements, for example. You can configure up to 4 timing references. The clock can be a captured clock signal, or it can be recovered from the data signal by software CDR.

### 8.3.1 General settings of the timing reference

Access: "Menu" > "Apps" > "General" tab > "Signal config" > "Timing reference".

**Type**
Defines the origin of the clock signal - either an existing clock signal or waveform, or a clock generated by clock data recovery (CDR).

"Explicit clock" The clock is an existing clock signal.

"Software CDR" The clock is generated by a software algorithm.

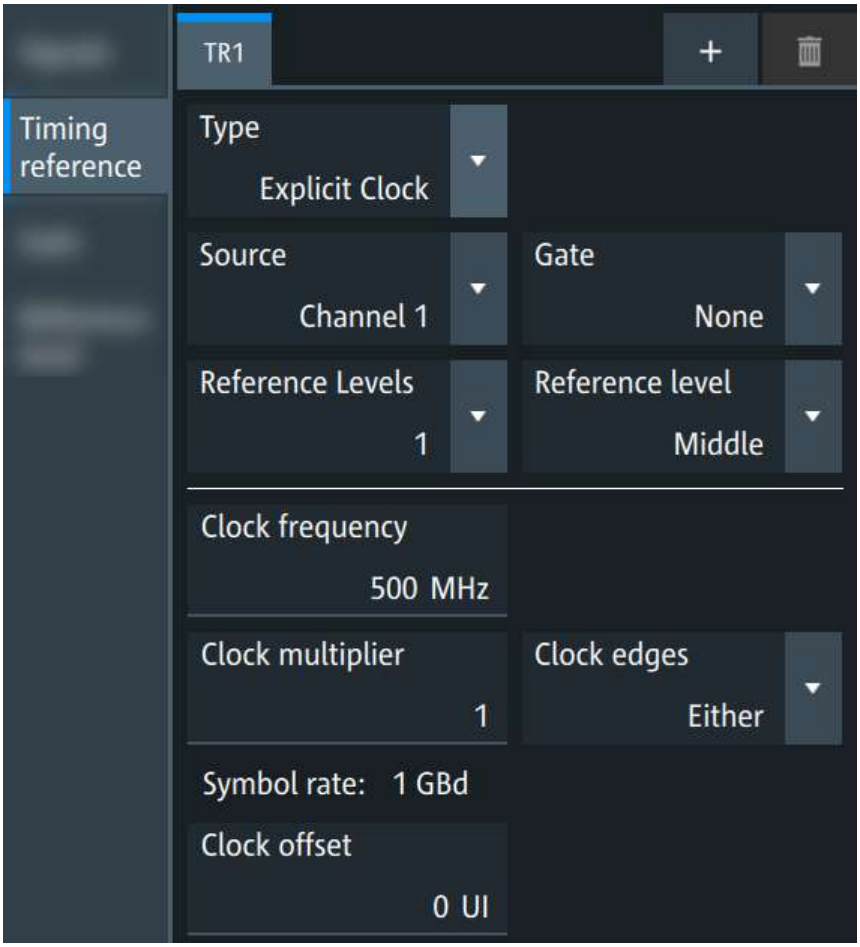
Remote command:
TREFerence<m>:TYPE on page 1088

**Reference levels**
Selects the set of reference levels that is used for the timing reference measurements.

Remote command:
TREFerence<m>:RFLSet on page 1087

### 8.3.2 Explicit clock signal

Access: "Menu" > "Apps" > "General" tab > "Signal config" > "Timing reference" > "Type" = "Explicit clock".




**Reference level**
Selects the reference level that is used for the timing reference measurement if the clock is an existing clock signal.

Remote command:
`TREFerence<m>:CLK:REFLevel` on page 1085

**Source**
Selects the clock source if the clock is an existing clock signal.

Remote command:
`TREFerence<m>:SOURce` on page 1087

**Clock frequency**
Sets the frequency of the clock signal. The setting is relevant if the clock is an explicit clock signal.

Remote command:
TREFerence<m>:CLK:FREQuency on page 1085

**Clock multiplier**
Sets a value for the clock multiplier, which is the ratio of an internal clock rate to the externally supplied clock. It defines the number of samples per clock interval. The setting is relevant if the clock is an explicit clock signal.

Remote command:
TREFerence<m>:CLK:FACTor on page 1084

**Clock edges**
Sets the clock edges that are used for measurements. The setting is relevant if the clock is an explicit clock signal.

"Positive" The positive clock slope can be used, for example, for single data rate (SDR) signals with bit start at the positive clock edge.

"Negative" The negative clock slope can be used, for example, for SDR signals with bit start at the negative clock edge.

"Either" Both edges can be used for double data rate (DDR) signals.

Remote command:
TREFerence<m>:EDGE on page 1086

**Symbol rate**
Shows the symbol rate of the data signal. The setting is relevant if the clock is an explicit clock signal.

*SymbolRate* = *ClockFrequency* * *Multiplier* * *EdgeCount*, where EdgeCount = 2 for either slope and EdgeCount = 1 for positive and negative slopes.

Remote command:
TREFerence<m>:SYMRate on page 1088

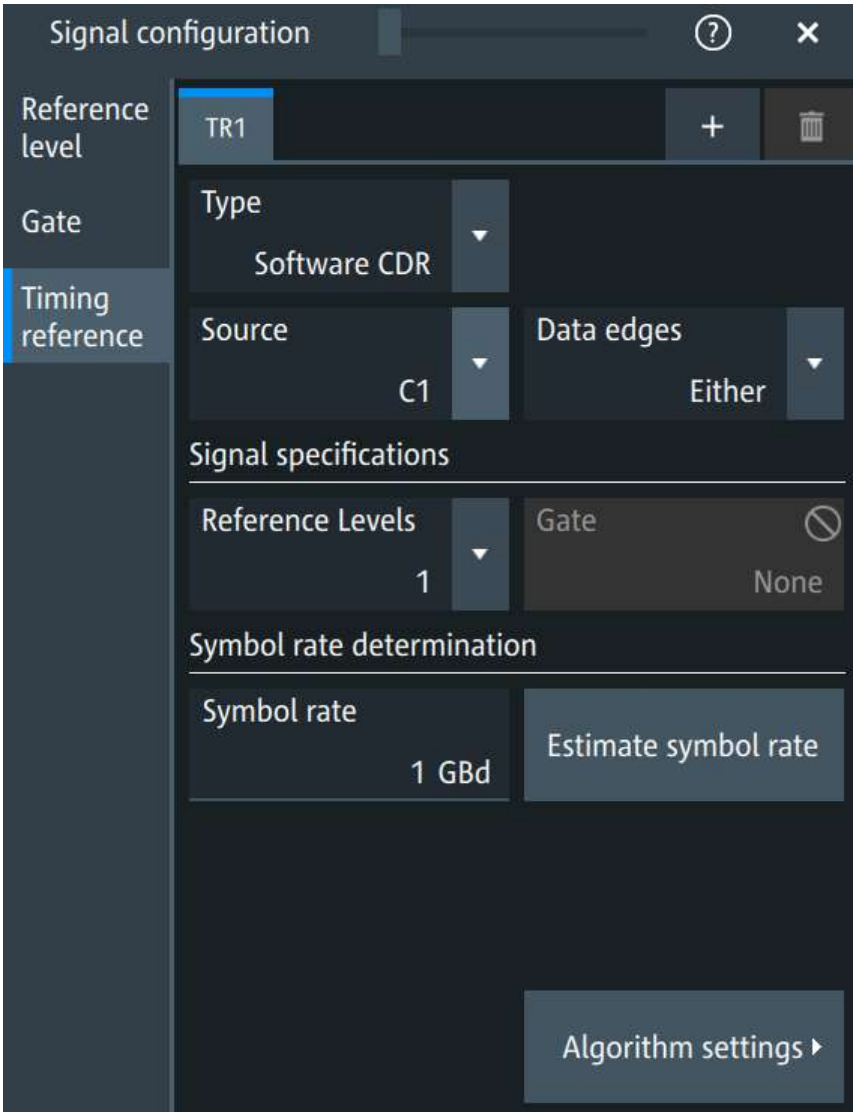
**Clock offset**
Sets the offset between the clock edge and the data edge in unit intervals. The setting is relevant if the clock is an explicit clock signal.

Remote command:
TREFerence<m>:CLK:OFFSet on page 1085

### 8.3.3 Clock data recovery by software CDR

Access: "Menu" > "Apps" > "General" tab > "Signal config" > "Timing reference" > "Type" = "Software CDR".

If "Type" is set to "Software CDR", you need to set up the algorithm for the clock data recovery.




**Source**
Selects the signal from which the clock is recovered.

Remote command:
`TREFerence<m>:SOURce` on page 1087

**Data edges**
Selects the edges of the data stream that are used for the clock recovery.

- "Either": Both positive and negative edges are used.
- "Positive" or "Negative": Only one edge direction is used. Use one of these settings if the other edge delivers unreliable results.

Remote command:
`TREFerence<m>:EDGE` on page 1086

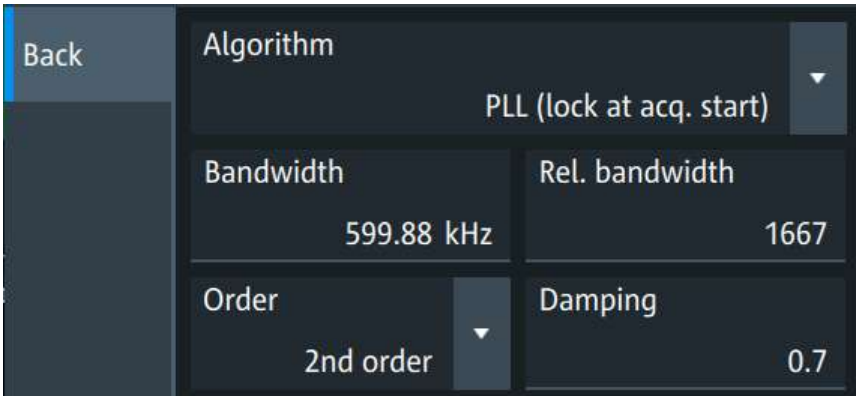
**Symbol rate**
Sets the quiescent frequency of the PLL. It corresponds to the data rate of the data stream from which the clock is to be recovered.

Remote command:
`TREFerence<m>:SYMRate` on page 1088

**Estimate symbol rate**
Starts a software algorithm to detect the symbol rate for the selected source automatically.

Remote command:
`TREFerence<m>:CDR:SOFTware:SREStimate` on page 1083

**Algorithm**
Sets the software algorithm that is used for software clock data recovery.




| | |
|---|---|
| "Constant freq." | CDR uses the nominal symbol rate to generate the clock signal. The method assumes that the frequency of the signal is constant during the complete acquisition. |
| "PLL" | PLL is the phase-locked loop control system. It can follow slow deviations in the frequency of the data stream. Thus, it acts as a highpass filter regarding the jitter that remains on the signal. |
| "Feed Forward" | The feed-forward algorithm is a mathematical optimization method used for clock data recovery. It is faster than PLL, i.e., the settling time is shorter. Feed forward is also more precise than PLL. |
| "PLL (lock at acq. start)" | Phase-locked loop control system, which is locked at the acquisition start. |

Remote command:
`TREFerence<m>:CDR:SOFTware:ALGorithm` on page 1081

**Clock resync**
Enables continuous synchronization of the clock with the data signal. At constant clock frequency without resync, the clock starts at the first detected edge and clocks with the given or estimated bit rate or symbol rate. With synchronization, the clock searches for the next edge and synchronizes to this edge. Thus, the sampling points are not exactly at constant frequency but aligned to the defined edges of the data signal.

Remote command:
TREFerence<m>:CDR:SOFTware:CFRequency:CRSYnc on page 1082

**Bandwidth**
Sets the PLL bandwidth. It defines the part of the spectrum that the PLL can follow during synchronization. The PLL bandwidth is usually defined by the transmission standard.

Remote command:
TREFerence<m>:CDR:SOFTware:BWIDth on page 1082

**Rel. bandwidth**
Sets the relative bandwidth, which is the ratio of the nominal bit rate to the PLL bandwidth.

Remote command:
TREFerence<m>:CDR:SOFTware:RELBwidth on page 1084

**Order**
Sets the order of the PLL: first or second order. PLL of higher order can compensate for more complex jitter behavior.

Remote command:
TREFerence<m>:CDR:SOFTware:PLL:ORDer on page 1083

**Damping**
Sets the damping factor, which is only relevant for second order PLL.

Remote command:
TREFerence<m>:CDR:SOFTware:PLL:DAMPing on page 1082

**Initial phase sync**
Available for "Algorithm" = "PLL".

Defines the phase reference for the first clock edge.

| | |
|---|---|
| "First data edge" | The first clock edge matches the first edge of the data signal. |
| "First sample" | The first clock edge matches the first sample of the waveform at the left border of the display. |

Remote command:
TREFerence<m>:CDR:SOFTware:PLL:SYNC on page 1083

**Selected results**
Available for "Algorithm" = "PLL".

The PLL requires some time to synchronize to the phase of the data stream. You can select when the CDR algorithm returns clock edges:

| | |
|---|---|
| "All" | All clock edges are used. |
| "After initial sync." | The clock edges of the synchronization time are discarded; results are gathered after initial synchronization of the CDR. Thus, meaningful TIE measurement results can be obtained. |

Remote command:
`TREFerence<m>:CDR:SOFTware:SELResults` on page 1084

# 9 Waveform analysis

This chapter describes general methods to check and analyze waveforms. These are:

- Zoom ..... 222
- Mathematics ..... 226
- History ..... 239
- Reference waveforms ..... 245
- Waveform histograms ..... 254

## 9.1 Zoom

The zoom magnifies a part of the waveform to view more details. The zoom is applied to all waveforms that are visible in a diagram.

For each diagram, you can define one zoom area.

- You can define the zoom by drawing, moving and adjusting the zoom area on the touchscreen.
- You can precisely define the size of the zoom area by entering start and stop values in a dialog box.

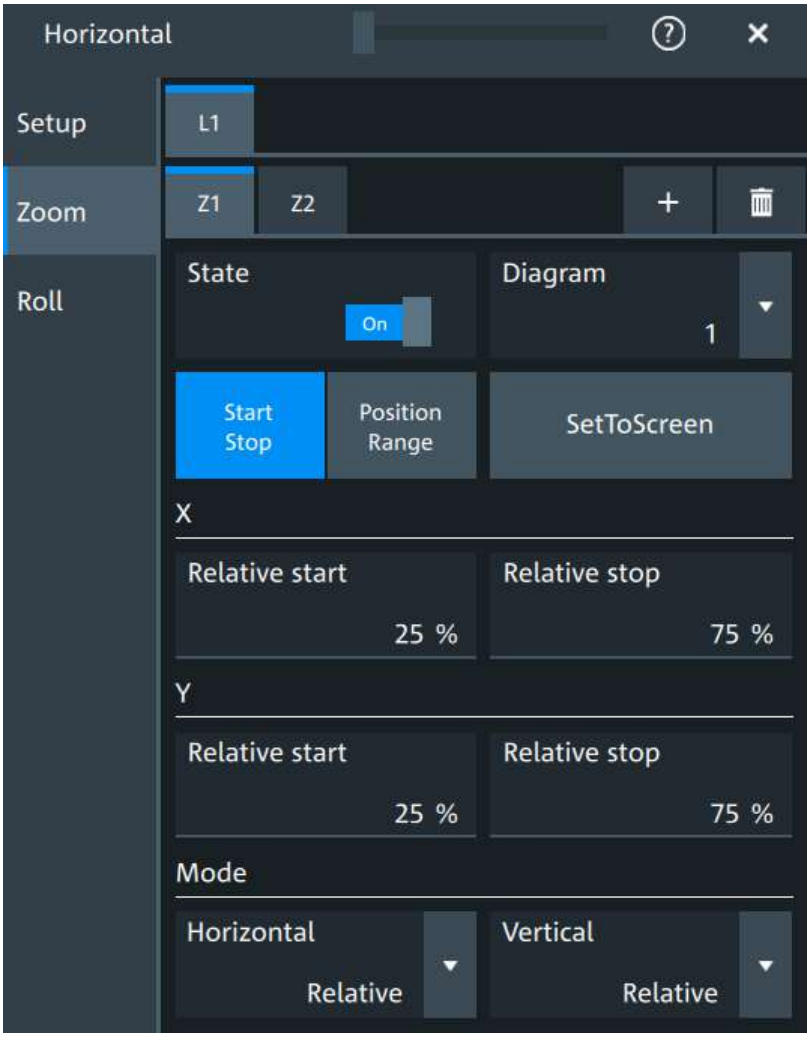
Zoomed waveforms are shown in separate zoom diagrams, in addition to the waveform diagrams. On the original waveform diagram, a rectangle indicates the zoomed section of the waveform - this is the zoom area. You can modify the zoom area by moving the rectangle as a whole, and by moving its edges. To toggle between these modes, tap the zoom area. You can also set exact positions.

Zoom areas can be used for gating, for example, to define a measurement gate. You can set the gate exactly to the limits of the zoom.

### 9.1.1 Zoom settings

Access: "Menu" > "Horizontal" > "Zoom" tab.

The "Zoom" tab allows you to specify start and stop values for the x- and y-axes. The acquired data within these ranges is zoomed.




**Tabx**
Selects one of the enabled SmartGrid layouts.

**Zx**
Selects the zoom window. You can add up to 8 different zooms.

**State**
Enables the zoom.

Remote command:
`LAYout<ly>:ZOOM<zo>[:ENABle]` on page 1090

**Diagram**
Indicates which of the waveform diagrams is selected for zooming. The number is displayed on the screen in the middle of each diagram.

Remote command:
`LAYout<ly>:ZOOM<zo>:SOURce` on page 1094

**Start Stop, Position range**
Selects how the window for the zoom diagram is defined. You can select between defining "Start Stop" values or "Position range".

**Start Stop**
The "Start Stop" tab allows you to specify start and stop values for the x- and y-axes. The acquired data within these ranges is zoomed.

According to the selected "Mode", absolute or relative values are used.

"Start", "Relative start"
Defines the lower limit of the zoom area on the axis.

"Stop", "Relative stop"
Defines the upper limit of the zoom area on the axis.

Remote command:
`LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:STARt` on page 1091
`LAYout<ly>:ZOOM<zo>:HORizontal:RELative:STARt` on page 1093
`LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:STOP` on page 1091
`LAYout<ly>:ZOOM<zo>:HORizontal:RELative:STOP` on page 1094
`LAYout<ly>:ZOOM<zo>:VERTical:RELative:STARt` on page 1097
`LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:STARt` on page 1095
`LAYout<ly>:ZOOM<zo>:VERTical:RELative:STOP` on page 1097
`LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:STOP` on page 1096

**Position range**
If "Position range" is selected, you specify the x and y position of center point of the zoom area plus a range for the x- and y-axes. The area defined by that point and the ranges is zoomed. You can set absolute values or relative values (in percent of the screen, depending on the "Mode" selection.

"Range" Defines the width or height of the zoom area.

"Position" Defines the x or y value of the centerpoint of the zoom area.

Remote command:
`LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:POSition` on page 1090
`LAYout<ly>:ZOOM<zo>:HORizontal:RELative:POSition` on page 1092
`LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:SPAN` on page 1091
`LAYout<ly>:ZOOM<zo>:HORizontal:RELative:SPAN` on page 1093
`LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:POSition` on page 1095
`LAYout<ly>:ZOOM<zo>:VERTical:RELative:POSition` on page 1096
`LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:SPAN` on page 1095
`LAYout<ly>:ZOOM<zo>:VERTical:RELative:SPAN` on page 1097

**Set to screen**
Sets the zoom area to the whole screen.

Remote command:
`LAYout<ly>:ZOOM<zo>:SSCReen` on page 1098

**Mode**
Defines if absolute or relative values are used to specify the "Horizontal" (x-axis) and the "Vertical" (y-axis) values.

Remote command:
`LAYout<ly>:ZOOM<zo>:HORizontal:MODE` on page 1092
`LAYout<ly>:ZOOM<zo>:VERTical:MODE` on page 1096

## 9.1.2 Zooming for details

**To define the zoom area on the touchscreen**

For graphical zooming, you use your finger on the screen.

1. On the toolbar, tap the "Add zoom" icon.

   The zoom overlay menu opens.

2. Draw a rectangle in the diagram that defines the zoomed area.
   While you drag your finger on the screen, a dotted rectangle indicates the current zoom area.

   The indicated area is magnified in a new zoom diagram. The original diagram shows the zoom area as a rectangle.

3. If the position of the zoom area is not correct, drag the rectangle in the overview to the correct position.

4. If the size of the zoom area is not yet ideal, tap the rectangle in the overview diagram.
   Now, 4 white lines indicate the edges of the zoom area. A dashed white line indicates the selected edge, which you can adjust.

5. Touch the edge that you want to move. Drag it to the required position.

**To create a zoom using the zoom dialog box**

► If you want to create a new, unconfigured zoom, tap the "Add" icon.



**To define the zoom area numerically using start/stop values**

1. Open "Menu" > "Horizontal".
2. In the "Zoom" tab, select "Start Stop".

3. Select a value for "Mode" > "Horizontal" to define "Absolute" or "Relative" x-axis values. Relative values cause the zoom area to adapt to the input values dynamically.
4. For "X", define the "Relative start" and "Relative stop" values. They define the lower and upper borders of the zoom area on the x-axis.
5. Select the "Mode" > "Vertical" to define "Absolute" or "Relative" y-axis values.
6. For "Y", define the "Relative start" and "Relative stop" values. They define the lower and upper borders of the zoom area on the y-axis.

   When you close the dialog box, the specified area is magnified in a new zoom diagram. The original diagram is displayed with the zoom area indicated as a rectangle.

**To define the zoom area numerically using position and range values**

1. Open "Menu" > "Horizontal".
2. In the "Zoom" tab, select "Position range".
3. Select "Mode" > "Horizontal" to define "Absolute" or "Relative" x-axis values. Relative values cause the zoom area to adapt to the input values dynamically.
4. Under "X" > "Position", define the x-value of the center point of the zoom area.
5. Under "X" > "Range", define the width of the zoom area.
6. Select the "Mode" > "Vertical" to define "Absolute" or "Relative" y-axis values.
7. Under "Y" >"Position", define the y-value of the center point of the zoom area.
8. Under "Y" > "Range", define the height of the zoom area.

   When you close the dialog box, the specified area is magnified in a new zoom diagram. The original diagram is displayed with the zoom area indicated as a rectangle.

## 9.2 Mathematics

Math waveforms are calculated waveforms. You can define up to 8 math waveforms and display them on the screen, and use it as source for further analysis.

It is calculated out of one or two analog channels, or another math waveform using several predefined operations. You can define up to 5 equations. The complete configuration is called equation set and can be saved for later use.

Activating math waveforms reduces the available memory.

You can analyze math waveforms in the same way as channel waveforms: use zoom, perform automatic and cursor measurements, and save as reference waveform.

You can store a math waveform as a reference waveform and restore it later. See Section 9.4, "Reference waveforms", on page 245.

### 9.2.1 Creating math waveforms

Math waveforms are created and displayed in addition to the channel and other waveforms. They can also be used for analysis, e.g. measurements, even if the math waveform is not active.

1. Open "Menu" > "Math" > "Setup".
   Alternatively, press the [Math] key.
2. Define the mathematic operation with one of the following methods:
   - Use basic operations in "Operator" subtab.
   - Define a filtered waveform in the "Filter" subtab.
   - Create a mathematic equation in the formula editor on the "Equation" subtab.
3. Enable "State".

   The math waveform is displayed on the screen. A green dot in the math waveform tab indicates that the waveform is active.

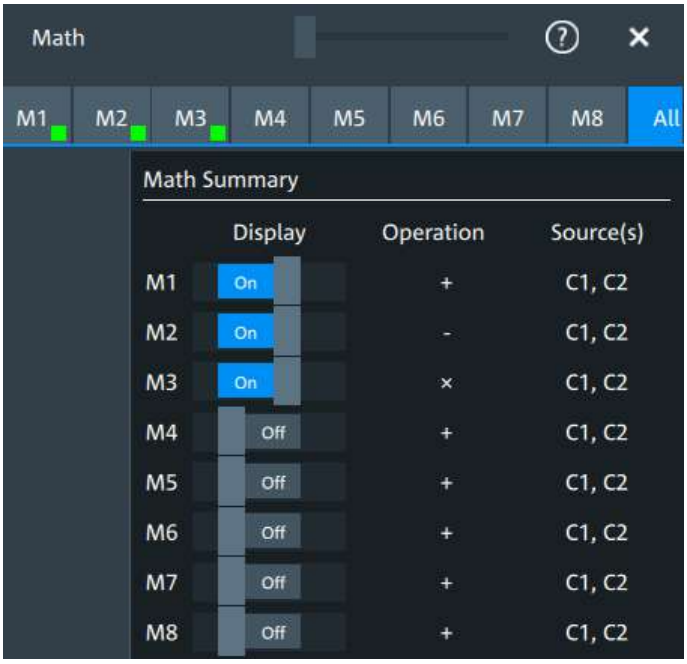
   
   
4. To change the vertical scaling of the math waveform, open the "Scale" tab.
5. Set the "Scale mode" to "Manual".
6. Enter the "Vertical scale" factor (per division). If necessary, add a "Vertical offset". Default is the automatic scaling.
7. Close the "Math" dialog box.

### 9.2.2 General math waveforms settings

Access: "Menu" > "Math" > "Setup" tab

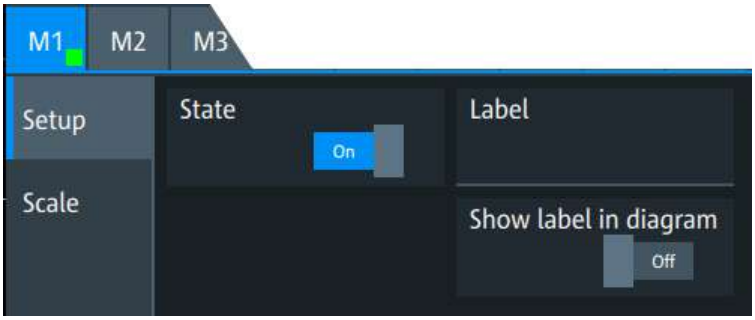
You can define up to 8 different math waveforms. Each waveform is defined in a separate tab in the "Math" dialog.

You can find a summary of the state and operations of all math waveforms in the "All" tab.




This section describes the general settings in the "Setup" tab. The subtabs for the definition of the mathematic operation are explained separately:

- Section 9.2.3, "Operator settings", on page 229
- Section 9.2.4, "Filter settings", on page 230
- Section 9.2.5, "Equations", on page 232




**State**
Activates the selected Math channel and displays the defined math waveforms.

Remote command:
`CALCulate:MATH<m>:STATe` on page 1098

**Label**
Defines a label for the selected math waveform.

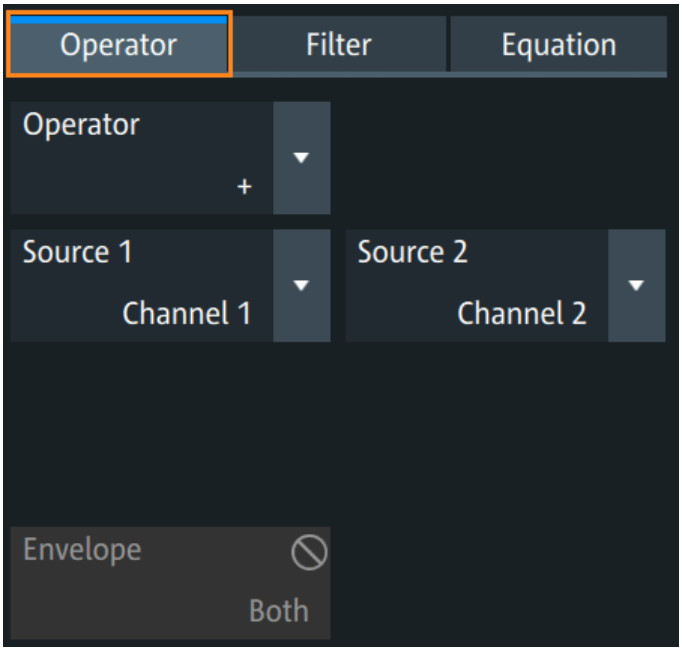
Remote command:
`CALCulate:MATH<m>:LABel` on page 1100

**Show label in diagram**
Displays a waveform label at the waveform in the diagram, with the content defined in "Settings" > "Appearance" > "Grid". See Section 5.3.2, "Grid", on page 99 for details on the content settings.

### 9.2.3 Operator settings

Access: "Menu" > "Math" > "Setup" tab > "Operator".

To create simple math waveforms, use the settings in the "Operator" subtab. It provides common math functions.




**Operator**
Defines the type of operation to be performed on the selected signal sources.

The following functions are available:

| | |
|---|---|
| "+" | Adds the values of 2 sources (channel or math waveform). |
| "-" | Subtracts source 2 from source 1. |
| "x" | Multiplies the two sources. |
| "/" | Divides source 1 by source 2. |
| "1/x" | Calculates the reciprocal of the source. |
| "- x" | Calculates the negative value of the source. |
| "\|x\|" | Determines the absolute value of the source. |
| "Δx/Δt" | Differentiates the source value relating to the time value.<br>Not possible on envelope waveforms and waveforms with "Peak detect" decimation. |
| "Integral" | Calculates the definite integral of the source. |

| | |
|---|---|
| "log(x)" | Calculates the logarithm of the source value based on 10. |
| "ln(x)" | Calculates the natural logarithm of the source value (based on e). |
| "ld(x)" | Calculates the binary logarithm of the source value (binary logarithm, based on 2). |
| "$x^2$" | Squares the source. If the source contains negative values that have been clipped, then the result contains positive clipping. |
| "√x" | Calculates the square root of the source. Note that the square root of a negative number is undefined and the result is clipped. |
| "Rescale" | Rescaling of x. |

Remote command:
`CALCulate:MATH<m>[:EXPRession][:DEFine]` on page 1098

**Source 1, Source 2**
Defines the signal source to be evaluated by the math function.

**a, b**
Define the values for the rescale function, if "Operator" = "Rescale"

| | |
|---|---|
| "a" | Multiplication factor |
| "b" | Offset of the signal source on the y-axis. |

**Noise reject**
Only available for "Operator" = "Δx/Δt".

Sets the number of neighboring samples that are skipped for differentiation.

To suppress noise effects during differentiation, it can be useful not to consider two directly neighboring points to calculate *dx* ($x_n$-$x_{n-1}$). Instead, some samples in-between are skipped and a point a few samples further is used (e.g. $x_n$-$x_{n-3}$).

**Envelope wfm selection**
Selects the upper or lower part of the input waveform for mathematic calculation, or a combination of both.

Remote command:
`CALCulate:MATH<m>:ENVSelection` on page 1101

### 9.2.4 Filter settings

Access: "Menu" > "Math" > "Setup" > "Filter" tab

The finite impulse response filter is a filter to create filtered math waveforms with lowpass, highpass, or bandpass. The filter requires additional settings.

You can set up the filter in the "Filter" tab, or enter the filter formula in the formula editor. The editor provides more filters than the "Filter" tab. For details, see Table 9-3.

You can also define the filter coefficients in a CSV file and use the file for filtering. See "Bandpass and bandstop FIR filter with CSV file" on page 231.

In remote control, filter settings are defined in the expression string, see `CALCulate:MATH<m>[:EXPRession][:DEFine]` on page 1098.



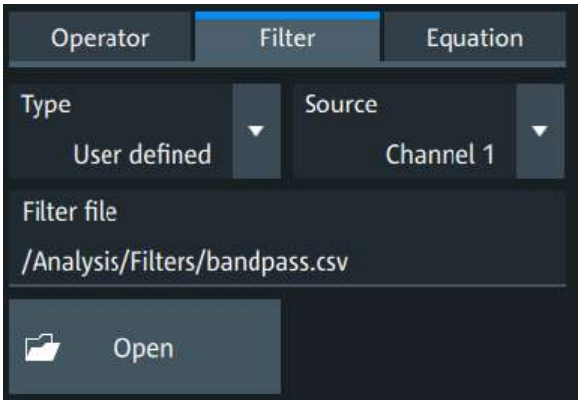

**Bandpass and bandstop FIR filter with CSV file**

To define a bandpass or bandstop, you need a CSV file that contains the comma-separated filter coefficients in one row. The maximum number of filter coefficients is 100 000 taps. No other parameters are allowed in the file.

To create the CSV file, we recommend using the MATLAB Filter Design & Analysis tool. In the tool, enter the filter type, filter order and filter frequencies. Alternatively, you can create the filter and the CSV file with Python.

Make sure to set the sample frequency in MATLAB or Python and the sample rate at the oscilloscope to the same value. If the values differ, the filter is shifted in frequency.

On the MXO 3, select "Menu" > "Math" > "Setup" > "Filter", the filter "Type" = "User defined", and load the CSV file.




All settings on the "Filter" tab are explained below. You can also enter the appropriate function in the formula editor, see Table 9-3.

**Type**
Selects if the filter is a highpass, lowpass, or user-defined filter (bandpass/bandstop filter defined in a CSV file).

**Source 1**
Selects the input channel of the signal.

**Cut-off**
Sets the limit frequency for the filter.

**Cutoff frequency for lowpass filter**

The cutoff frequency depends on the horizontal resolution and the filter characteristics. The cutoff frequency range for the lowpass filter is:

$f_c = (0.01 ... 0.2) * f_a$ for Gaussian FIR filter

$f_c = (0,01 ... 0.4) * f_a$ for rectangular FIR filter

**Cutoff frequency for highpass filter**

The highpass filter is always rectangular. The cutoff frequency range for the highpass filter is:

$f_c = (0,01 ... 0.4) * f_a$

Where: $f_c$ = cutoff frequency to be set for the lowpass filter, and $f_a$ = reciprocal of the resolution, or sample rate of the postprocessing system (including, for example, interpolation).

**Characteristics**
Selects if a Gaussian or a rectangular shape is used for the lowpass filter. The high-pass filter always uses a rectangular shape.

**Filter file, Open**
If the "User defined" filter type is selected, choose the file with the filter coefficients. Select "Filter file" to type the path and filename, or tap "Open" to select the file in the file browser.

For information on filter files, see "Bandpass and bandstop FIR filter with CSV file" on page 231.

## 9.2.5 Equations

In the "Equation" tab, you can enter complex formulas to define a math waveform. The formula editor helps to enter formulas easily with correct syntax, using a large selection of operators and signal sources.

1. In the "Math" dialog, open the "Equation" tab.
2. Double-tap on "Math formula" to display the formula editor.

### 9.2.5.1 Math formula editor

Using the formula editor you can define math functions freely, using a selection of operators and signal sources.

9-1 shows an overview of the math formula dialog and its different sections. The separate functions are described below.

For a procedure to create equations, see Section 9.2.5.2, "Defining a formula in the formula editor", on page 237.




*Figure 9-1: Math formula editor overview*

1 = Math expression
2 = Numeric
3 = Unit
4 = Source
5 = Filters
6 = Algebra
7 = Math punctuation
8 = "Enter": press to complete the equation entry . The button is grayed out if the expression is not valid.
9 = Information display: additional information about the selected function is displayed.
10 = Cursor keys

### Math expression

Use the various buttons of the editor to assemble the equation. If you know the exact syntax, you can also type the equation. You can see immediately whether the input is complete and correct.

We recommend connecting a keyboard to type the equations.




*Figure 9-2: Math expression line*

1 = Indicates the math waveform that is configured, e.g "Math 1" = "M1"
2 = Math expression input field
3 = Displays the status of the current expression: valid or not
4 = Displays a list of the last 20 used expressions

### Buttons of the formula editor and their usage

***Table 9-1: Source***

| Unit | Description |
|---|---|
| Analog channels | All analog channel waveforms of the device |
| Other math waveforms | All math waveforms of the device |
| Track channels | All track waveforms of a device |
| Reference | All reference waveforms of the device for which the source of reference is an analog channel (e.g. R3) |

***Table 9-2: Unit***

| Unit | Description |
|---|---|
| A | Ampere: the unit for electrical current. |
| V | Volt: the unit for electrical difference of potential |
| W | Watt: the unit for power. It is used to quantify the rate of energy transfer. |
| Ohm | Ohm: the unit for electrical resistance |
| Siemens | Siemens: the unit for electrical conductance |
| S | Second: the unit for time |
| Hz | Hertz: the unit for frequency, the number of occurrences of a repeating event per second. |
| dBm | dBm: the unit for power level.<br>dBm is expressed in decibels (dB) with reference to 1 milliwatt (mW). |
| dB | dB: the unit for power gain, expressed in the base-10 logarithm of the ratio between two power levels. |
| dBµV | dBµV: the unit for voltage level.<br>dBµV is expressed in decibels (dB) relative to 1µV across 50Ω resistance. |
| dBV | dBV: the unit for voltage level.<br>dBV is expressed in decibels (dB) relative to 1V across 50Ω resistance |
| Degree | Degree: the unit for an angle.<br>A degree is a measure of angle equal to 1/360 of a full rotation. |
| rad | Radian: the unit for an angle.<br>2π radians make up a full rotation. |

If you type the unit, put it in square brackets, for example *C1[A]*C3[V]*. Additional parenthesizes are required if you want to assign the unit to the result of the equation, for example *(C1*C3)[W]*

*Table 9-3: Filters*

| Value | Usage/Comment *FormulaEditor expression* |
|---|---|
| FIR highpass | *FIR(highpass,source,limit,shape)*:<br>source = input channel of the signal,<br>limit = cutoff frequency<br>shape = rectangle for the highpass filter<br>**Example:**<br>*FIR(highpass,C2,3e+09,rectangle)*<br>Sets a rectangle highpass filter on Channel 2 with a 3 GHz cutoff frequency |
| FIR lowpass | *FIR(lowpass,source,limit,shape)*<br>source = input channel of the signal<br>limit = cutoff frequency<br>shape = Gaussian or rectangle for the lowpass filter<br>**Example:**<br>*FIR(lowpass,C1,1e+07,gaussian)*<br>Sets a Gaussian lowpass filter on Channel 1 with a 10 MHz cutoff frequency |
| FIR user defined | *FIR(userdef,source,path)*<br>source = input channel of the signal<br>path = string containing the path and filename of the filter file. The file contains the comma-separated filter coefficients.<br>**Example:**<br>*FIR(userdef,Ch1,"/run/media/usb/<MyDriveName>//bandpass.csv")*<br>Sets a bandpass on channel 1 with filter coefficients saved in the `bandpass.csv` file. |
| IIR lowpass | *IIR(lowpass,source,limit,shape,order)*<br>source = input channel of the signal<br>limit = cutoff frequency<br>shape = Bessel or Butterworth filter<br>order = feedback filter order, 1 or 2<br>**Example:**<br>*IIR(lowpass,C1,100000,bessel,1)*<br>Sets a Bessel lowpass filter of first order on channel 1 with 100 kHz cutoff frequency. |

*Table 9-4: Numeric*

| Icon | Description | Usage/Comment *FormulaEditor expression* |
|---|---|---|
| 0...9 | Numeric characters | |
|  <br>Backspace | Erases the character before the cursor | |
| Clr | Clear expression in editor | Restart editing |
| e | Math. constants, Euler number | Euler number: 2.7182... |
| π | Math. constants, Pi | Pi: 3.14159... |

| Icon | Description | Usage/Comment *FormulaEditor expression* |
|---|---|---|
| . | Decimal point | |
| ± | Changes the sign of the equation | |

***Table 9-5: Algebra***

| Icon | Description | Usage/Comment *FormulaEditor expression* |
|---|---|---|
| + | Addition | |
| - | Subtraction | |
| * | Multiplication | |
| / | Division | |
| $a^b$ | Exponentiation with base a | a: base, b: exponent<br>*a^b* |
| \|x\| | Absolute x value | *abs(x)* |
| √x | Square root of x | *sqrt(x)* |
| $x^2$ | x*x | *pow(x)* |
| 1/x | Reciprocal of x | *1/(x)* |
| d/dx | Derivation of x | *Derivation(x,y)*<br>With x = waveform and y = number of skipped samples (noise reject) |
| ∫xdx | Integral of x | *integral(x)* |
| ax+b | Scaling of x | *Rescale(x,a,b)*<br>With x = waveform, a= gain factor, and b= constant |
| $\log_{10}$ | Decimal logarithm (base 10) | *log(x)* |
| $\log_e$ | Natural logarithm (base e) | *ln(x)* |
| $\log_2$ | Binary logarithm (base 2) | *ld(x)* |

***Table 9-6: Math punctuation***

| Icon | Description | Usage/Comment *FormulaEditor expression* |
|---|---|---|
| ( | Left bracket | Enclose operands |
| ) | Right bracket | Enclose operands |
| , | Comma | Separates operands |
| [ | Left square bracket | Enclose unit |
| ] | Right square bracket | Enclose unit |

*Table 9-7: Cursor keys*

| Icon | Description |
|---|---|
| \|← | Moves cursor to beginning |
| ← | Moves cursor 1 step to the left |
| → | Moves cursor 1 step to the right |
| →\| | Moves cursor to end |

#### 9.2.5.2 Defining a formula in the formula editor

1. Open the "Menu" > "Math" dialog.
2. Select the "Equation" tab.
3. Double-tap the editing area.

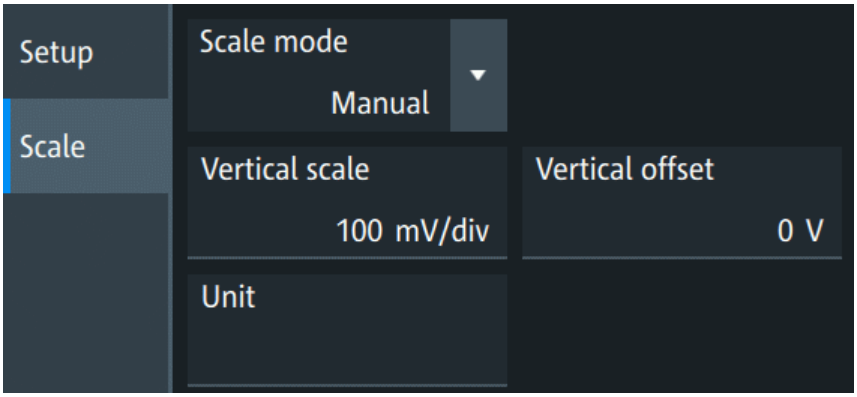
   The "Formula Editor" is displayed.
4. Enter the math formula including all required signal sources and operators by selecting the corresponding keys in the editor. For details on the available keys, see Section 9.2.5.1, "Math formula editor", on page 232.
5. To perform a rescaling function, proceed as follows:
   a) Select the rescaling function using the "ax+b" key.
   b) Behind the left bracket, insert the signal source using one of the following keys:
      - "Cx" for a channel
      - "Mx" for a math function
      - "Rx" for a reference waveform
   c) After the first comma, insert the "a" value, i.e. the scaling factor, using the number keys.
   d) After the second comma, insert the "b" value, i.e. the scaling offset, using the number keys.

   The resulting expression could be, for example: `rescale(C1,3,4)`
6. To insert a unit, press "Unit". Select a value from the list.

   If you type the unit, put it in square brackets, for example *C1[A]*C3[V]*. Additional parenthesizes are required if you want to assign the unit to the result of the equation, for example *(C1*C3)[W]*

### 9.2.6 Scale settings for math waveforms

Access:"Menu" > "Math" > "Scale" tab




**Scale mode**
By default, the vertical scale is adapted to the current measurement results automatically to provide an optimal display. However, if necessary, you can define scaling values manually to suit your requirements.

"Manual" Enter the required values for "Vertical scale" and "Vertical offset".

"Auto" "Vertical scale" and "Vertical offset" are read-only.

Remote command:
`CALCulate:MATH<m>:VERTical:SCALe:MODE` on page 1101

**Vertical scale**
Sets the scale of the y-axis in the math function diagram. The value is defined as "<unit> per division", e.g. *50 mV/div*. In this case, the horizontal grid lines are displayed in intervals of 50 mV.

If Scale mode is set to "Auto", this setting is read-only.

Remote command:
`CALCulate:MATH<m>:VERTical:SCALe[:VALue]` on page 1100

**Vertical offset**
Sets a voltage offset to adjust the vertical position of the math function on the screen. Negative values move the waveform up, positive values move it down.

If Scale mode is set to "Auto", this setting is read-only.

Remote command:
`CALCulate:MATH<m>:VERTical:OFFSet` on page 1101

**Unit**
Sets a user-defined unit for the math operation.

Remote command:
`CALCulate:MATH<m>:UNIT` on page 1100

# 9.3 History

The history accesses the data of previous acquisitions and provides them for further analysis.

## 9.3.1 About history

If a continuous acquisition runs, the captured data is stored in the sample memory and the current acquisition is processed and shown on the display. After the acquisition is stopped, the history accesses the captured samples that were stored, displays these samples as history waveforms, and makes them available for further analysis. It considers all channels that were enabled during the running acquisition.

When a new single acquisition is started, the memory is cleared and written anew. In continuous acquisition, stopping and starting the acquisition does not clear the memory, and the history memory is continued. When the memory is full, new acquisitions are stored, and the oldest are removed (first in, first out). The history memory is cleared when the acquisition parameters are changed. To clear the history memory on demand, use the "Clear" tool on the toolbar.

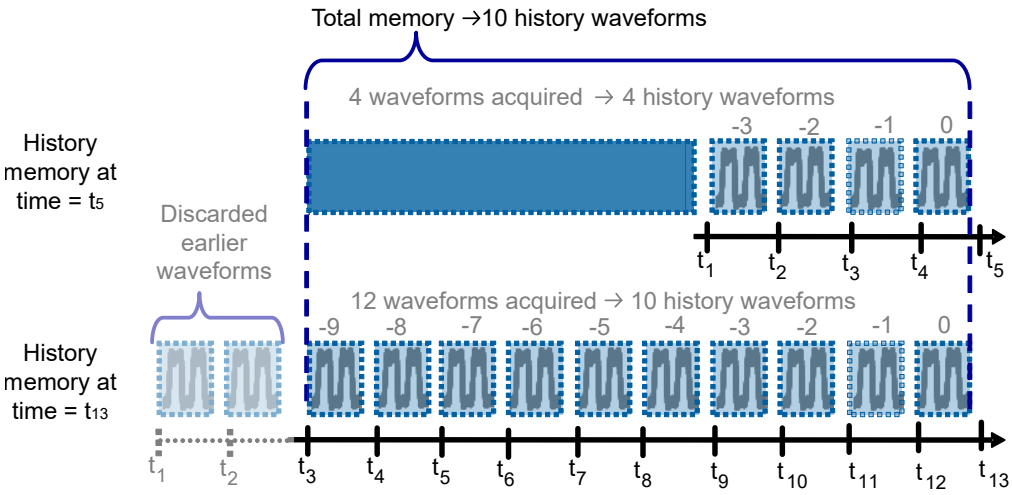



***Figure 9-3: History memory. In this example, the memory can store 10 waveforms.***

The number of stored history waveforms depends on the memory size, the number of enabled channels, and the record length. The shorter the record length, the less the number of channels, and the larger the memory, the more history waveforms are saved.

You can work with history waveforms in the same way as with the waveform of the latest acquisition: use zoom, cursors, measurements, create math waveforms and so on. You can also save a single history waveform for further analysis.

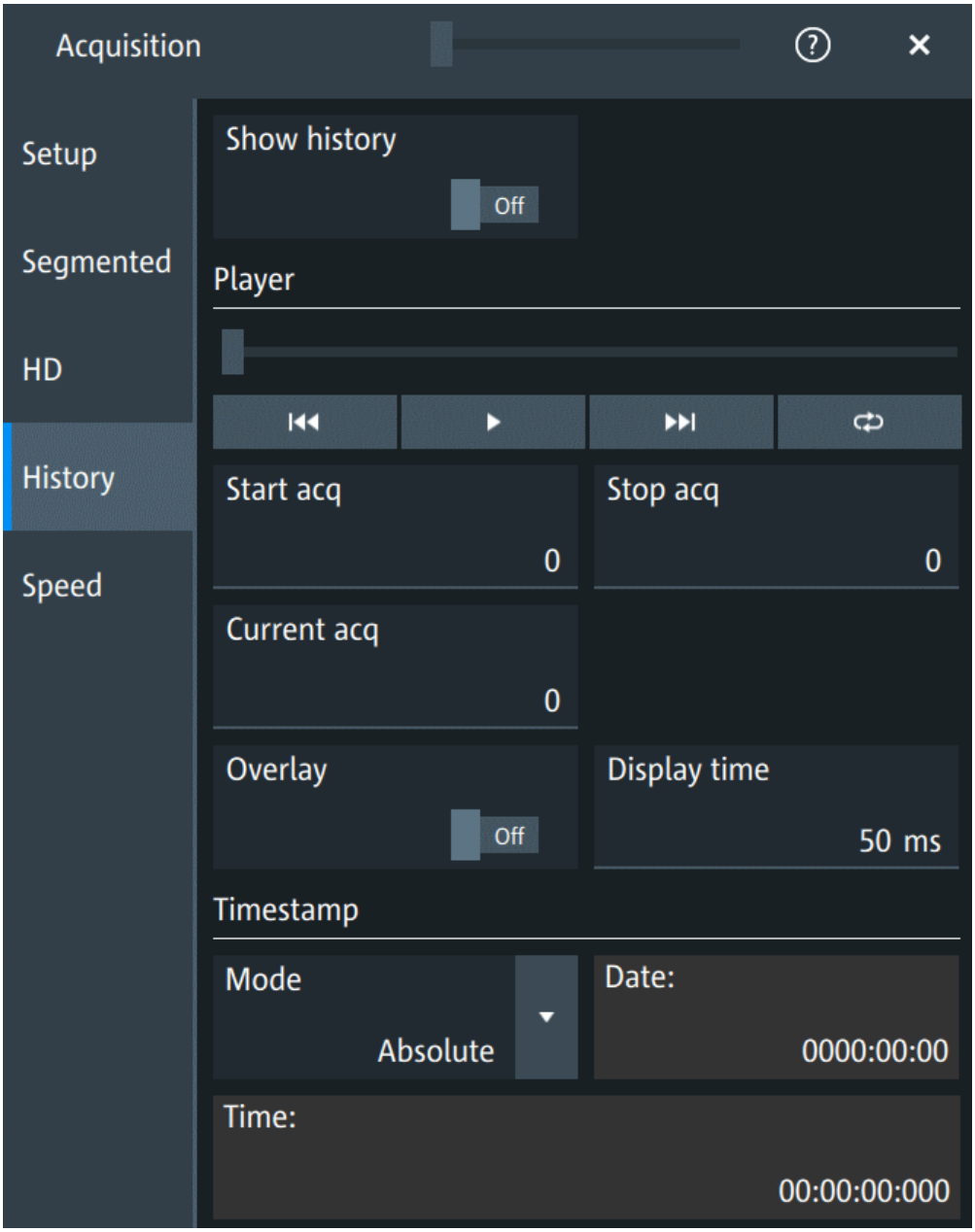
### 9.3.2 History setup

The "History" dialog contains the complete functionality on history viewing and information.

The most important information and functions are also provided in the quick-access history dialog box.

In the R&S ScopeStudio software, acquisition settings and functions that need an acquisition take effect if the channel source is a simulated waveform, created by the generator. If the channel source is loaded from a file, the acquisition-related settings can be imported with the data from the oscilloscope, either as a saveset, or included in a session. Changing imported acquisition-related settings has no effect.

Access: "Menu" > "Acquisition" > "History" tab.




Show history....241
Player....242
Start acq....242
Stop acq....242
Current acq....242
Overlay....242
Display time....242
Time stamp....243

**Show history**

Enables the history mode and allows you to save history waveforms to file.

The history display is enabled automatically when you press the [History] button.

To disable the history, you can also close the quick-access "History" dialog box.

Remote command:
ACQuire:HISTory[:STATe] on page 1105

**Player**
The player can be used to control the playback of the history waveforms.


Sets the oldest acquisition in the sample memory as "Start acq" and "Current acq".


Starts and stops the replay of the history waveforms from "Start acq" to "Stop acq".


Sets the newest acquisition in the sample memory as "Stop acq" and "Current acq". This acquisition always has the index "0".


"Auto repeat": If selected, the playback of the selected history segments repeats automatically. See also: "Auto repeat" on page 244.

Remote command:
ACQuire:HISTory:PLAY on page 1102

**Start acq**
Sets the index of the first (oldest) acquisition to be displayed or exported. The index is always negative.

Remote command:
ACQuire:HISTory:STARt on page 1103

**Stop acq**
Sets the index of the last (newest) acquisition to be displayed or exported. The newest acquisition of the complete acquisition series always has the index "0".

Remote command:
ACQuire:HISTory:STOP on page 1103

**Current acq**
Accesses a particular acquisition in the memory to display it, or to save it. The newest acquisition always has the index "0". Older acquisitions have a negative index.

If a history replay is running, the field shows the number of the currently shown acquisition.

Remote command:
ACQuire:HISTory:CURRent on page 1102

**Overlay**
Displays the segments with infinite persistence. Thus, you can see all data points of all displayed segments of a player cycle.

It is the same setting as "Infinite persistence" on page 105.

**Display time**
Sets the display time for one acquisition. The shorter the time, the faster the replay is.

The setting takes effect for usual history replay and the display of a fast segmentation series via the history.

Remote command:
ACQuire:HISTory:TPACq on page 1103

**Time stamp**
The time stamp shows the time of the currently displayed history acquisition. Thus, the time relation between acquisitions is always available.

The time stamp "Mode" can be absolute or relative:

- In "Absolute" mode, the instrument shows the date and the daytime of the current acquisition.
- In "Relative" mode, the time difference to the newest acquisition (index = 0) is shown.

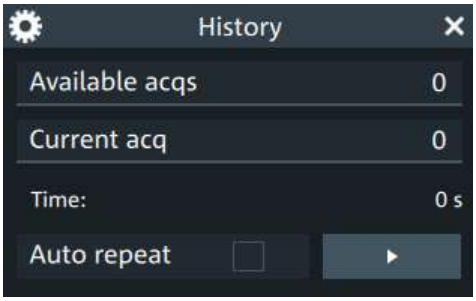
During history replay, the time value is displayed and updated if the replay speed ("Display time") is slow enough, that is 40 ms or slower.

The quick-access history dialog box always shows the time according to the "Mode" that is selected in the "Acquisition" > "History" tab.

Remote command:
ACQuire:HISTory:TSDate? on page 1104
ACQuire:HISTory:TSABsolute? on page 1104
ACQuire:HISTory:TSRelative? on page 1104
ACQuire:HISTory:TSRReference? on page 1105

### 9.3.3 Quick access history dialog

The quick access history dialog is visible when the history is active. Closing the dialog disables the history display ("Show history" = "Off")




Available acqs....243
Current acq....244
Time....244
Auto repeat....244
Play....244

**Available acqs**
Number of acquisitions that is saved in the memory and available for history viewing. It is also the number of acquisitions in a fast segmentation acquisition series.

Remote command:
ACQuire:AVAilable? on page 994

**Current acq**
Index of the current acquisition, same as Current acq in the history setup.

**Time**
Shows the timestamp of the current acquisition. See "Time stamp" on page 243.

**Auto repeat**
If selected, the playback of the selected history segments repeats automatically.

Remote command:
`ACQuire:HISTory:REPLay` on page 1103

**Play**
Starts and stops the replay of the history waveforms. See also: "Player" on page 242.

## 9.3.4 Using history

You can access the history of waveforms in two ways:

- Display a particular acquisition.
- Replay all or a part of the saved waveforms to track the signal run.

**To open the history and get information**

1. Press the [History] key on the front panel.

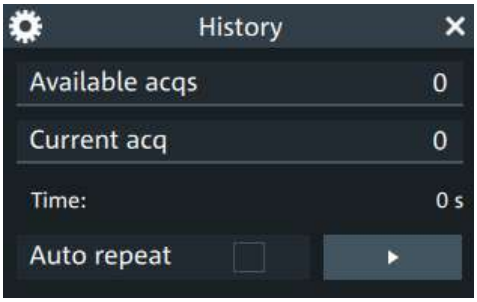
   A running acquisition is stopped, the history mode is enabled and the quick-access "History" dialog box is displayed. The [History] key lights up as long as the history mode is active.

2. Open the full configuration dialog box:
   - Tap the icon.
   - Open "Menu" > "Acquisition" dialog > "History" tab.

**To display a particular acquisition**

You can analyze a selected history waveform, and save it to a file.

► In the quick-access "History" dialog, enter the number of the required acquisition in the "Current acq" field. The newest acquisition always has the index "0", older acquisitions have a negative index.



Alternatively, you can configure and start the history display from the "History" configuration dialog:

1. Open the "History" configuration tab.
2. If the history mode is off (the [History] key is not illuminated), select "Show history".
3. Drag the slider to the required acquisition. The current number is shown in the "Current acq" field.
   Alternatively, enter the number of the required acquisition in the "Current acq" field.

**To save a single history waveform**

1. Select and display the required history waveform as described in "To display a particular acquisition" on page 244.
2. Save the selected waveform using "Save/recall" > "Save" tab > "Waveform". For details, see Section 13.4, "Waveform data", on page 455.

**To replay history waveforms**

If you want to see the complete acquisition series without any setup, simply tap in the quick-access "History" dialog box. For specific analysis of history data, use the history "Viewer" setup.

1. Open the "History" configuration dialog tab.
2. If the history mode is off, enable "Show history".
3. Tap  to start.

**To exit the history**

► Choose one of the following ways:
- Close the quick-access "History" dialog box.
- In the "History" configuration tab, disable "Show history".
- Start the acquisition.

## 9.4 Reference waveforms

Reference waveforms can be displayed in addition to the signal waveforms, saved to a file, and loaded back for further analysis.

You can configure up to 8 reference waveforms to display stored waveforms. Various waveform types can be used as reference waveforms, for example, analog signals, math waveforms, or spectrum.

**Supported file formats**

Reference waveforms can be loaded from REF, BIN and CSV files.

BIN is supported for waveforms that were saved on RTO2000, RTO6, RTE or RTP in BIN format.

Loading from CSV is supported under the following conditions:

- One analog channel waveform is stored in the CSV file. Envelope values are supported.
  Other waveform types than analog channels are not supported.
  Multichannel CSV files are not supported.
- The trigger time is not reconstructed, the trigger is always in the middle of the waveform.
- Clipping detection on measurements on a reference waveform from CSV is not supported.

### 9.4.1 Working with reference waveforms

- "To open the "Reference waveform" dialog" on page 246
- "To load a reference waveform" on page 246
- "To display and configure a reference waveform" on page 246
- "To save a reference waveform" on page 247

**To open the "Reference waveform" dialog**

► There are several ways to open the "Reference waveform" dialog:

- In the "Menu" > "Apps" > "General" tab, select "Reference".
- Select the "+ Ref" signal activator on the signal bar.
- Press the [Ref] key

**To load a reference waveform**

**Note:** Reference waveforms can be loaded from REF files and CSV files with single channel waveforms.

1. Open the "Reference waveform" dialog as described in "To open the "Reference waveform" dialog" on page 246.
2. Select the tab for the reference waveform that you want to load ("R1/2/3/4/5/6/7/8").
3. In the "Setup" tab, select "Open".
4. In the file browser dialog, select the "File type": "*.ref" or "*.csv".
5. Select the file.
6. Tap "Open".

   The selected waveform is loaded to the specified reference waveform.

**To display and configure a reference waveform**

1. Open the "Reference waveform" dialog as described in "To open the "Reference waveform" dialog" on page 246.
2. Select the tab for the reference waveform that you want to display, e.g. "R1".

3. Load a stored reference waveform as described in To load a reference waveform.
   Alternatively, select a source that you want to display as a reference:
   a) In the "Setup" tab, select "Source".
   b) Select the "Source" from the selection list. The list shows all active waveforms that you can save as references.
   c) Tap the "Create/Update" button to update the current reference waveform with the source data.
4. Tap the "Show" button.
   The reference waveform is displayed on the screen.
5. A reference waveform can have its own scaling settings or it can be scaled according to the source settings. By default, the scaling of the reference waveform is coupled to the source settings. Also, you can stretch it or compress it in vertical and horizontal directions.
   If necessary, change the settings on the "Vertical" and "Rescale" tabs of the "Reference waveform" dialog.
   To restore the original settings, tap the "Set to original" in the "Vertical" tab.
   For a description of the scaling settings, see Section 9.4.2.4, "Rescale settings", on page 252.

**To save a reference waveform**

1. Open the "Reference waveform" dialog as described in "To open the "Reference waveform" dialog" on page 246.
2. Select the tab for the reference waveform that you want to store, e.g. "R1".
3. Display and configure the reference waveform as described in "To display and configure a reference waveform" on page 246.
4. Tap "File extension" to select the file format.
   Note that reference waveforms can be loaded from REF files and CSV files only.
5. To save the waveform, tap "Save as".
6. Enter a filename and select the directory. The file type is defined according to the selection in the previous steps.
   The source settings of the reference waveform and the current scaling settings are stored to the specified file.

## 9.4.2 Settings for reference waveforms

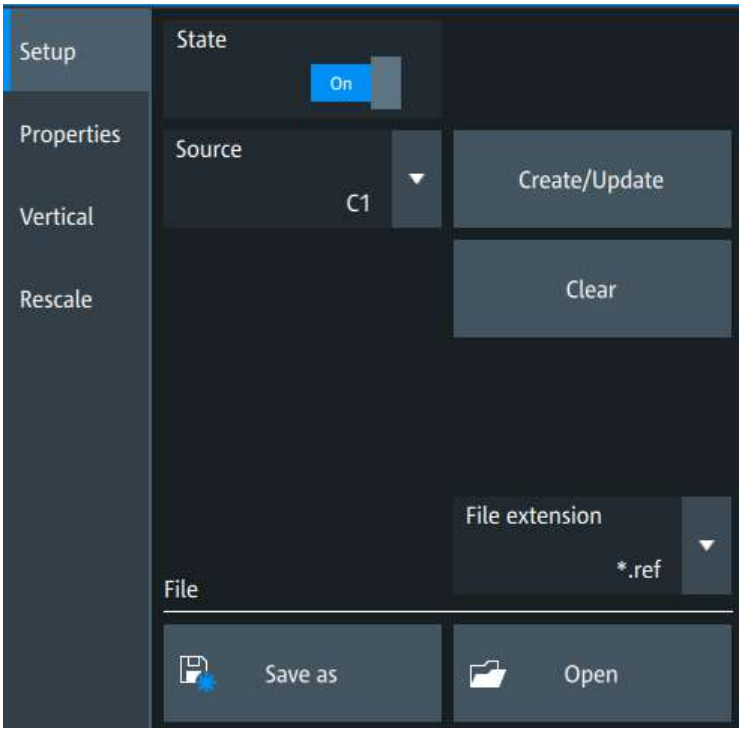
To compare waveforms and analyze differences between waveforms, you can use up to 8 reference waveforms.

You can save an unlimited number of reference waveforms and load them for further use.

The display of a reference waveform is independent from the display of the source waveform; you can move, stretch and compress the curve vertically and horizontally.

### 9.4.2.1 Reference waveform setup

Access: "Menu" > "Apps" > "General" tab > "Reference" > "Setup" tab




In the "Setup" tab, you select the target reference waveform and its source.

**R1/2/3/4/5/6/7/8**
Each tab contains the settings for one of the available reference waveforms.

**Show**
Enables the display of the reference waveform in the diagram. Before you can display it, create the reference waveform.

Remote command:
`REFCurve<rc>:STATe` on page 1109

**Source**
Selects the source waveform from the active waveforms, e.g. input channels, math waveforms, or spectrum.

Remote command:
`REFCurve<rc>:SOURce` on page 1108

**Create/Update**
Copies the selected source waveform with all its settings to the memory of the reference waveform. If there is a previously defined reference waveform in this memory, it is updated by the current source waveform. If the acquisition is running, the reference waveform is a snapshot.

A progress bar informs you about the process.

Remote command:
`REFCurve<rc>:UPDate` on page 1109

**Clear**
Deletes the selected reference waveform. It disappears from the display, and its memory is deleted.

Remote command:
`REFCurve<rc>:CLEar` on page 1106

**File extension**
Selects the file format.

**Save as**
Saves the reference waveform. The complete reference waveform is stored in a file and can be reloaded later.

A progress bar informs you about the process.

See also:
- Section 13.4, "Waveform data", on page 455
- Section 13.4.3, "Waveform export files", on page 462

Remote command:
`REFCurve<rc>:NAME` on page 1106
`REFCurve<rc>:SAVE` on page 1107
To abort the saving, use `REFCurve<rc>:ABORt` on page 1106.

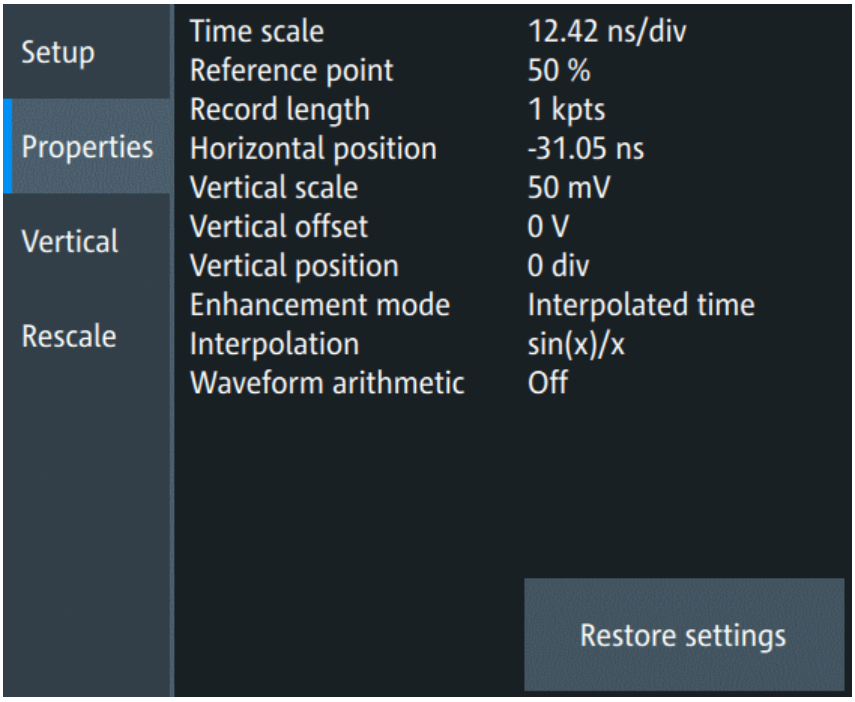
**Open**
Opens a file selection dialog box and loads the selected reference waveform file.

See "Supported file formats" on page 245.

Remote command:
`REFCurve<rc>:OPEN` on page 1107

#### 9.4.2.2 Reference waveform properties

Access: "Menu" > "Apps" > "General" tab > "Reference" > "Properties" tab




A reference waveform can be scaled, stretched and positioned in the diagram. The "Properties" tab shows the original settings of the reference waveform, which are stored together with the waveform data.

The following settings are displayed:

- Time scale
- Reference point
- Record length
- Position
- Scale
- Offset
- Position
- Enhancement mode
- Interpolation
- Waveform arithmetic

**Enhancement mode**
The enhancement mode is taken from the source waveform of the reference. The enhancement mode is set automatically by the instrument and depends on the sample rate. As long as the waveform sample rate is not higher than the ADC sample rate, the instrument works automatically in real time mode (RT). At higher waveform sample rates, the instrument changes to interpolated time mode (IT), where interpolation adds waveform points between the captured ADC samples by various mathematical methods, see Interpolation.

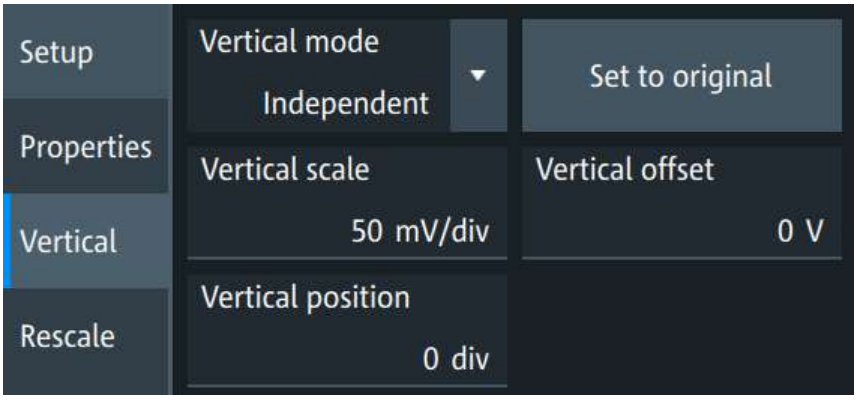
**Waveform arithmetic**
The waveform arithmetic is taken from the source waveform of the reference, and it cannot be changed for the reference. Waveform arithmetic builds the resulting waveform from several consecutive acquisitions of the signal. The arithmetic methods are envelope and average, which are set in the Acquisition mode.

**Restore settings**
Applies the original settings of the reference waveform to the horizontal and vertical settings of the selected waveform.

Remote command:
`REFCurve<rc>:RESTore` on page 1107

### 9.4.2.3 Vertical settings for reference waveforms

Access: "Menu" > "Apps" > "General" tab > "Reference" > "Vertical" tab




**Vertical mode**
Selects the type of vertical settings:

"Coupled to source" Vertical position and scale of the source are used.

"Independent" Scaling and position can be set specific to the reference waveform.

Remote command:
`REFCurve<rc>:VMODe` on page 1113

**Set to original**
Available, if "Vertical mode" = "Independent".

Restores the original vertical settings of the reference waveform.

Remote command:
`REFCurve<rc>:TOORiginal` on page 1113

**Vertical scale**
Available, if "Vertical mode" = "Independent".

Sets the vertical scale, which defines the displayed amplitude of the selected waveform.

The unit of the reference is the same as the unit of the source waveform.

Remote command:
`REFCurve<rc>:SCALe` on page 1110

**Vertical offset**
The vertical offset moves the reference waveform vertically. Enter a value with the unit of the waveform.

Remote command:
`REFCurve<rc>:OFFSet` on page 1107

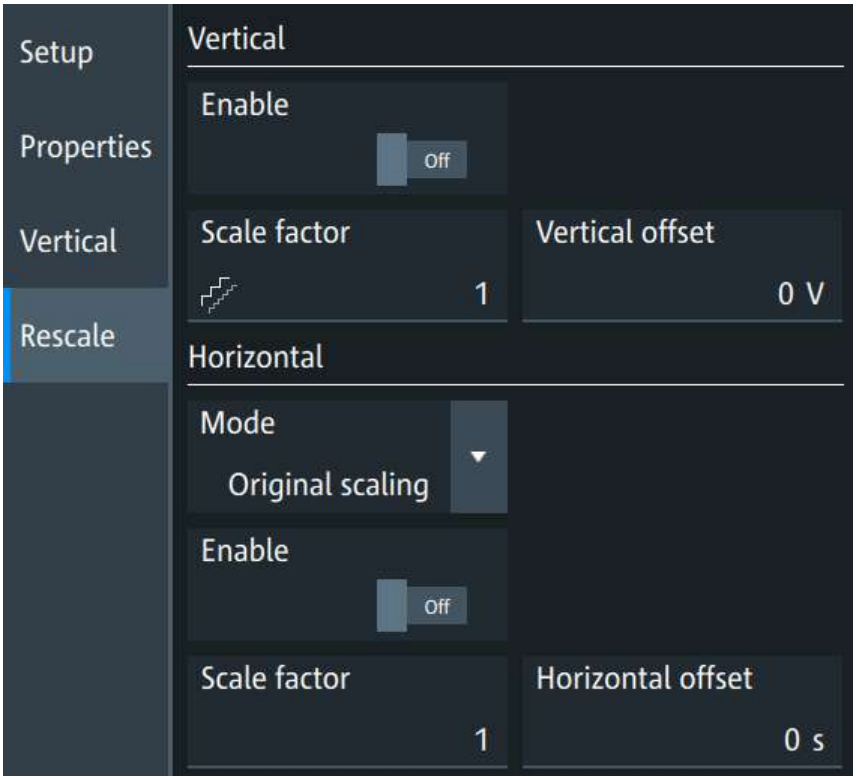
**Vertical position**
Available, if "Vertical mode" = "Independent".

Moves the reference waveform up or down in the diagram.

Remote command:
`REFCurve<rc>:POSition` on page 1110

### 9.4.2.4 Rescale settings

Access: "Menu" > "Apps" > "General" tab > "Reference" > "Rescale" tab




In the "Rescale" tab you can adjust the vertical and horizontal scaling of the reference waveform.

**Vertical**
Stretching and offset change the display of the waveform independent of the vertical scale and position.

**Enable ← Vertical**
If enabled, the vertical offset and factor are applied to the reference waveform.

Remote command:
`REFCurve<rc>:RESCale:VERTical:STATe` on page 1112

**Scale factor ← Vertical**
Sets the vertical scale factor. A factor greater than 1 stretches the waveform vertically, a factor lower than 1 compresses the curve.

Remote command:
`REFCurve<rc>:RESCale:VERTical:FACTor` on page 1112

**Vertical offset ← Vertical**
The vertical offset moves the reference waveform vertically. Enter a value with the unit of the waveform.

Like vertical offset of a channel waveform, the offset of a reference waveform is subtracted from the measured value. Negative values shift the waveform up, positive values shift it down.

Remote command:
`REFCurve<rc>:RESCale:VERTical:OFFSet` on page 1112

**Horizontal**
Stretching and offset change the display of the waveform independent of the horizontal settings of the source waveform and of the horizontal diagram settings.

**Mode ← Horizontal**
Selects the type of horizontal settings:

| | |
|---|---|
| "Original scaling" | Horizontal scaling and reference point of the source waveform are used. |
| "Adjust to X Axis" | The current horizontal settings of the diagram are used. |

Remote command:
`REFCurve<rc>:HMODe` on page 1110

**Enable ← Horizontal**
If enabled, the horizontal offset and factor are applied to the reference waveform.

Remote command:
`REFCurve<rc>:RESCale:HORizontal:STATe` on page 1111

**Scale factor ← Horizontal**
Sets the horizontal scale factor. A factor greater than 1 stretches the waveform horizontally, a factor lower than 1 compresses the curve.

Remote command:
`REFCurve<rc>:RESCale:HORizontal:FACTor` on page 1111

**Horizontal offset ← Horizontal**
Moves the waveform horizontally. Enter a value with a time unit suitable for the time scale of the diagram.

Positive values shift the waveform to the right, negative values shift it to the left.

Remote command:
`REFCurve<rc>:RESCale:HORizontal:OFFSet` on page 1111

## 9.5 Waveform histograms

Histograms are used to plot density of data. They show graphically how often which signal values occur. The histogram accumulates the measured y-values (vertical histogram), or the occurrence of a sample at a given time on the x-axis (horizontal histogram). The data is accumulated during the running acquisition cycle. They are a prerequisite for histogram measurements.

You can analyze the complete waveform by a histogram, or a part of it by defining a histogram window. The maximum count of a waveform value is assigned to the full height (histogram peak). All other count values are displayed relative to the maximum.

You can define up to 8 histograms. They are created quickly using the toolbar icon and overlay menu, or in the "Histogram" app dialog.

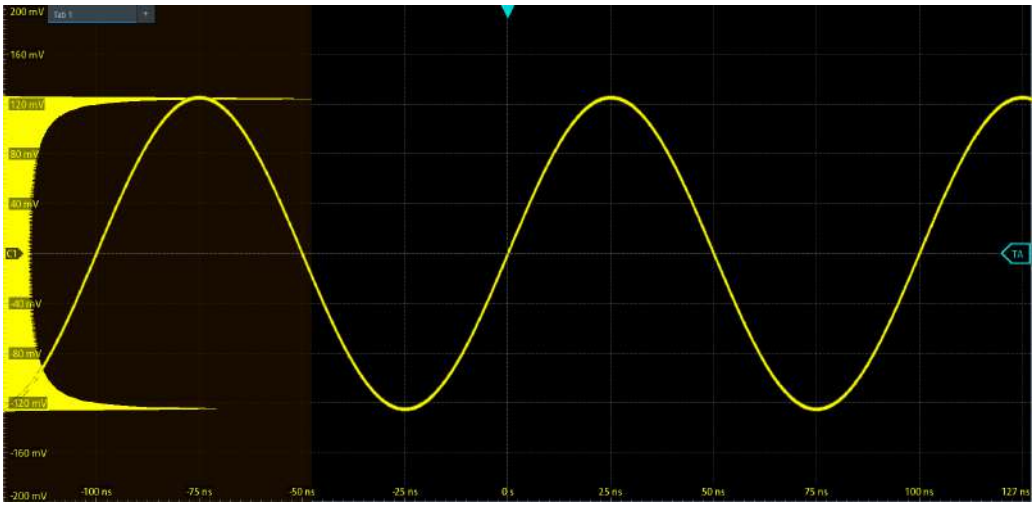



*Figure 9-4: Full histogram*

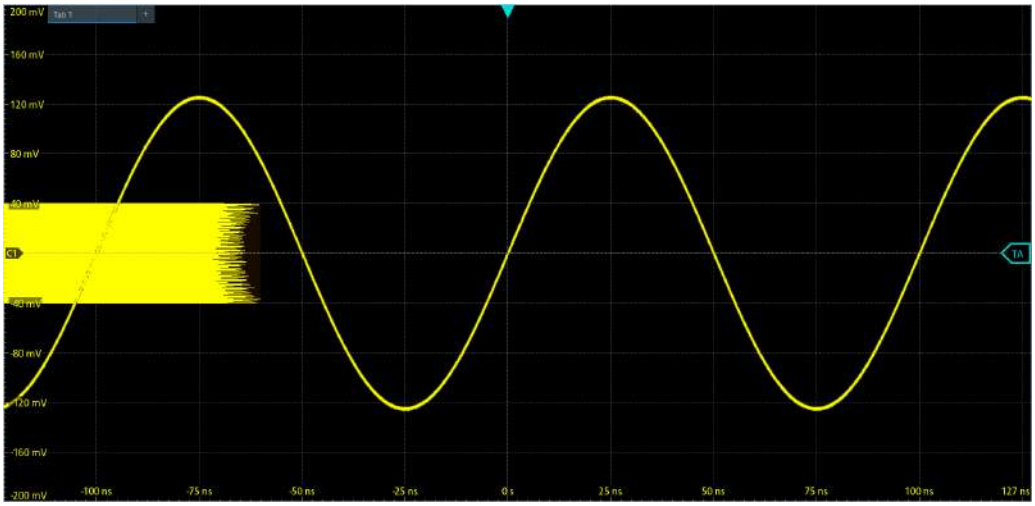



*Figure 9-5: Histogram window*

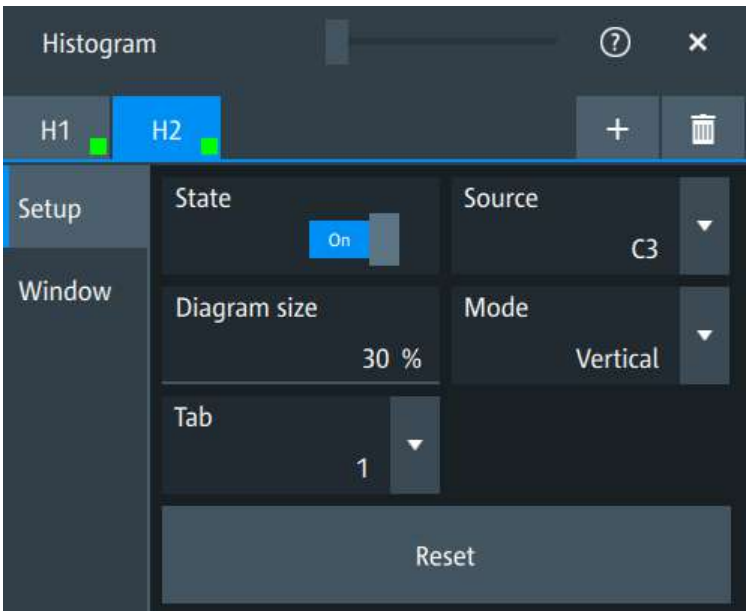
## 9.5.1 Histogram settings

Access: "Menu" > "Apps" > "General" tab > "Histogram".

The general histogram setup is done in the "Setup" tab. To analyze a part of the waveform, use the "Window" tab.

### 9.5.1.1 Setup for histograms

Access: "Menu" > "Apps" > "General" tab > "Histogram" > "Setup" tab



**State**
Enables or disables the histogram evaluation and display. The histogram settings are kept until the histogram is deleted.

Remote command:
`HISTogram<m>:ENABle` on page 1114

**Source**
Defines the source, the waveform that is analyzed by the histogram. Any analog channel waveform, math or reference waveform can be selected.

Remote command:
`HISTogram<m>:SOURce` on page 1115

**Diagram size**
Defines the size of the histogram in percent of the diagram width or height.

**Mode**
Defines the type of histogram.

A vertical histogram has horizontal bars that show the occurrence of the vertical values.

A horizontal histogram has vertical bars the show the occurrence of a sample at a given time on the x-axis. This histogram type is in particular relevant for histogram windows.

Remote command:
`HISTogram<m>:MODE` on page 1115

**Tab**
If several tabs are available in the diagram, you can select the tab that displays the histogram.

The histogram is always in the same diagram as the source waveform is. If the source waveform is not displayed in the tab selected for the histogram, the source is automatically enabled in this tab, together with the histogram. If you delete a tab with a histogram, the histogram is deleted also.

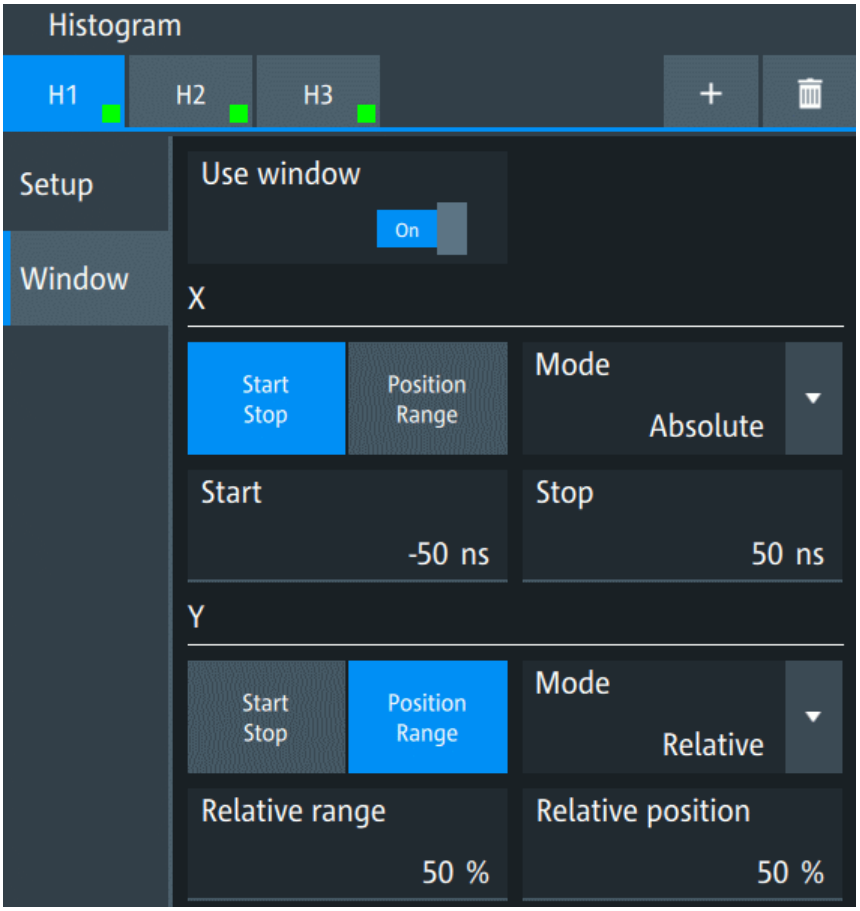
**Reset**
Restarts the calculation of the selected histogram.

To reset all histograms and measurements, select "Clear" on the toolbar.

Remote command:
`HISTogram<m>:RESet` on page 1115

#### 9.5.1.2 Window settings for histograms

Access: "Menu" > "Apps" > "General" tab > "Histogram" > "Window" tab




**Use window**
When you use a histogram window, the analyzed part of the source waveform is limited vertically and horizontally.

Remote command:
`HISTogram<m>:WINDow:ENABle` on page 1116

**Start Stop | Position Range**
Select whether you want to use start and stop values to limit the histogram window, or the center position and range of the window. The selections can be different for the vertical and horizontal limit.

**Mode**
Defines whether the window limits are entered as absolute or relative values.

Remote command:
`HISTogram<m>:WINDow:HORizontal:MODE` on page 1116
`HISTogram<m>:WINDow:VERTical:MODE` on page 1116

**Start, Stop, Range, Position**
Set the horizontal window limits as absolute values: start and stop, or center position and range of the window.

Remote command:
HISTogram<m>:WINDow:HORizontal:ABSolute:STARt on page 1117
HISTogram<m>:WINDow:HORizontal:ABSolute:STOP on page 1117
HISTogram<m>:WINDow:HORizontal:ABSolute:POSition on page 1117
HISTogram<m>:WINDow:HORizontal:ABSolute:SPAN on page 1117

**Relative start, Relative stop, Relative range, Relative position**
Set the horizontal window limits as relative values in % of the diagram width: start and stop, or center position and range of the window.

Remote command:
HISTogram<m>:WINDow:HORizontal:RELative:STARt on page 1117
HISTogram<m>:WINDow:HORizontal:RELative:STOP on page 1117
HISTogram<m>:WINDow:HORizontal:RELative:POSition on page 1117
HISTogram<m>:WINDow:HORizontal:RELative:SPAN on page 1117

**Start, Stop, Range, Position**
Set the vertical window limits as absolute values: start and stop, or center position and range of the window. The values remain unchanged if you change the unit of the y-axis.

Remote command:
HISTogram<m>:WINDow:VERTical:ABSolute:STARt on page 1118
HISTogram<m>:WINDow:VERTical:ABSolute:STOP on page 1118
HISTogram<m>:WINDow:VERTical:ABSolute:POSition on page 1118
HISTogram<m>:WINDow:VERTical:ABSolute:SPAN on page 1118

**Start, Stop, Range, Position**
Set the vertical window limits as relative values in % of the diagram height: start and stop, or center position and range of the window. The values remain unchanged if you change the unit of the y-axis.

Remote command:
HISTogram<m>:WINDow:VERTical:RELative:STARt on page 1118
HISTogram<m>:WINDow:VERTical:RELative:STOP on page 1118
HISTogram<m>:WINDow:VERTical:RELative:POSition on page 1118
HISTogram<m>:WINDow:VERTical:RELative:SPAN on page 1118

### 9.5.2 Creating histograms

**To create a histogram quickly with toolbar icons**

1. Select the waveform for which you need a histogram.
2. Tap the histogram icon on the toolbar.
3. Select the histogram type on the overlay menu: vertical or horizontal. See "Mode" on page 256.
4. Tap the diagram with the waveform to be measured, or draw a rectangle on the screen to define the window for histogram calculation.

The histogram range is shown in the diagram and a histogram with the selected waveform as a source is defined and displayed.

5. To adjust the histogram range:
   a) Double-tap the histogram.
   b) Tap "Setup" on the overlay menu.
   c) Adjust the start and stop values.

**To create and configure a histogram in the dialog box**

1. Select "Menu" > "Apps" > "General" tab > "Histogram".



2. To create a histogram, tap the "Add" icon in the upper right corner of the dialog box.
3. Select a "Source" for the histogram. The source can be any analog input signal, math or reference waveform.
4. Define the histogram "Mode": vertical for an amplitude, horizontal for a time-based histogram.
5. If you need to evaluate only a part of the waveform:
   a) Select the "Window" tab.
   b) Enter the start value and the stop value in x and in y direction, either as absolute or relative values. Alternatively, you can enter the center position of the window, and its range.
   c) Enable "Use window".
6. On the "Setup" tab, enable the histogram.

### 9.5.3 Analysis of waveform histograms

To view the histogram in more detail, use the zoom.

To get exact numeric results of the histogram, use the measurement functions. See Section 10.2.5, "Histogram measurements", on page 289.

You can export the histogram data to file, one histogram per file. See Section 13.5, "Histogram data export", on page 471.

For remote transfer of histogram data, use `EXPort:HISTogram<m>:DATA:HEADer?` and `EXPort:HISTogram<m>:DATA[:VALues]?`.

# 10 Measurements

Using the MXO 3 you can perform and display different measurements simultaneously, based on the active signal or math waveforms. The color of the results in the result table corresponds with the source waveform color.

## 10.1 Cursor measurements


- Cursors and results of cursor measurements ..... 260
- Using cursors ..... 261
- Settings for cursor measurements ..... 263


### 10.1.1 Cursors and results of cursor measurements

Cursor measurements determine the results at the current cursor positions. The cursors can be positioned manually, or can be configured to follow the waveform. You can measure on one waveform, on two different waveforms, or on multiple different waveforms (sources).

Up to 4 cursor sets can be configured and displayed. Each cursor set consists of a pair of horizontal or vertical cursor lines, or both. Cursor lines can be coupled so that the initially defined distance is always maintained.

The cursors are displayed in the diagrams of the source waveform only, or in all diagrams. For each measurement, labels can be defined for the cursors. By default, the cursor lines are labeled as Cu1.1 and Cu1.2.

How to set up cursor measurements is described in Section 10.1.2, "Using cursors", on page 261. The Section 10.1.3, "Settings for cursor measurements", on page 263 provides a detailed description of all settings.

For details on using the result table, see Section 4.11, "Displaying results", on page 76.

#### 10.1.1.1 Cursor measurements on time-based waveforms

The cursor for measurement on time-based waveforms returns the following results. The results are displayed automatically when a cursor measurement is enabled.




| Label | Description |
|---|---|
| "X1, X2" | Time at the position of the vertical cursors. |
| "Y1, Y2" | Vertical values of the waveform at the position of the horizontal cursors in V or A. |

| Label | Description |
|---|---|
| "Δx" | Difference between the vertical cursor (time) values |
| "1/Δx" | Inverse time difference |
| "Δy" | Difference between the horizontal cursor values |
| "Δy/Δx" | Slope of the waveform between the cursors (if measured on one source) |

For measurements on multiple waveforms, the Y-values are measured on each source waveform.

## 10.1.2 Using cursors

You can start cursor measurements in the toolbar, or using the [Cursor] key. For detailed configuration, use the "Cursor" dialog.

### 10.1.2.1 Starting a simple cursor measurement

**To add cursors using the toolbar**

1. Tap the "Add cursor" icon on the toolbar.




2. Select the channel that you want to apply the cursor to.
   Tap the waveform that you want to measure. Alternatively, you can draw a rectangle in the diagram to position the cursor lines.

   The cursor lines appear and the cursor results are displayed in the result table.

**To display cursors using the [Cursor] key**

1. Select the waveform that you want to measure.
2. Press the [Cursor] key.

   The cursor lines and the measurement results are displayed.

### 10.1.2.2 Configuring a cursor measurement

To modify the position of the cursor lines, you can drag the lines on the screen. In addition, various settings are possible to refine the measurement.

The complete configuration of cursor measurements is provided in the "Cursor" dialog. The settings of the dialog are described in Section 10.1.3, "Settings for cursor measurements", on page 263.

1. To open the "Cursor" dialog, use one of these ways:
   - Select "Menu" > "Cursor".

- Press the [Cursor] key.
- Double-tap in the results table (but not on a result).

2. Select the "Setup" tab.
3. Select the subtab for the cursor set that you want to use.
4. Select the Source mode.
5. Depending on the selected "Source mode", select one, two or several sources - the measured waveforms. Available sources are shown in the source list.
6. For single source and second source mode, select the "Type": X (vertical), Y (horizontal), or XY (both).
7. If the acquisition mode is set to "Envelope" or "Peak detect", select which horizontal cursor is positioned to the maximum and which to the minimum envelope values.
8. Define the exact position of the cursor lines:
   a) Enter the X-position for each vertical cursor.
   b) To position the horizontal cursors automatically, select "Track waveform". The Y-position follows the waveforms.
   In this case, Cu 1.1 indicates the current maximum, Cu 1.2 indicates the current minimum. If both horizontal and vertical cursor lines are displayed, the horizontal cursors are placed at the crossing points of the vertical cursor lines with the waveform. If you adjust the vertical cursor lines, the horizontal cursors follow.
   c) If "Track waveform" is off, enter the Y-position for each horizontal cursor.
9. To display the cursor in all diagrams that are in the same domain as the selected source (time or spectrum), enable "Show in all diagrams" in the "Advanced" tab.
10. To keep the distance between the vertical cursors when one cursor is moved, enable "X" or "Y" in the "Advanced" tab.
11. To set the cursors for a spectrum measurement to peak values, select the "Peak search" tab.
    Tap one of the buttons to place the cursors on the selected peak value. For details, see Section 10.1.3.3, "Peak search tab", on page 268.
12. Tap "Show cursor" in the "Setup" tab to activate the cursor measurement.

    The cursors lines and the results are displayed. For details on cursor measurement results, see Section 10.1.1, "Cursors and results of cursor measurements", on page 260.

#### 10.1.2.3 Configuring the cursor display

By default, the cursors are displayed as lines in the diagrams and labeled according to the syntax: Cu<cursor set number>.<1|2>

For example, the cursors for the cursor set 2 are labeled 2.1 and 2.2. The horizontal and the vertical cursors lines have the same labels.

You can change the default cursor display and labels.

**To set the cursor style**

1. Select "Menu" > "Cursor".
2. Select the subtab for the cursor set you want to configure.
3. Select the "Setup" tab.
4. Select the "Cursor style". See also: "Cursor style" on page 264.

**To add labels to cursor lines**

1. Select "Menu" > "Cursor".
2. Select the subtab for the cursor set you want to configure.
3. Select the "Advanced" tab.
4. Enter a label for "Vertical cursor 1", "Vertical cursor 2", "Horizontal cursor 1", "Horizontal cursor 2".
5. Enable "Show label".

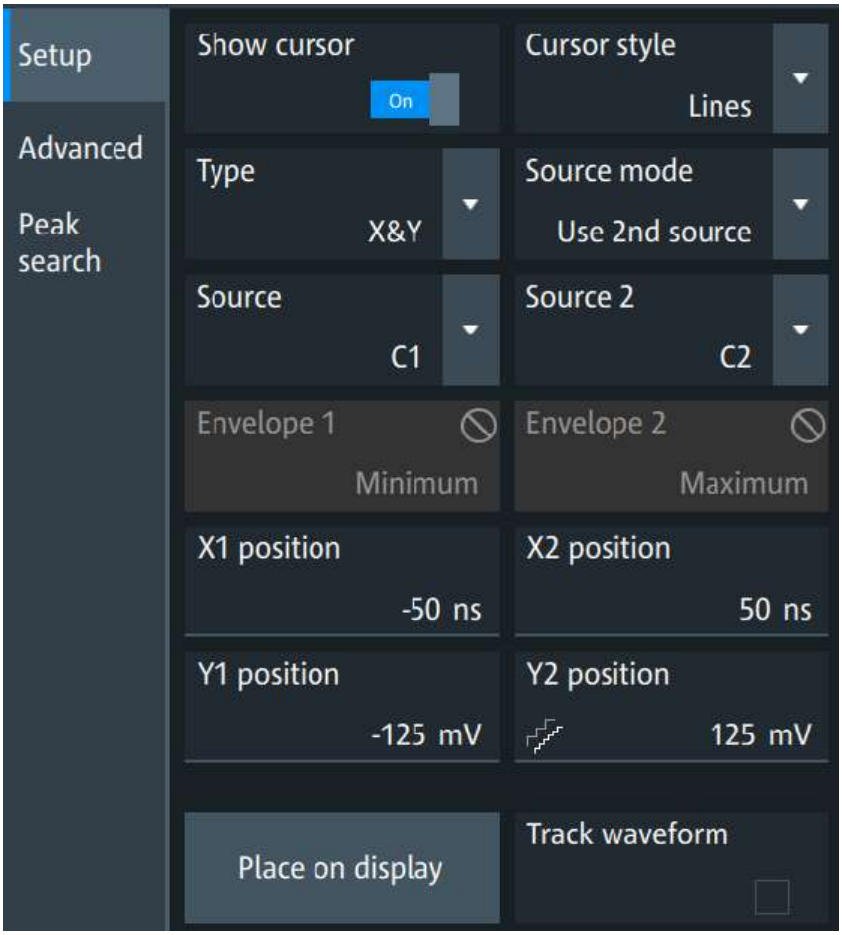
## 10.1.3 Settings for cursor measurements

Cursor measurements are configured in the "Cursor" dialog.

### 10.1.3.1 Setup tab

Access: "Menu" > "Cursor" > "Setup" tab

You can define up to 4 cursor sets (or cursor measurements). Each cursor set is configured on a separate tab.




**Show cursor**
Enables the selected cursor measurement.

Remote command:
`CURSor<cu>:STATe` on page 1177

**Cursor style**
Defines how the cursor is displayed in the diagram.

"Lines" The cursors are displayed as lines.

"Line & Rhombus"
The cursors are displayed as lines. The intersections of the cursors with the waveforms are displayed by rhombus-shaped points.

"VLine & Rhombus"
Vertical line and rhombus: the cursors are displayed as vertical lines. The intersections of the cursors with the waveforms are displayed by rhombus-shaped points.

"Rhombus" The intersections of the cursors with the waveforms are displayed by rhombus-shaped points.

Remote command:
`CURSor<cu>:STYLe` on page 1189

**Type**
Defines the cursor type to be used for the measurement.

"X" (vertical cursors )
Both vertical cursor lines are set automatically to the trigger position, and you can reposition them manually. Not available for measurements on multiple sources.

"Y" (horizontal cursors)
Horizontal cursors are positioned automatically along the waveform and can be adjusted manually. Not available for measurements on multiple sources.

"X&Y" (both vertical and horizontal cursors)
Horizontal cursors are positioned automatically along the waveform and vertical cursors are set to the trigger position. You can reposition all cursor lines manually.

Remote command:
`CURSor<cu>:FUNCtion` on page 1177

**Source mode**
Selects the number of sources that you want to measure. The dialog adapts to the selected mode.

| | |
|---|---|
| "Single source" | The cursor lines are set on one waveform. |
| "Use 2nd source" | The cursor lines are set on two waveforms. The second cursor lines Cu2 are set on the second source to measure differences between the two waveforms. |
| "Multiple sources" | Several waveforms are selected to be measured with one cursor set. The cursor type is always "X&Y", and "Track waveform" is active. For each waveform, the Y-value at the position of the vertical cursor line is measured. |

Remote command:
`CURSor<cu>:SMODe` on page 1177

**Source**
Defines the source of the cursor measurement. Various waveform types can be measured. The available sources are listed.

Remote command:
`CURSor<cu>:SOURce` on page 1178

**Source 2**
Selects a second source for the cursor measurement if "Source mode" is set to "Use 2nd source". The second cursor lines Cu2 measure on the second source. Using a second source, you can measure differences between two channels with cursors.

Remote command:
`CURSor<cu>:SSOurce` on page 1180

**Select sources**
Select the waveforms to be measured if "Source mode" is set to "Multiple sources". The selection dialog lists all possible active sources.

Remote command:
CURSor<cu>:SOURce on page 1178

**Envelope 1, Envelope 2**
Define which horizontal cursor is positioned to the maximum and which to the minimum envelope values.

Envelope selection is effective under the following conditions:
- The acquisition mode of the cursor source waveform is set to envelope or peak detect, see Acquisition mode.
- Both horizontal and vertical cursors are enabled (Type = "X&Y").
- Track waveform is enabled.

"Minimum" The horizontal cursor is set to the crossing point of the vertical cursor line with the minimum waveform envelope.

"Maximum" The horizontal cursor is set to the crossing point of the vertical cursor line with the maximum waveform envelope.

Remote command:
CURSor<cu>:X1ENvelope on page 1185
CURSor<cu>:X2ENvelope on page 1185

**X1 position, X2 position**
Define the left and right position of the vertical cursor lines.

Remote command:
CURSor<cu>:X1Position on page 1180
CURSor<cu>:X2Position on page 1181

**Y1 position, Y2 position**
Define the upper and lower position of the horizontal cursor lines.

If Track waveform is enabled, the user setting is disabled and the Y-position is determined by the instrument.

Remote command:
CURSor<cu>:Y1Position on page 1181
CURSor<cu>:Y2Position on page 1182

**Track waveform**
If enabled, the horizontal cursor lines track the waveform. If the waveform changes, e.g. during a running measurement, the cursors move along with it.

The function is available if both horizontal and vertical cursors are displayed. The horizontal cursor lines are positioned to the crossing points of the vertical cursors with the waveform. For cursor measurements on multiple sources, tracking is always active.

Tracking disables the Y-coupling (coupling horizontal cursor lines) and the Y-position settings.

In envelope or peak detect acquisition mode, the envelope is tracked.

Remote command:
CURSor<cu>:TRACking[:STATe] on page 1183

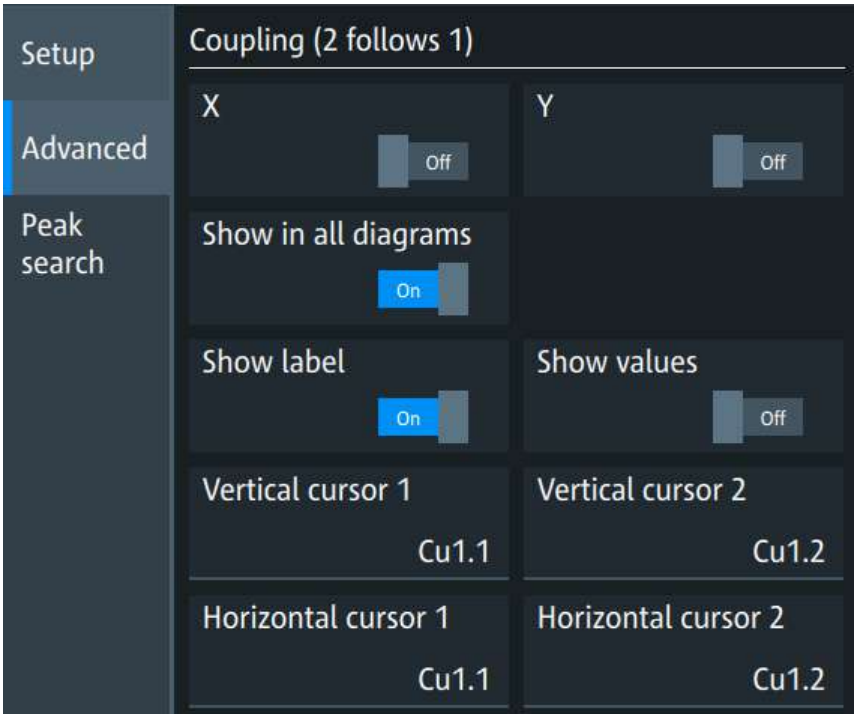
**Place on display**
Resets the cursors to their initial positions. Reset is helpful if the cursors have disappeared from the display or need to be moved for a larger distance.

Remote command:
`CURSor<cu>:SSCReen` on page 1182

### 10.1.3.2 Advanced settings

Access: "Menu" > "Cursor" > "Advanced" tab

The settings in the "Advanced" cursor tab configure the behavior and display of cursor lines, and labels for the lines.




**X, Y**
Couple the horizontal or vertical cursor lines so that the distance between the two lines remains the same if one cursor is moved. "X" couples the vertical cursor lines, and "Y" couples the horizontal cursor lines.

Remote command:
`CURSor<cu>:XCOupling` on page 1185
`CURSor<cu>:YCOupling` on page 1185

**Show in all diagrams**
Shows the enabled cursor measurements in all active diagrams of the time domain.

In the spectrum domain, the setting is disabled. The cursors are shown only on the source spectrum of the measurement.

Remote command:
`CURSor<cu>:SIAD` on page 1184

**Show label**
Shows the cursor labels in the diagram.

Remote command:
`CURSor<cu>:LABel` on page 1183

**Show values**
Shows the measured values in the cursor labels.

Remote command:
`CURSor<cu>:DISPlay:VALues` on page 1184

**Vertical cursor 1, Vertical cursor 2**
Defines the label to be displayed with the vertical cursor lines. By default, the cursors are labeled as Cu1.1, Cu1.2, Cu2.1, ...

Remote command:
`CURSor<cu>:VERTical<n>:LABel` on page 1184

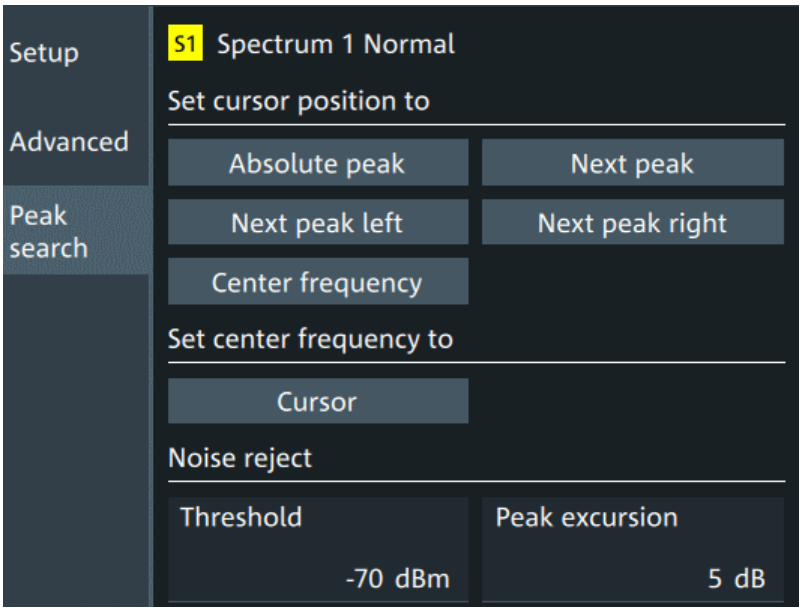
**Horizontal cursor 1, Horizontal cursor 2**
Defines the label to be displayed with the horizontal cursor lines. By default, the cursors are labeled as Cu1.1, Cu1.2, Cu2.1, ...

Remote command:
`CURSor<cu>:HORizontal<n>:LABel` on page 1183

### 10.1.3.3 Peak search tab

Access: "Menu" > "Cursor" > "Peak search" tab

The settings on this tab are only available in spectrum mode, i.e. the source of the cursor measurement is a spectrum. In this case, the cursors can indicate the results of a peak search on the waveform. Changing the noise reject settings, you can define which peaks are detected.



**Absolute peak**
Sets both cursors to the absolute peak value.

Remote command:
CURSor<cu>:MAXimum[:PEAK] on page 1188

**Next peak**
Cursor 2 is set to the next smaller absolute peak from the current position.

Remote command:
CURSor<cu>:MAXimum:NEXT on page 1189

**Next peak left**
Cursor 2 is set to the next peak to the left of the current position.

Remote command:
CURSor<cu>:MAXimum:LEFT on page 1188

**Next peak right**
Cursor 2 is set to the next peak to the right of the current position.

Remote command:
CURSor<cu>:MAXimum:RIGHt on page 1188

**Center frequency**
Sets the vertical cursor line Cu1 to the center frequency.

Remote command:
CURSor<cu>:FFT:TOCenter on page 1188

**Set center frequency to**
Sets the center frequency to the frequency value that is measured at cursor line Cu1.

Remote command:
CURSor<cu>:FFT:SETCenter on page 1188

**Threshold**
Sets an absolute threshold as an additional condition for the peak search. Only peaks that exceed the threshold are detected.

This setting is only available for spectrum waveforms. It is valid for cursor measurements, spectrum measurements and peak search.

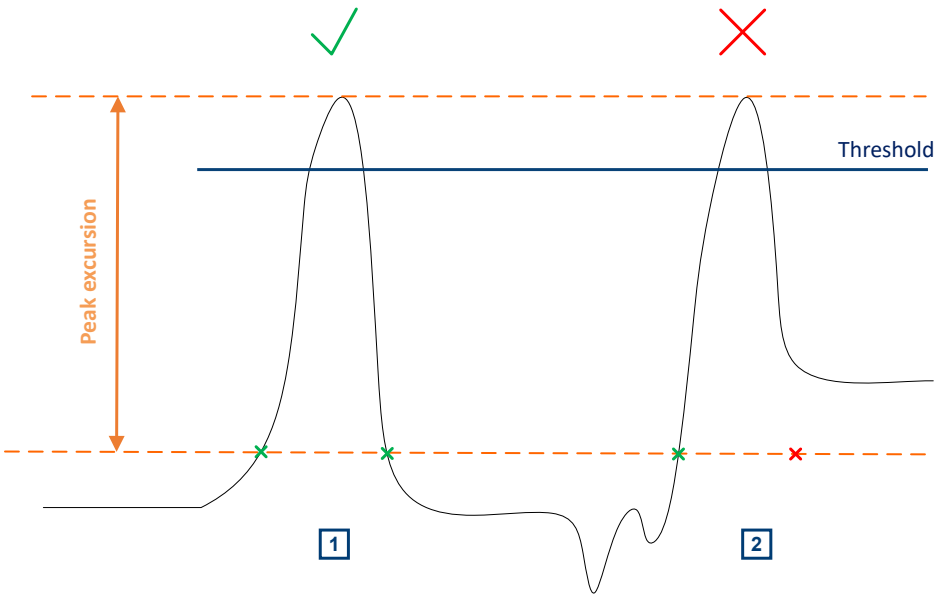
Remote command:
CALCulate:SPECtrum<sp>:THReshold on page 1198

**Peak excursion**
Defines a minimum level value by which the waveform must drop left and right of the local maximum to be listed as a peak. Enter a peak excursion value to omit close by peaks and list just the highest peak.

This setting is only available for spectrum waveforms. It is valid for cursor measurements, spectrum measurements and peak search.

**Example:**

In the figure below, 2 peaks are shown, that rise above the defined "Threshold". Peak 1 fulfills the defined "Peak excursion" value and is counted as a peak. Peak 2 does not fulfill the defined "Peak excursion" value on the right and is not counted as a peak.




Remote command:
`CALCulate:SPECtrum<sp>:PEXCursion` on page 1198
`CURSor<cu>:PEXCursion` on page 1189

## 10.2 Automatic measurements

You can perform up to 16 different measurements simultaneously. Available sources and measurement types depend on the selected category. For most measurement types, specific settings are available for configuration.

### 10.2.1 Measurement setup in general

#### 10.2.1.1 Starting an automatic measurement

There are several methods to start a measurement, each with slightly different effects:

- Using the "Measure" icon on the toolbar:
  See: "To start a measurement using the toolbar icon" on page 271.
- Pressing the [Measure] key on the front panel.

- Using the "Measurement" dialog.
  See: "To add a new measurement in the measurement dialog" on page 271.

**To start a measurement using the toolbar icon**

1. Tap the "Measure" icon on the toolbar.

   The measurement overlay menu opens.




2. Select the "Category".
3. Tap the measurements that you want to perform. You can scroll left and right through the measurement list to view all available measurements for the selected "Category".
4. Select the source that you want to apply the measurement on.
5. If needed, click "Setup" to configure further measurements.

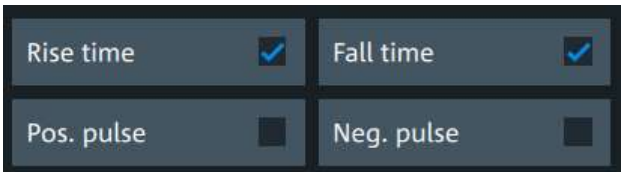
   The measurement results are displayed.

**To add a new measurement in the measurement dialog**

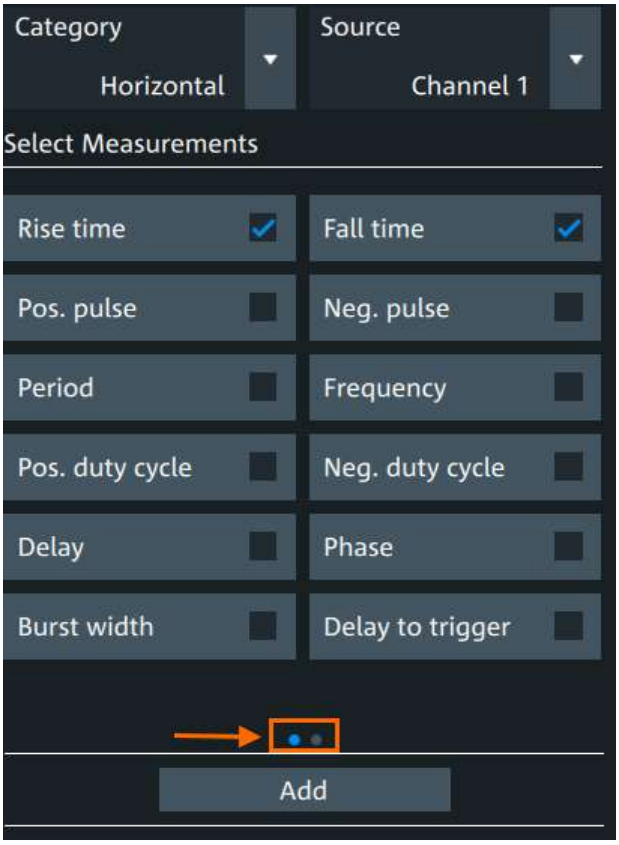
1. Press the [Measure] key to open the measurement dialog.
2. In the "Setup" tab, press "+ Add".

   A dialog opens to select the measurements.
3. Select the "Source" for the measurement.
4. Select the "Category" of the measurement that you want to add, e.g. "Horizontal".

   All available measurements for this category are displayed. For more details on the available measurement types, see the correspondig section.
5. Tap a measurement on the list to select it.
   Selected measurements are marked with a blue check mark in their checkbox.

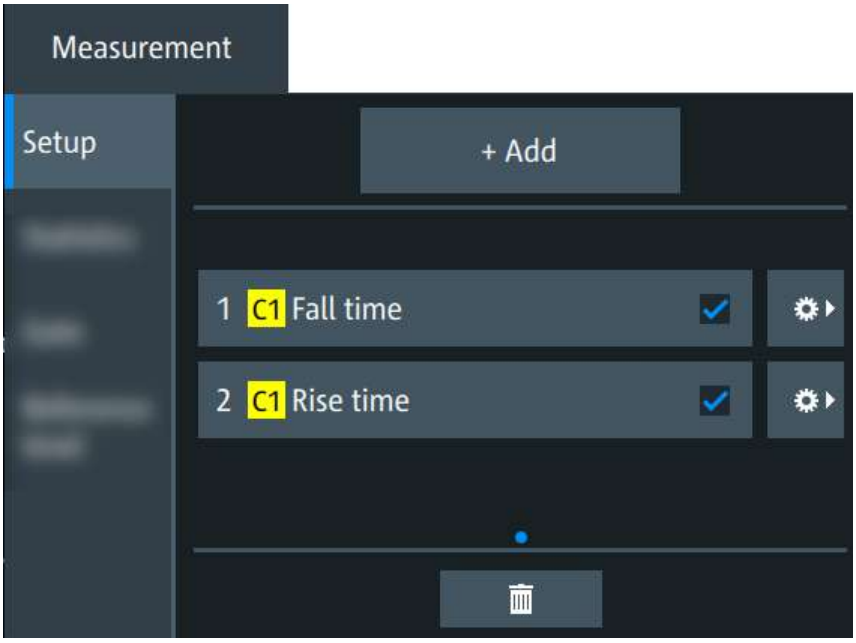



6. The measurements of some categories are listed on several tabs. To switch between the tabs, tap the points at the bottom of the list.




7. Tap "Add" to add the selected measurements.

   All selected measurements are active by default.



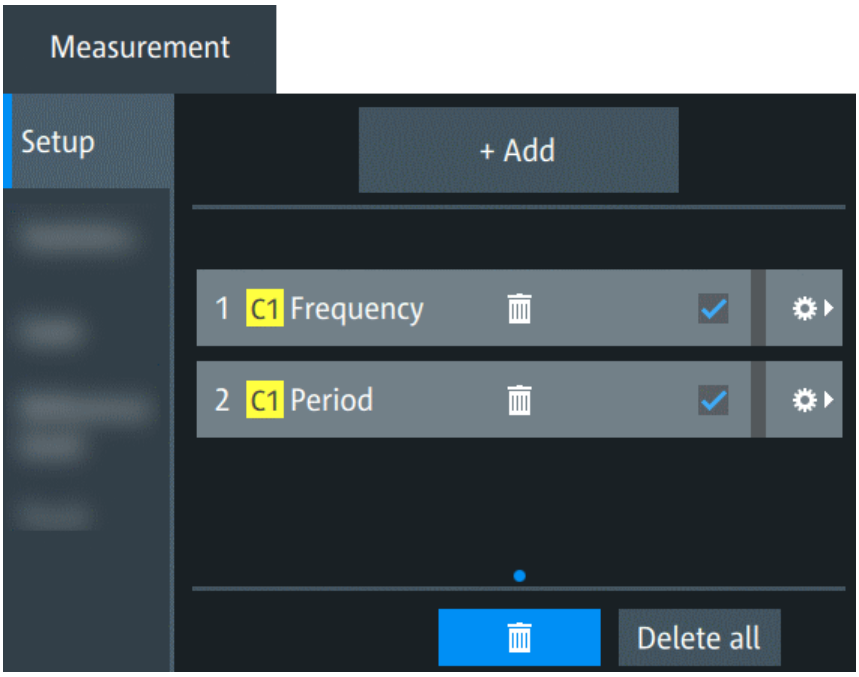

8. To deactivate a measurement while leaving it in the measurement list, deselect the checkmark.

9. To configure a measurement, select the (settings icon).

**To delete a measurement**

1. Press the [Measure] key to open the measurement dialog.
2. In the "Setup" tab, tap .

   A delete icon appears in the selection button of each measurement.

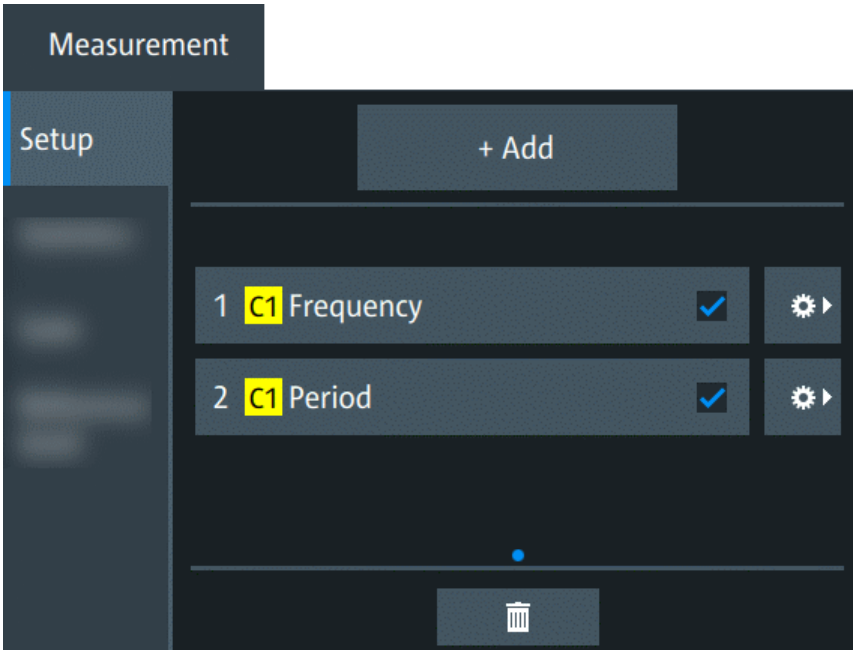



3. Tap the button of the measurement that you want to delete. Alternatively, tap "Delete All" to delete all measurements.

### 10.2.1.2 Setup tab

Access: "Menu" > "Measurement" > "Setup" tab.

The "Setup" tab shows the selected measurements. You can activate or deactivate the measurements. Tap the "Settings" icon to open the specific settings for the measurement type.




**+ Add**
Opens a dialog to select the measurements. See Section 10.2.1.3, "Add measurement dialog", on page 274.

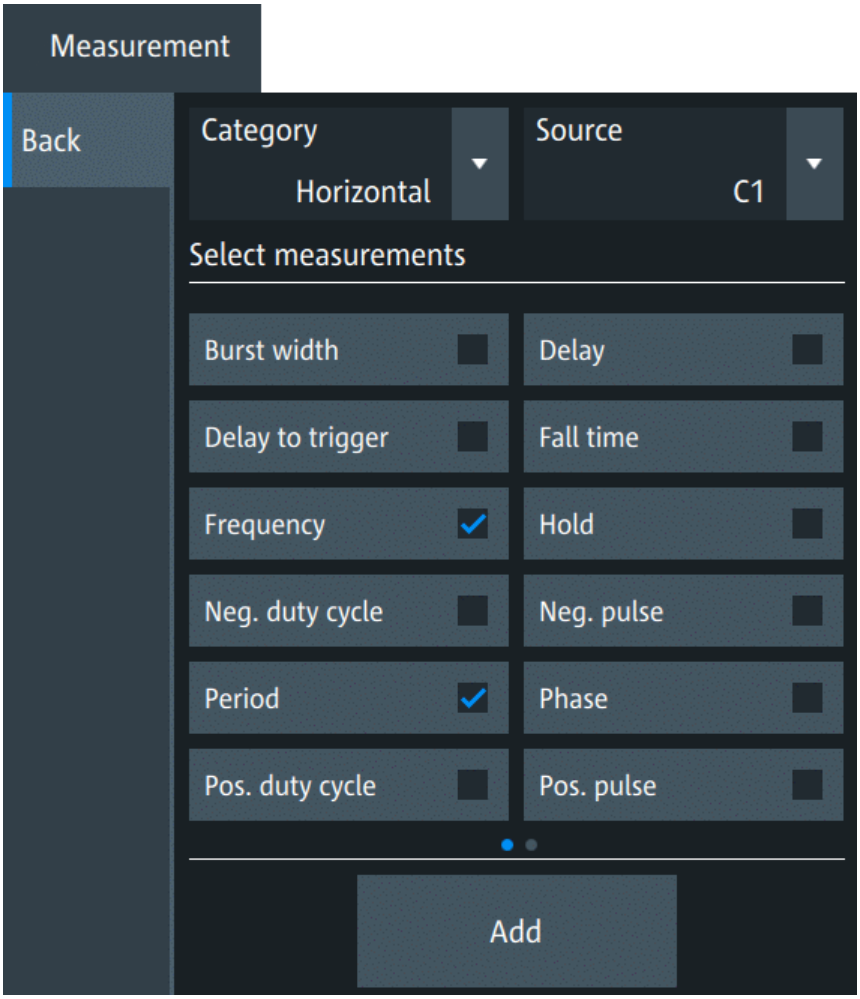
**Delete icon**
Enables the removal of measurements. You can delete a single measurement by tapping the "Delete" icon next on it.

You can also delete all current measurements with "Delete All".

### 10.2.1.3 Add measurement dialog

Access: "Menu" > "Measurement" > "Setup" tab > "+ Add".

In this dialog, you select the category and the measurements to be done.




**Category**
Selects the measurement category. Some measurements are listed in more than one category.

The following categories are available:

- Amp/time: amplitude and time measurements, includes the categories "Horizontal", "Vertical" and "Count"
- Jitter measurements (available if the jitter option is installed)
- Histogram: measurements on waveform histograms
- Protocol: advanced analysis of serial protocols with option R&S MXO3-K500

Remote command:
In remote control, the measurements are selected directly. A preselection of the category is not neccessary.

**Source**
Sets the source of the measurement.

Available sources depend on the selected category. For example, histogram measurements require a waveform histogram as source, and protocol measurements require a serial bus.

The source is applied to all measurements that are selected in the dialog. You can change it later for the individual measurement.

Remote command:
MEASurement<mg>:SOURce on page 1152

**Select measurements**
Select the required measurements of the selected category. The list can have several pages, which are represented by dots at the bottom of the list. Tap a dot to change the page.

Remote command:
MEASurement<mg>:MAIN on page 1154

**Add**
Adds the selected measurements and returns to the "Setup" tab.

The measurements are listed and activated by default. You can deactivate single measurements in the "Setup" tab.

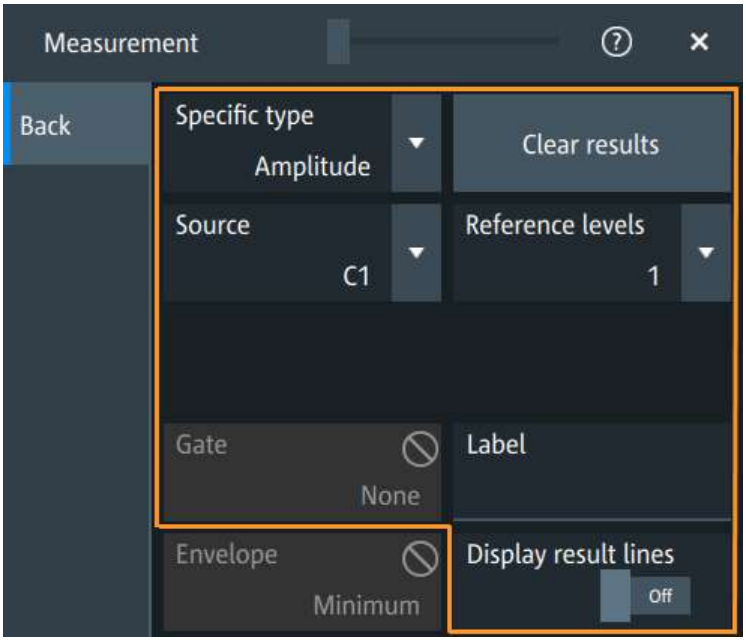
Remote command:
MEASurement<mg>[:ENABle] on page 1152
MEASurement<mg>:ACTive on page 1152

#### 10.2.1.4 Configuration of measurement types: common settings

When you select the settings icon of a measurement in the "Setup" tab, a dialog with more setting opens. Some of these settings are common for many or all measurement types, independent of their category, others are specific for the selected measurement type.

For category and type-specific settings, see:

- Section 10.2.4.2, "Settings for amplitude/time measurements", on page 284
- Section 10.2.6.2, "Settings for jitter measurements", on page 294
- Section 10.2.5.2, "Settings for measurements on histograms", on page 291
- Section 10.2.7.2, "Settings for protocol measurements", on page 300




This section describes the common settings.

**Specific type**
Selects the type of measurement that you want to define the settings for. The list provides all measurement types of the same category.

Remote command:
`MEASurement<mg>:MAIN` on page 1154

**Clear results**
Deletes all results of the selected measurement.

Remote command:
`MEASurement<mg>:CLEar` on page 1155

**Source**
Sets the source of the measurement.

Available sources depend on the category of the measurement. For example, histogram measurements require a waveform histogram as source, and protocol measurements require a serial bus.

In setup/hold measurements, this source is the "Data source".

Remote command:
`MEASurement<mg>:SOURce` on page 1152

**Reference levels**
Selects the set of reference levels that is used for the measurement and for the first measurement source. Each source of the measurement can have its own reference level set. For most measurements, one source is needed. For delay measurements, for example, two measurement sources are required, and each source can have its own reference levels.

Only the numbers of defined reference level sets are listed. Define the reference level set before you select it, see Section 8.1, "Reference level setup", on page 211.

Remote command:
`MEASurement<mg>:REFLevel<rl>:REFerence` on page 1159

**Gate**
Assigns a gate to the selected measurement to limit the measurement range. Define a gate before you select it, see Section 8.2, "Gate setup", on page 214.

Alternativley, you can assign the gates to measurements in the "Gate" tab, see Section 10.2.3, "Gate settings for measurements", on page 279.

Remote command:
`MEASurement<mg>:GATE` on page 1155

**Label**
Adds a user-defined label to the measurement.

**Display result lines**
Enables the measurement annotations for the selected measurement. These annotations are, for example, periods, maximum and minimum values, relevant reference levels, and more.

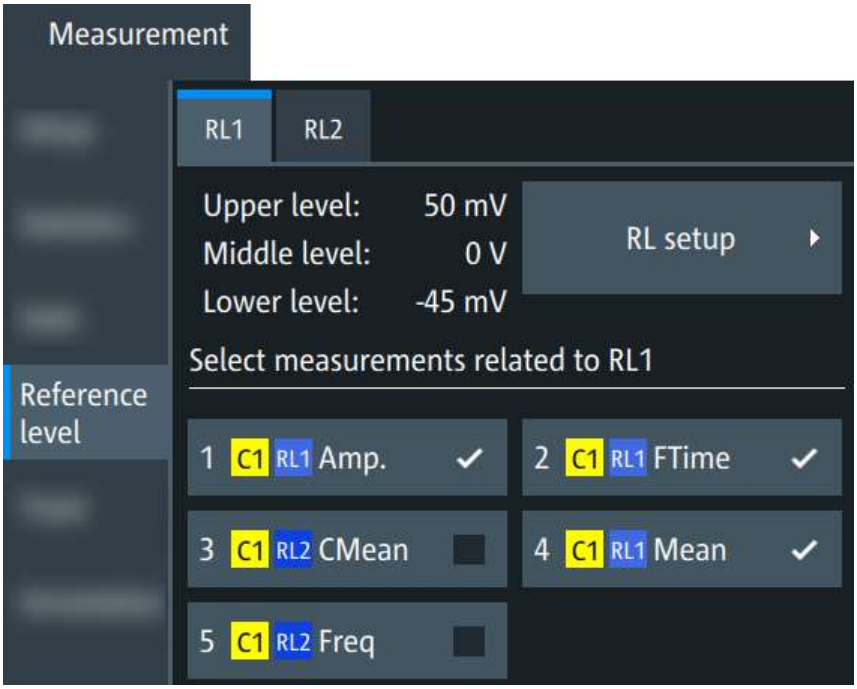
See also: Section 10.2.8, "Annotation for measurements", on page 302.

Remote command:
`MEASurement<mg>:DISPlay:RESults` on page 1155

### 10.2.2 Reference levels for measurements

Access: "Menu" > "Measurement" > "Reference level" tab

Measurements require reference levels to obtain the measurement points, e.g. time measurements or pulse count. The dialog shows the definition of the selected reference level set and all active measurements. If no reference level set has been defined before, or the correct definition is missing, select "RL setup" and configure the the reference level set before you assign it. See Section 8.1, "Reference level setup", on page 211 for details.




1. Select the RL subtab, e.g. "RL1".
2. Select the measurements that use the selected reference level set.

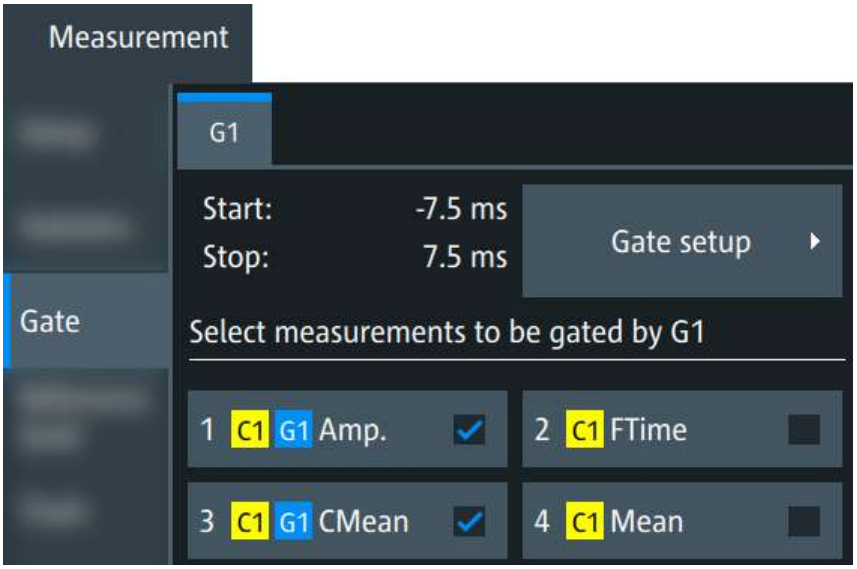
Remote command: `MEASurement<mg>:REFLevel<rl>:REFerence` on page 1159

### 10.2.3 Gate settings for measurements

Access: "Menu" > "Measurement" > "Gate" tab.

You can limit the measurements to a user-defined range of the waveform. The dialog shows the limits of the selected gate and all active measurements. If no gate has been defined before, or the correct gate is missing, select "Gate setup" and configure the gate before you assign it. See Section 8.2, "Gate setup", on page 214 for details.

1. Select the gate subtab, e.g. "G1".
2. Select the measurements to be limited by the selected gate.



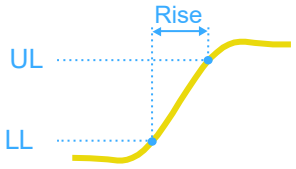
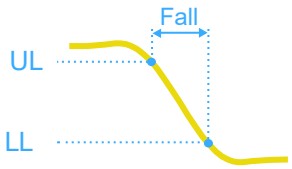
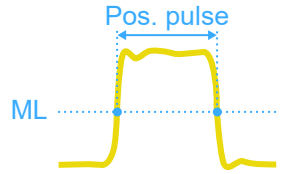

Remote command: MEASurement<mg>:GATE on page 1155.

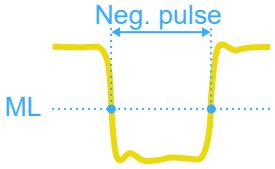
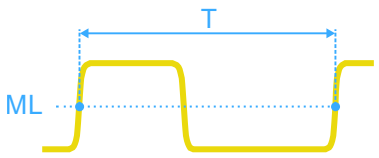
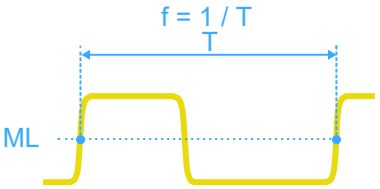
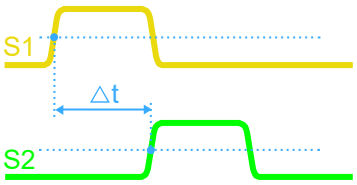
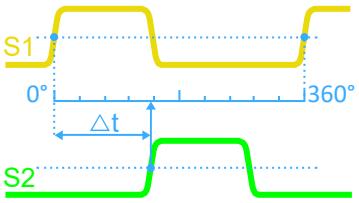
## 10.2.4 Amplitude/time measurements

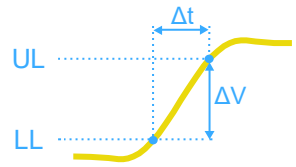
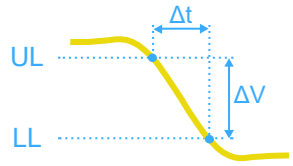
### 10.2.4.1 Overview of amplitude/time measurements

The MXO 3 provides many measurement types in the "Amp/time" category to measure time, amplitude and area characteristics, and to count pulses and edges.

#### Horizontal measurements (time)

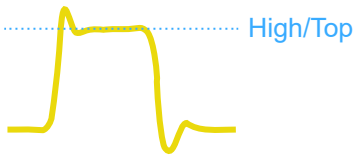
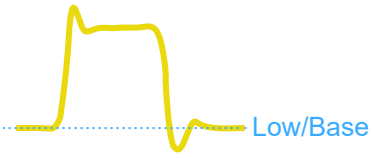
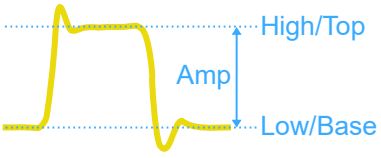
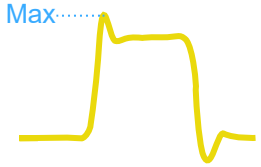
| Meas. type | Label | Description | Formula, graphic |
|---|---|---|---|
| Rise Time | RTime | Rise time of the first rising edge, the time it takes the signal to rise from the lower reference level to the upper reference level.<br>Measurement of all events in the acquisition is possible. |  |
| Fall Time | FTime | Fall time of the first falling edge, the time it takes the signal to fall from the upper reference level to the lower reference level.<br>Measurement of all events in the acquisition is possible. |  |
| Pos. pulse | PPuls | Duration of the first positive pulse: time between a rising edge and the following falling edge measured on the middle reference level.<br>Measurement of all events in the acquisition is possible. |  |

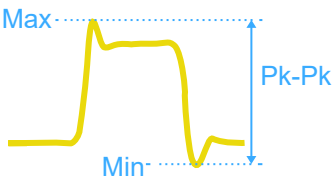
| Meas. type | Label | Description | Formula, graphic |
|---|---|---|---|
| Neg. pulse | NPuls | Duration of the first negative pulse: time between a falling edge and the following rising edge measured on the middle reference level.<br>Measurement of all events in the acquisition is possible. |  |
| Period | T | Time between the first two consecutive waveform edges of the same direction, measured on the middle reference level. The measurement requires at least one complete period of a triggered signal.<br>Measurement of all events in the acquisition is possible. |  |
| Frequency | Freq | Frequency of the signal, reciprocal value of the measured period. |  |
| Delay | Delay | Time difference between two slopes of the same or different waveforms, measured on the middle reference level.<br>A negative result indicates that the slope of the second source comes before the slope of the first source. |  |
| Phase | Phs<br>in ° | Phase difference between two waveforms, measured on the middle reference level. | *Phase = Delay / Period * 360*<br>  |
| Burst width | Bst | Duration of one burst. The burst is measured on the middle reference level from the first edge to the last edge, regardless of whether these edges rise or fall. | |
| Setup<br>Hold<br>Setup/Hold time | Setup<br>Hold<br>SHT | Setup and Hold time measurements with positive and/or negative clock edge, measured on the middle reference level.<br>See "Setup/Hold time measurement" on page 286. | |
| Setup/Hold ratio | SHR | Setup/Hold ratio measurement with positive and/or negative clock edge. | *SHR = Setup / (Setup + Hold)* |
| Delay to trigger | DTO | Time between the trigger point and the next signal slope, measured on the middle reference level. Slope direction can be selected. If the edge is to the left of the trigger (before trigger), the result is negative. | |

| Meas. type | Label | Description | Formula, graphic |
|---|---|---|---|
| Slew rate rising | SRRise<br>in V/s = V*Hz | Steepness of the rising edge: voltage difference between the lower and higher reference level, divided by the rise time. | *SRRise = ΔV / Δt*<br>  |
| Slew rate falling | SRFall<br>in V/s = V*Hz | Steepness of the falling edge: voltage difference between the higher and lower reference level, divided by the fall time. | *SRFall = ΔV / Δt*<br>  |
| Pos. duty cycle | PDCyc | Width of the first positive pulse in relation to the period in %. | $PDCyc = \frac{T_{PosPulse}}{T_{Period}} \times 100\%$ |
| Neg. duty cycle | NDCyc | Width of the first negative pulse in relation to the period in %. | $PDCyc = \frac{T_{NegPulse}}{T_{Period}} \times 100\%$ |

### Vertical measurements (amplitude)

The unit of most amplitude measurement results depends on the measured source.

| Meas. type | Label | Description | Formula, graphic |
|---|---|---|---|
| High | High | High level of the displayed waveform - the upper maximum of the sample distribution, or the mean value of the high level of a square wave without overshoot. The measurement requires at least one complete period of the signal.<br>Also known as top level. |  |
| Low | Low | Low level of the displayed waveform - the lower maximum of the sample distribution, or the mean value of the low level of a square wave without overshoot. The measurement requires at least one complete period of the signal.<br>Also known as base level. |  |
| Amplitude | Amp | Difference between the high level and the low level of the signal. The measurement requires at least one complete period of the signal. |  |
| Max | Max | Maximum value within the displayed waveform. |  |

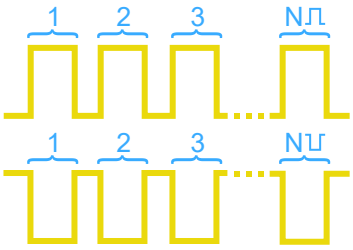
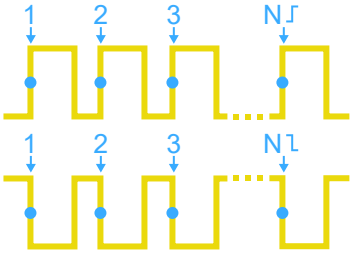
| Meas. type | Label | Description | Formula, graphic |
|---|---|---|---|
| Min | Min | Minimum value within the displayed waveform. |  |
| Peak to peak | PTP | Difference of maximum and minimum values. |  |
| Mean | Mean | Arithmetic average of the complete displayed waveform. | $Mean = \frac{1}{N}\sum_{k=1}^{N} x^k$ |
| RMS | RMS | RMS (root mean square) value of the voltage of the complete displayed waveform. | $RMS = \sqrt{\frac{1}{N}\sum_{k=1}^{N} {x_k}^2}$ |
| σ (S-dev/AC-RMS) | σ | Standard deviation of one cycle, usually of the first, left-most signal period. | $\sigma = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N} (x_k - Mean)^2}$ |
| Crest factor | Crest | The crest factor is also known as the peak-to-average ratio. It is the maximum value divided by the RMS value of the displayed waveform. | $Crest = \frac{Max\,\lvert x_k \rvert}{RMS}$ |
| Pos. Overshoot | P.OS | Overshoot of a square wave after a rising edge. It is calculated from measurement values High, local Max and Amplitude. | $OSpos = \frac{Max_{local} - High}{Amplitude} \times 100\%$ |
| Neg. Overshoot | N.OS | Overshoot of a square wave after a falling edge. It is calculated from measurement values Low, local Min and Amplitude. | $OSneg = \frac{Low - Min_{local}}{Amplitude} \times 100\%$ |
| Cycle Amp | CAmp | Amplitude of one cycle | |
| Cycle mean | CMean | Mean value of one cycle | |
| Cycle RMS | CRMS | RMS (root mean square) value of one cycle | |
| Cycle σ(S-dev) | Cycσ | Standard deviation of one cycle | |
| Cycle max | CMax | Maximum value of one cycle | |
| Cycle min | CMin | Minimum value of one cycle | |
| Cycle peak to peak | CPTP | Peak-to-peak value of one cycle: the difference of CMax and CMin | |
| Cycle crest | CCrest | Crest factor of one cycle | |

### Area and cycle measurements

All cycle measurements require that at least one complete period of the signal is acquired. Cycle measurements can be found in the "Amp/time" category.

| Meas. type | Label | Description/result | |
|---|---|---|---|
| Area | Area | Area between the waveform and ground.<br>$T_{Eval}$: evaluation time, time of a full waveform or limited by a gate | $Area = \frac{T_{Eval}}{N_{Eval}} \times \sum_{k=1}^{N_{Eval}} x(k)$ |
| Cycle area | CArea | Area between the waveform and ground, measured for one period, see also "Area" measurement. | |

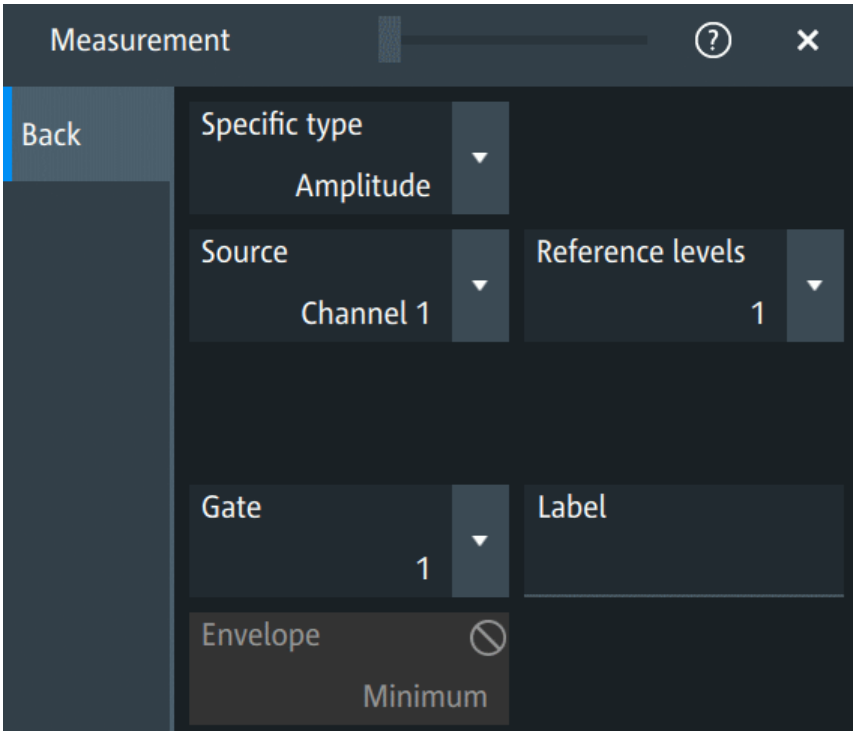
### Counting

| Meas. type | Label | Description/result | |
|---|---|---|---|
| Pulse count | PCount | The number of positive or negative pulses of the waveform, or of both positive and negative pulses.<br>The mean value of the signal is determined. If the signal passes the mean value, an edge is counted. A positive pulse is counted if a rising edge and a following falling edge are detected. A negative pulse is counted if a falling edge and a following rising edge are detected. |  |
| Edge count | EdCo | The number of positive or negative edges, or of both positive and negative edges.<br>The mean value of the signal is determined, and an edge is counted every time the signal passes the mean value. |  |

## 10.2.4.2 Settings for amplitude/time measurements

You can define additional parameters for some measurements.

Access: "Menu" > "Measurement" > "Setup" tab > add amplitude/time measurement > ⚙ settings of a measurement.




For a description of common measurement settings, see:

- "Clear results" on page 277
- "Specific type" on page 277
- "Source" on page 277
- "Reference levels" on page 277
- "Gate" on page 278
- "Label" on page 278
- "Display result lines" on page 278

**Envelope**
This setting is only available for measurements on envelope waveforms, see Acquisition mode.

| | |
|---|---|
| "Both" | The upper and the lower envelope are used in measurements. For time measurements, the averages of min and max values are used, that is, the measurement is performed on the average waveform built from the upper and lower envelope. |
| "Maximum" | Measurements are performed on the upper envelope. |
| "Minimum" | Measurements are performed on the lower envelope. |

Remote command:
`MEASurement<mg>:ENVSelect` on page 1156

**Pulse count**
Available, if "Specific type" is set to "Pulse train".

Sets the number N of positive pulses for the "Pulse train" measurement. This measurement measures the duration of N pulses, measured from the selected edge of the first pulse to the opposite edge of the N-th pulse.

Remote command:
`MEASurement<mg>:AMPTime:PTCount` on page 1157

**Pulse slope**
Available, if "Specific type" is set to "Pulse count" or "Pulse train".

Sets the first slope of the pulses to be counted.

"Positive" Positive pulses are counted.

"Negative" Negative pulses are counted.

"Either" Both positive and negative pulses are counted.

Remote command:
`MEASurement<mg>:AMPTime:PSLope` on page 1158

**Edges slope**
Available, if "Specific type" is set to "Edge count".

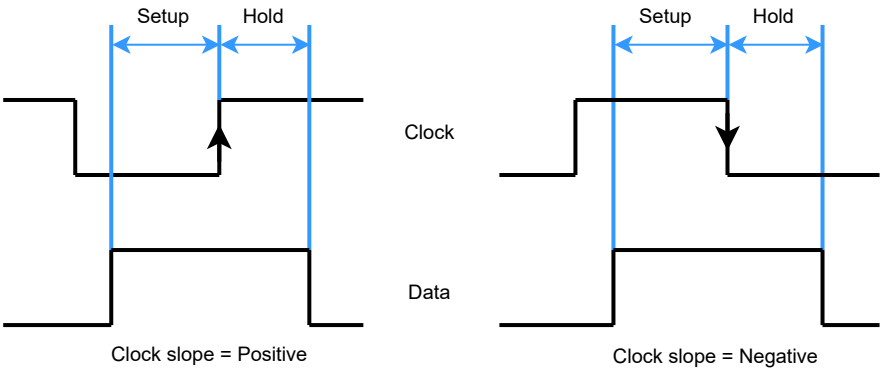
Sets the edge direction to be used for the selected measurement: positive, negative, or either edge.

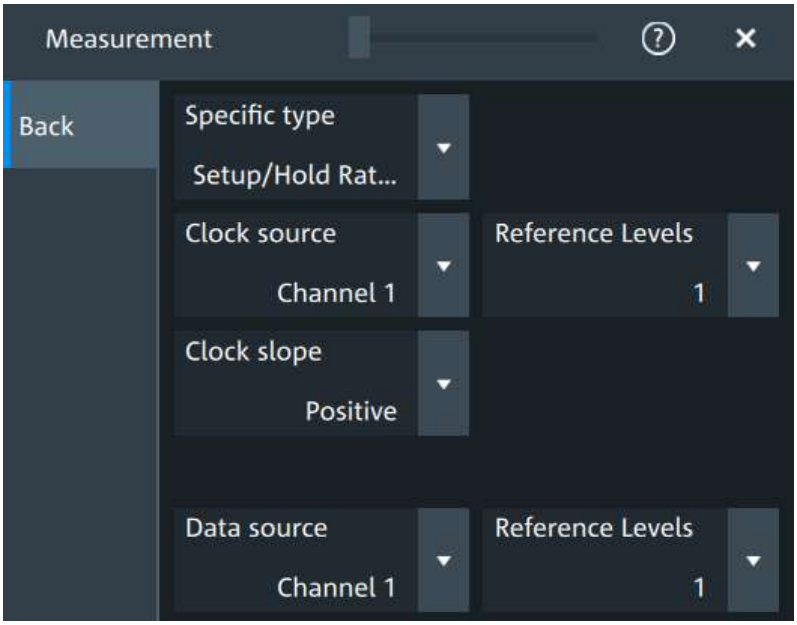
Remote command:
`MEASurement<mg>:AMPTime:ESLope` on page 1157

**Setup/Hold time measurement**
Setup/Hold measurements analyze the relative timing between two signals: a data signal and the synchronous clock signal. Setup time is the time that the data signal is steady before the clock edge. Hold time is the time that the data signal is steady after the clock edge.

"Setup/Hold time" measures and displays the setup and hold durations. "Setup/Hold ratio" measurements return the ratio of the setup time to the sum of hold and setup time: $T_{Setup} / (T_{Setup} + T_{Hold})$.

The clock edge can be defined, the polarity of the data signal does not matter.






**Source 2, Clock source ← Setup/Hold time measurement**
Sets the waveform used as the clock for the setup/hold measurement, or the second source waveform for delay and phase measurements.

Remote command:
`MEASurement<mg>:SSRC` on page 1154

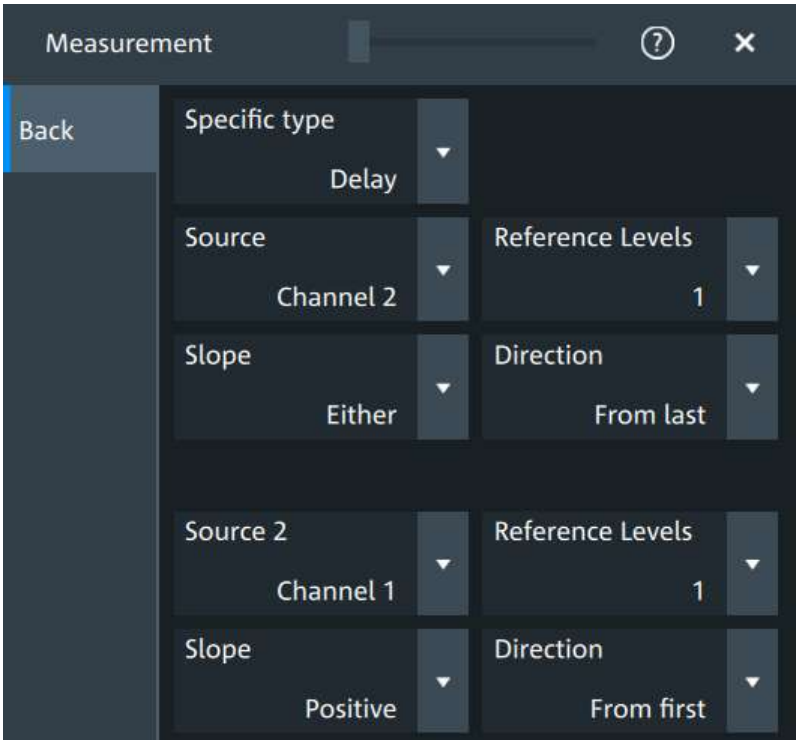
**Clock slope ← Setup/Hold time measurement**
Sets the edge of the clock from which the setup and hold times are measured: positive, negative, or either of them. If "Either" is selected, the clock edges next to the data edge are considered regardless of the clock slope.

Remote command:
`MEASurement<mg>:AMPTime:CSLope` on page 1157

**Data source ← Setup/Hold time measurement**
Sets the source for the data signal. See also: "Source" on page 277.

**Delay and phase measurement**
The specific settings for delay measurement allow you to measure the time between any two slopes at any reference level. Therefore, the reference levels and the slopes must be defined for each source individually. The measurement result is negative if the edge of the second source comes before the edge of the first source.

To measure the phase difference between two waveforms, you also set the slope, direction and reference level for each source individually. You can also specify the phase unit.




**Slope ← Delay and phase measurement**
Available, if "Specific type" is set to "Phase" or "Delay".

Sets the edge direction to be used for the selected measurement: positive, negative, or either edge.

Remote command:
`MEASurement<mg>:AMPTime:DELay<n>:SLOPe` on page 1159

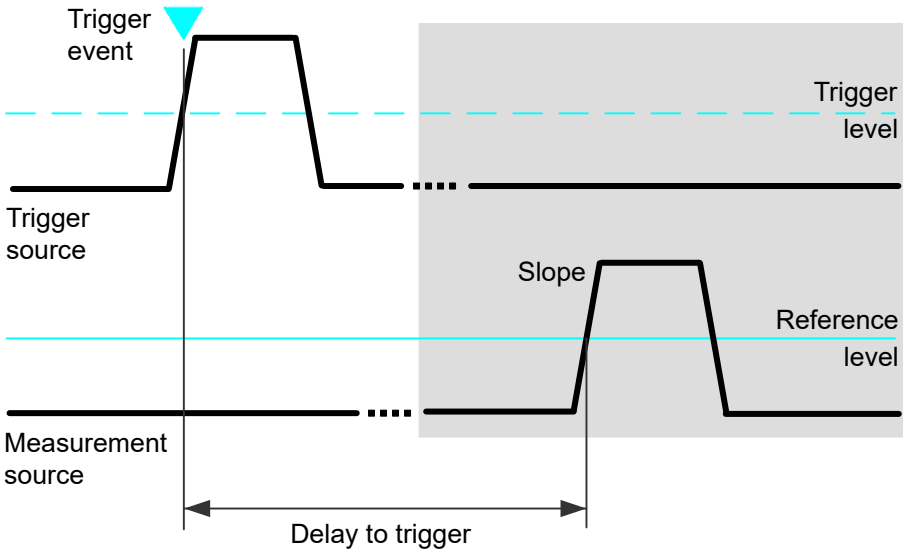
**Direction ← Delay and phase measurement**
Selects the direction for counting slopes for each source: from the beginning of the waveform, or from the end.

Remote command:
`MEASurement<mg>:AMPTime:DELay<n>:DIRection` on page 1158

**Phase unit ← Delay and phase measurement**
Selects the unit of the oase: degrees or radians.

Remote command:
`MEASurement<mg>:AMPTime:PHASe:UNIT` on page 1159

**Delay to trigger measurement settings**
Delay to trigger measures the time between the trigger point and the following slope of a waveform. The delay between the trigger and the slope can be high compared to the accuracy of the acquisition, and the trigger point can even be outside of the current acquisition.




To configure the trigger conditions, use the trigger setup.

"Edges slope" Sets the edge direction to be used for delay measurement: positive, negative, or either edge.

Remote command:
`MEASurement<mg>:AMPTime:DTOTrigger<n>:SLOPe` on page 1158

**Unit (slew rate measurements)**
Sets the denominator of the unit for slew rate measurements.

## 10.2.5 Histogram measurements

You can perform measurements on waveform histograms.

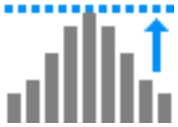
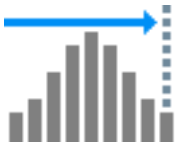
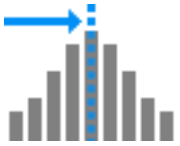
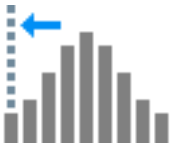
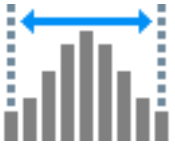
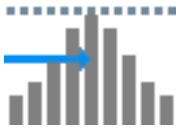
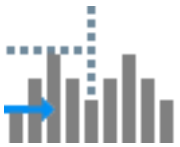
Note that graphical changes on the display, e.g. changing the diagram size or the size of the result table, reset the histogram measurements in run mode and also in stop mode.

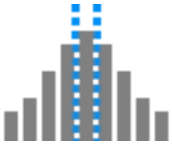
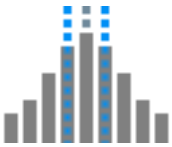
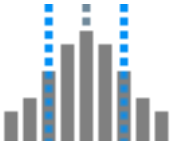
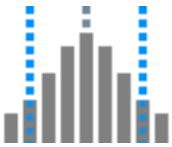
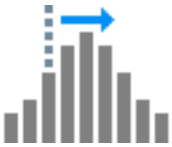
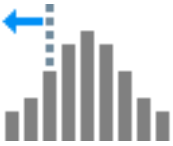
### 10.2.5.1 Overview of measurements on histograms

The specific measurement types for waveform histograms are provided in the "Histogram" category. To see these measurement types, an active waveform histogram is required.

To select and activate measurements in remote control, use:

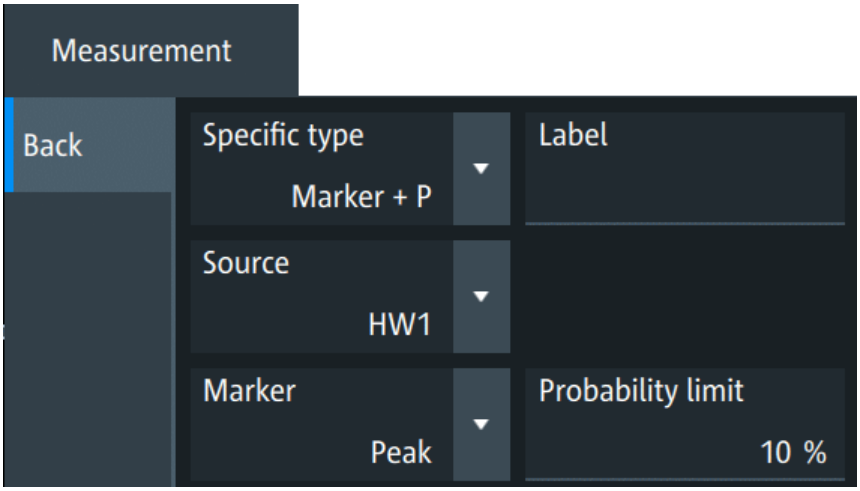
- `MEASurement<mg>:MAIN` on page 1154
- `MEASurement<mg>[:ENABle]` on page 1152
- `MEASurement<mg>:ACTive` on page 1152

| Meas. type | Label | Description, result | Graphic |
|---|---|---|---|
| Histogram peak | HPeak | The maximum sample value in the histogram. |  |
| Histogram samples | HSamp | The number of samples from all acquisitions included in the current histogram. | |
| Maximum | HMax | The highest signal value with a probability > 0. |  |
| Mean | HMean | The weighted arithmetic average of the histogram. |  |
| Minimum | HMin | The lowest signal value with a probability > 0. |  |
| Max - Min | MxMn | The range of signal values with a probability > 0. |  |
| Median | Med | The signal value for which half the samples lie above, the other half below in the histogram.<br>The sample count of one signal value after the other are accumulated until half the total number of samples in the histogram is reached. The signal value for which 50% of the samples are accumulated is the median. | |
| Peak value | Peak | The signal value at the histogram peak. |  |
| Lower peak value | LPeak | The signal value at the maximum sample value in the lower half of the histogram. |  |
| Upper peak value | UPeak | The signal value at the maximum sample value in the upper half of the histogram. |  |

| Meas. type | Label | Description, result | Graphic |
|---|---|---|---|
| σ(S-dev) | H σ | Standard deviation of the sample numbers. |  |
| Mean ± σ | Mn±σ | Percentage of samples in the range between (mean value + standard deviation) and (mean value - standard deviation). |  |
| Mean ± 2 * σ | Mn±2*σ | Percentage of samples in the range between (mean value + 2 * standard deviation) and (mean value - 2 * standard deviation) |  |
| Mean ± 3 * σ | Mn ±3*σ | Percentage of samples in the range between (mean value + 3* standard deviation) and (mean value - 3* standard deviation) |  |
| Marker - P | Mkr-% | The marker value (according to the selected probability domain marker type) minus the defined limit. The value is restricted to the histogram range. |  |
| Marker + P | Mkr+% | The marker value (according to the selected probability domain marker type) plus the defined limit. The value is restricted to the histogram range. |  |
| Waveform count | WCount | The number of waveforms the histogram is based on. | |

### 10.2.5.2 Settings for measurements on histograms

Access: "Menu" > "Measurement" > "Setup" tab > add histogram measurement > ⚙ settings of a measurement.




For most measurements on histograms, only common settings are required.

The measurement types "Marker + P" and "Marker - P" require additional settings.

For a description of common measurement settings, see:

- "Specific type" on page 277
- "Clear results" on page 277
- "Source" on page 277
- "Label" on page 278

**Marker**
Defines the marker position in the probability domain.

The setting is relevant for measurements "Marker + P" and "Marker - P".

| | |
|---|---|
| "Peak" | The y-value with the maximum sample value in the histogram |
| "Upper Peak" | The y-value at the maximum sample value in the upper half of the histogram |
| "Lower Peak" | The y-value at the maximum sample value in the lower half of the histogram |
| "Maximum" | The highest y-value with a probability > 0 |
| "Minimum" | The lowest y-value with a probability > 0 |
| "Median" | The y-value for which half the samples lie above, the other half below in the histogram |
| "Mean" | The weighted arithmetic average of the histogram |

Remote command:
`MEASurement<mg>:HISTogram:PROBability:TYPE` on page 1160

**Probability limit**
Defines a range around the marker.

The setting is relevant for measurements "Marker + P" and "Marker - P".

Remote command:
`MEASurement<mg>:HISTogram:PROBability:LIMit` on page 1160

## 10.2.6 Basic jitter analysis (option R&S MXO3-K12)

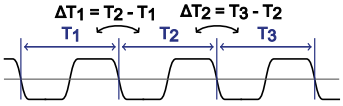
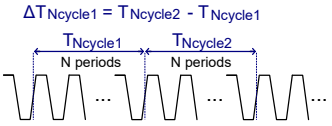
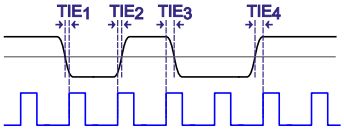
### 10.2.6.1 Overview of basic jitter measurements

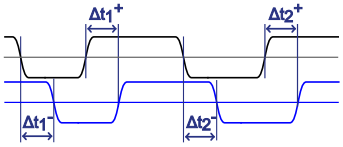
The specific measurements for basic jitter analysis are provided in the "Jitter" category.

To select and activate measurements in remote control, use:

- `MEASurement<mg>:MAIN` on page 1154
- `MEASurement<mg>[:ENABle]` on page 1152
- `MEASurement<mg>:ACTive` on page 1152

***Table 10-1: Jitter measurements***

| Meas. type | Label | Description | Formula, graphic |
|---|---|---|---|
| Cycle-cycle duty cycle | CCDCyc | Difference between the duty cycle of two adjacent cycles. The measurement is based on the duty cycle measurement. You can select the pulse polarity for the duty cycle measurement. | $\Delta R_{Cyc\ k} = R_{Cyc\ k+1} - R_{Cyc\ k}$ for $k = 1,...,K-1$ |
| Cycle-cycle width | CCWid | Difference between the pulse width of two adjacent cycles. The measurement is based on the pulse width measurement. You can select the pulse polarity to be measured. | $\Delta T_{Pulse\ k} = T_{Pulse\ k+1} - T_{Pulse\ k}$ for $k = 1,...,K-1$ |
| Cycle-cycle jitter | CCJit | Difference between the periods of two adjacent cycles. The measurement is based on the period measurement. You can select the slope and the reference level on which the period is measured. | $\Delta T_{Period\ k} = T_{Period\ k+1} - T_{Period\ k}$ for $k = 1,...,K-1$   |
| N-cycle jitter | NCJit | Difference between the time of two adjacent groups of N cycles (periods) each. You can select the slope and the reference level on which the time is measured, and the number of periods in the groups. | $\Delta T_{Ncycle\ k} = T_{Ncycle\ k+1} - T_{Ncycle\ k}$ for $k$ = *cycle group index*   |
| Time interval error | TIE | Time difference between the slope of the input signal and the slope of a reference signal. The reference signal can be a captured clock waveform, or a clock generated by clock data recovery (CDR, software algorithm or hardware generation). You can select the slope and the reference level on which the TIE is measured. | $TIE_k = t_{Signal\ k} - t_{Clock\ k}$ for $k = 1,...,K$   |
| Unit interval | UI | Period of the clock signal. If no clock signal is available, it is recovered by CDR. The period is calculated as the time difference between two consecutive clock edges of the same polarity. | $UI_k = t_{Clock\ k+1} - t_{Clock\ k}$ for $k = 1,...,K-1$ |
| Data rate | DR | Frequency of the clock signal. If no clock signal is available, it is recovered by CDR. The measurement is based on the unit interval measurement. | $R_{Clock\ k} = 1 / UI_k$ for $k = 1,...,K-1$ |

| Meas. type | Label | Description | Formula, graphic |
|---|---|---|---|
| Skew delay | SkwD | Delay between the edges of two interdependent waveforms. The measurement is a simplified variant of the "Delay" measurement assuming that both sources are similar except for the delay. | *Skew delay* = $\Delta t_k = t_{Source2} - t_{Source1}$ for $k = 1,...,K$   |
| Skew phase | SkwP | Phase difference between the edges of two waveforms. | *Skew phase = Skew delay / Period * 360°* = $\Delta t_k / \Delta T_{Period\ k} * 360°$ |

### 10.2.6.2 Settings for jitter measurements

Access: "Menu" > "Measurement" > "Setup" tab > add jitter measurement > ⚙ settings of a measurement.

For a description of common measurement settings, see:

- "Specific type" on page 277
- "Source" on page 277
- "Reference levels" on page 277
- "Gate" on page 278
- "Label" on page 278

#### Settings for cycle-cycle measurement

The cycle-cycle measurements are used to analyze the quality of clock signals. They require a few additional settings for period and pulse width measurement.

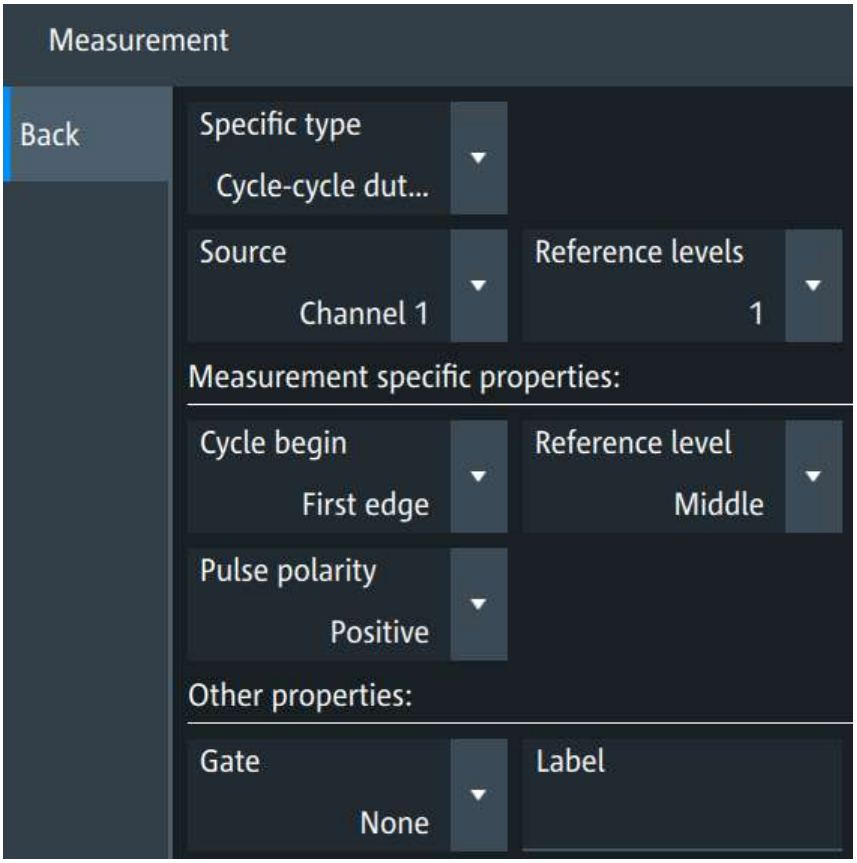



*Figure 10-1: Setup for cycle-cycle duty cycle measurement*

**Cycle begin, Slope**
For cycle-cycle jitter, N-cycle jitter, and cycle-cycle duty cycle measurements, "Cycle begin" selects the slope at which the periods and thus the jitter is measured: first edge, positive, negative or either.

For time-interval error measurements, "Slope" sets the edges of the data signal that are used for measurements: positive, negative or either.

"First edge" Measures the period from the first edge that is found, no matter of its direction.

"Positive" Measures the period at positive going edges.

"Negative" Measures the period at negative going edges.

"Either" Measures the period at both positive and negative going edges. This option is useful, for example, to check the clock stability of a double data rate clock.

Remote command:
`MEASurement<mg>:JITTer:SLOPe` on page 1163

**Pulse polarity**
For cycle-cycle width and the cycle-cycle duty cycle measurements, "Pulse polarity" sets the polarity of pulses for which the pulse width is measured: positive or negative.

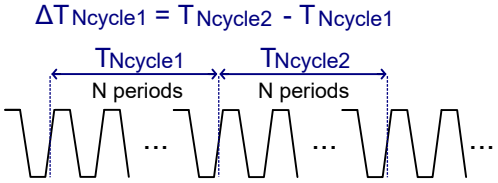
For skew delay and skew phase measurements, "Edge polarity" sets the edge of the first waveform from which the measurements starts: positive, negative or either.

Remote command:
MEASurement<mg>:JITTer:POLarity on page 1162

**Reference level**
Selects the reference level that is used for time measurements.

Remote command:
MEASurement<mg>:JITTer:REFLevel<rl> on page 1162

**Number of cycles**
Sets the number of periods (cycles) that are accumulated to measure the N-cycle jitter.




Remote command:
MEASurement<mg>:JITTer:NCYCles on page 1161

### Settings for delay measurements

Skew delay and skew phase measurements are intended to measure the time difference between the edges of two waveforms. The measurements are simplified variants of the "Delay" and "Phase" measurements assuming that both sources are similar except for the delay.

**Source 1, Source 2**
"Source 1" is the reference signal, and "Source 2" is the signal compared to the reference signal.

*Skew delay* = $\Delta t_k = t_{Source2} - t_{Source1}$ for $k = 1,...,K$

See also: "Clock slope" on page 287.

Remote command:
MEASurement<mg>:SOURce on page 1152

**Edge polarity**
See "Pulse polarity" on page 295.

**Relative polarity**
Sets the edge of the second waveform relative to the first waveform.

"Matching" Measures from positive to positive edge or from negative to negative edge.

"Inverse" Measures from positive to negative edge or from negative to positive edge.

Remote command:
MEASurement<mg>:JITTer:RELPolarity on page 1163

**Reference level**
Selects the reference level that is used for time measurements on "Source 2".

Remote command:
`MEASurement<mg>:JITTer:REFLevel<rl>` on page 1162

### Settings for data measurements

The following measurements are intended to analyze serial data:

- Time interval error
- Unit interval
- Data rate

The clock can be a captured clock signal, or it can be recovered from the data signal. The clock or CDR is configured in the "Timing reference" dialog, see Section 8.3, "Timing reference", on page 215.

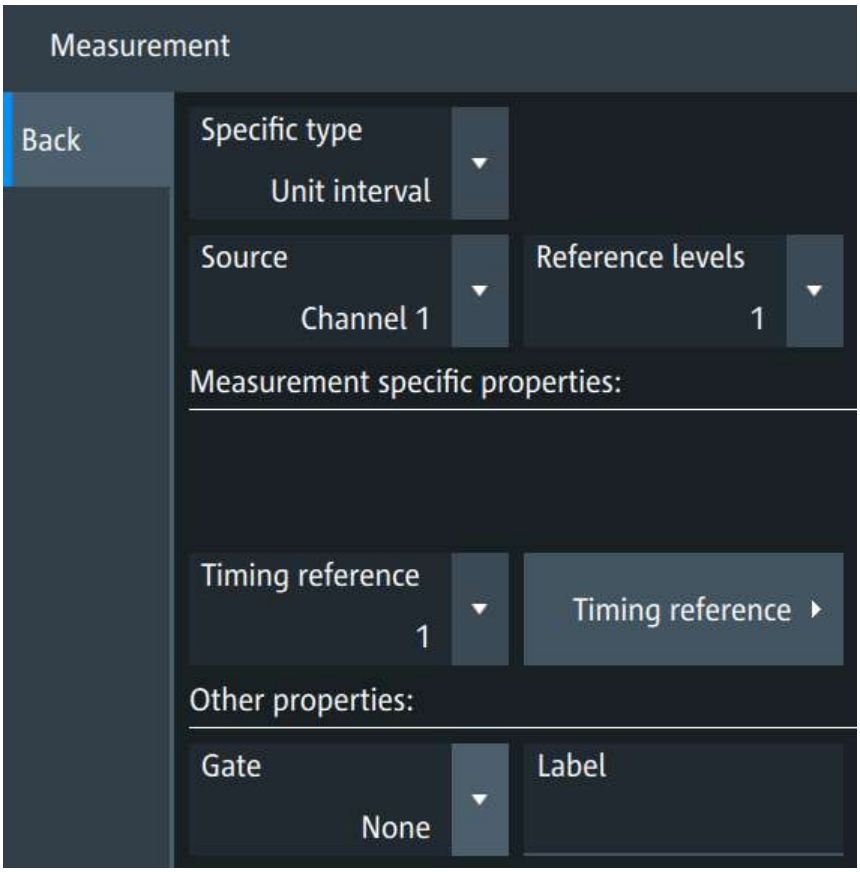



*Figure 10-2: Setup for unit interval measurement*

**Slope**
Sets the edges of the data signal that are used for measurements. The setting is relevant for time-interval error measurements.

See also: "Cycle begin, Slope" on page 295.

**Timing reference**
Selects the timing reference, which is one of the available clock configurations.

For configuration of the timing reference, see Section 8.3, "Timing reference", on page 215.

Remote command:
MEASurement<mg>:JITTer:TREF on page 1164

**Unit**
Sets the unit to bps (bit per second), or to Hz for analysis of radar signals. The setting is available for data rate measurements.

Remote command:
MEASurement<mg>:JITTer:UNIT on page 1164

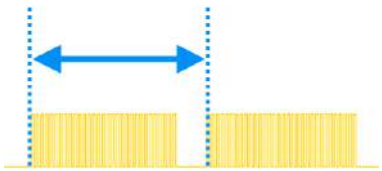
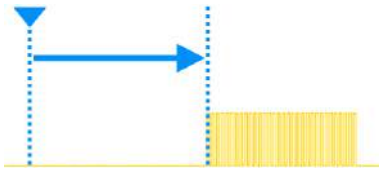
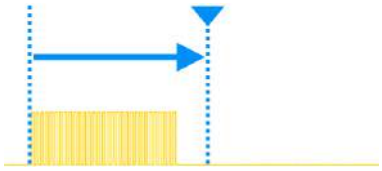
## 10.2.7 Protocol measurements (option R&S MXO3-K500)

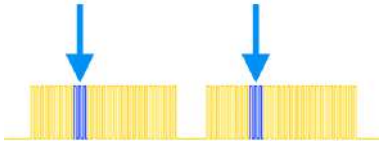
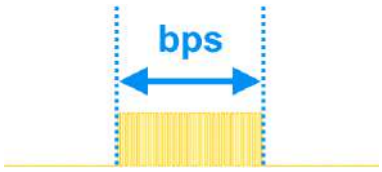
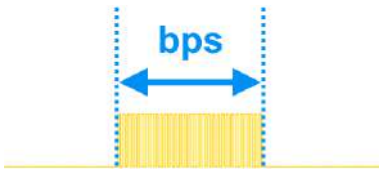
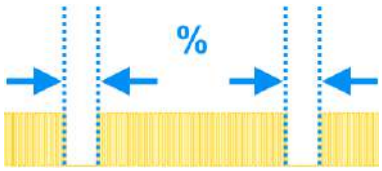
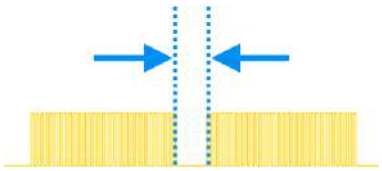
### 10.2.7.1 Overview of protocol measurements

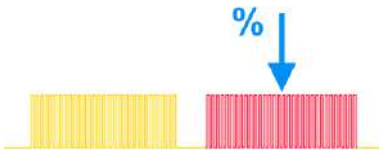
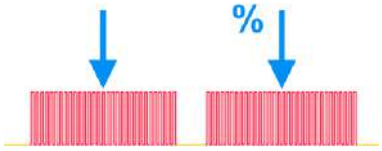
The specific measurement types for protocol measurements are provided in the "Protocol" category. To see these measurement types, an active serial bus is required.

To select and activate measurements in remote control, use:

- MEASurement<mg>:MAIN on page 1154
- MEASurement<mg>[:ENABle] on page 1152
- MEASurement<mg>:ACTive on page 1152

| Meas. type | Label | Description/result | |
|---|---|---|---|
| Frame/frame | F2F | The time from the start of one frame to the start of the next frame. |  |
| Trig/frame | T2F | The time from a trigger event to the start of the next frame. |  |
| Frame/trig | F2T | The time from the start of a frame to the next trigger event. |  |

| Meas. type | Label | Description/result | |
|---|---|---|---|
| Field value | FVal | The value of a selected field in a frame. |  |
| Main bit rate | BPS | The nominal bits per second (bps) in a selected frame. The measurement ignores, for example, gaps in a LIN frame or a bus park bit in an RFFE or SPMI frame. |  |
| 2nd bit rate | BPS2 | Like main bit rate, but optional for protocols with multiple bit rates, e.g. CAN/CAN-XL, SPMI or RFFE write / read. |  |
| Bus idle | Idle | Percentage of idle times between all frames in an acquisition. |  |
| Gap | Gap | The gap time (not necessarily idle) between two frames. |  |
| Frame count | FCnt | The number of complete frames in an acquisition. |  |
| Frame errors | FECnt | The number of erroneous frames in an acquisition. |  |

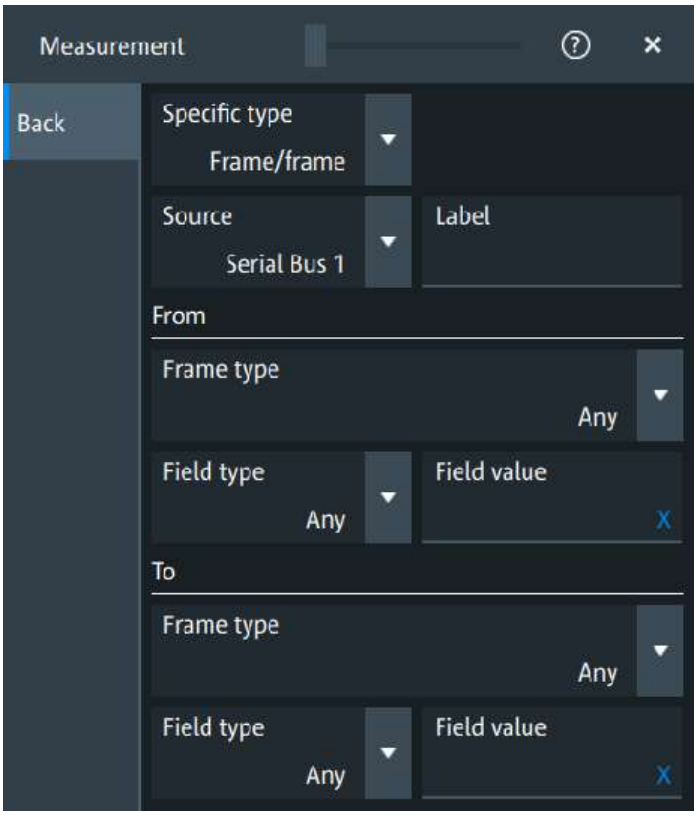
| Meas. type | Label | Description/result | |
|---|---|---|---|
| FER | FER | The frame error rate, hence the percentage of erroneous frames in an acquisition. |  |
| Cons FER | CFER | The frame error rate (percentage) of two or more consecutive erroneous frames. |  |

### 10.2.7.2 Settings for protocol measurements

You can define additional parameters for some protocol measurements.

Access: "Menu" > "Measurement" > "Setup" tab > add protocol measurement > ⚙ settings of a measurement.

Available only, if "Source" = "Serial Bus N".




For a description of common measurement settings, see:

- "Specific type" on page 277
- "Clear results" on page 277

- "Source" on page 277
- "Label" on page 278

**Frame Identification or From**
"Frame Identification" is available only for the measurement types "Frame/trig", "Trig/frame" and "Field value". "Frame Identification" specifies where the measurement is executed.

"From" is available only for the measurement type "Frame/frame", which requires a "From" - "To" condition. "From" specifies where the measurement starts.

All other measurement types do not require these settings.

**Frame Type ← Frame Identification or From**
In a "Frame Identification", this setting specifies the type of the frame at which the measurement is executed.

In a "From" - "To" condition, this setting specifies the type of the frame at which the measurement starts.

Remote command:
`MEASurement<mg>:PROTocol:FNAMe` on page 1164

**Field Type ← Frame Identification or From**
In a "Frame Identification", this setting specifies the type of the field at which the measurement is executed.

In a "From" - "To" condition, this setting specifies the type of the field at which the measurement starts.

Remote command:
`MEASurement<mg>:PROTocol:FDName` on page 1165

**Field Value ← Frame Identification or From**
In a "Frame Identification", this setting specifies the value of the field at which the measurement is executed.

In a "From" - "To" condition, this setting specifies the value of the field at which the measurement starts.

Remote command:
`MEASurement<mg>:PROTocol:FDValue` on page 1165

**To**
"To" is available only for the measurement type "Frame/frame", which requires a "From" - "To" condition. "To" specifies where the measurement ends.

**Frame Type ← To**
Specifies the type of the frame at which the "Frame/frame" measurement ends.

Remote command:
`MEASurement<mg>:PROTocol:F2Name` on page 1165

**Field Type ← To**
Specifies the type of the field at which the "Frame/frame" measurement ends.

Remote command:
MEASurement<mg>:PROTocol:FD2Name on page 1166 (used also for Tracked)

**Field Value ← To**
Specifies the value of the field at which the "Frame/frame" measurement ends.

Remote command:
MEASurement<mg>:PROTocol:FD2Value on page 1166

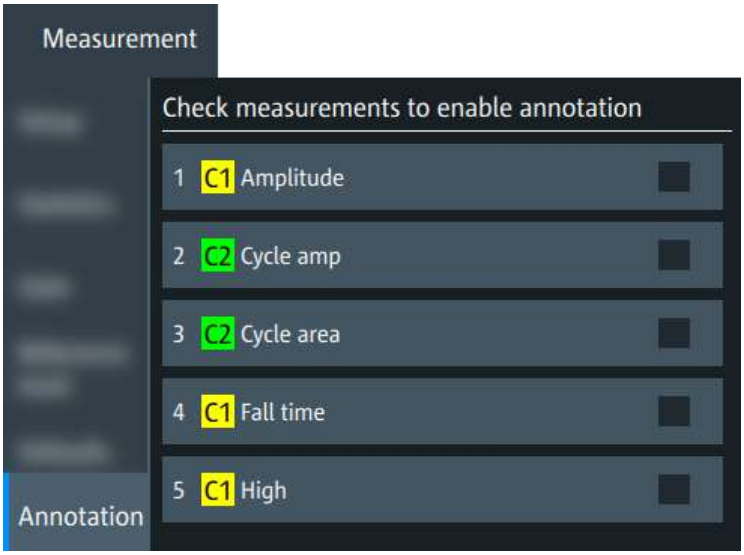
**Tracked**
Available only for the measurement type "Field value". This setting selects the field type to be displayed.

Remote command:
MEASurement<mg>:PROTocol:FD2Name, the same command is used for Field Type.

## 10.2.8 Annotation for measurements

Measurement annotations mark the significant places in the diagram, for example, periods, maximum and minimum values, relevant reference levels, and more. The lines help understand the measurement results.

In the "Annotation" tab, you can select the measurements to be annotated. Alternatively, you can enable "Display result lines" in the "Setup" of a selected measurement. See "Display result lines" on page 278.




The "Check measurements to enable annotations" list shows all active measurements. The list is empty if no measurement is active. Configure and activate the measurements first, then select them for annotation.

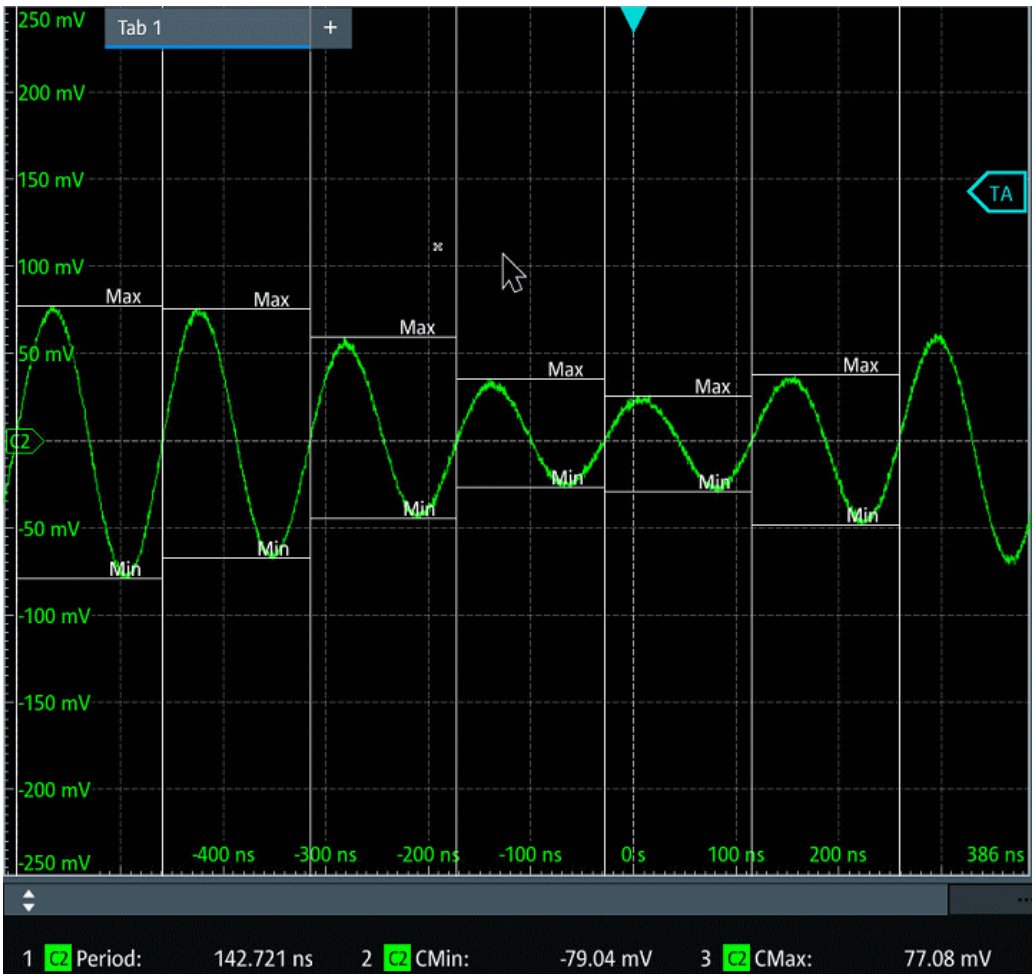



*Figure 10-3: Example of annotations for cycle measurements*

## 10.2.9 Measurement results

The measurement results are shown in a table below the grid.




Dashes (---) indicate that no valid waveform is available, for example, if the source waveform is off.

For further analysis of measurement results, the following evaluations are available:

- Section 10.2.10.1, "Statistics", on page 304
- Section 10.2.10.2, "Track", on page 306

You can save the measurement results to file, see Section 13.6, "Results", on page 475.

## 10.2.10 Analysis of measurement results

The behavior of measurement results over time can be evaluated in different ways, for example, with statistics and tracks.


- Statistics....304
- Track....306
- Histograms on measurement results....310


### 10.2.10.1 Statistics

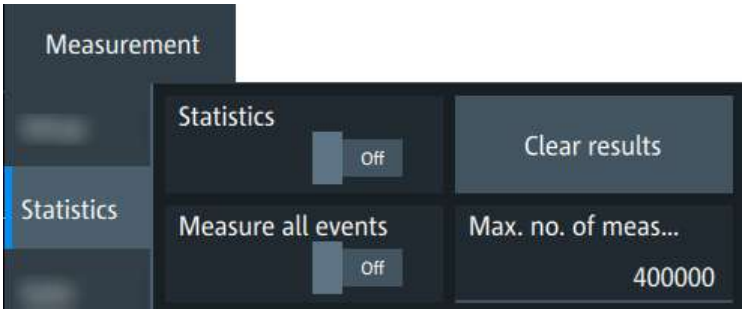
In addition to the current measurement results, you can enable a statistical evaluation. It returns the current, minimum and maximum measurement values, the average and standard deviation, and the number of measured waveforms. The results are shown in a separate tab below the grid. If the cursor measurement is active simultaneously, its results are shown beside the statistics results.

If statistics are enabled, the following results are calculated. .

| Label | Description |
|---|---|
| Max | Positive peak value (maximum) |
| Min | Negative peak value (minimum) |
| Mean | Arithmetic mean (average) |
| RMS | Root mean square |
| σ (S-dev) | Standard deviation |
| Rel. σ | Relative standard deviation (*S-dev / Mean*) |
| Event count | Number of measured events (e.g. rising edges, pulses etc.) |
| Wave count | Number of waveforms (acquisitions) the measurement is based on |




Access: "Menu" > "Measurement" > "Statistics" tab.




**Statistics**
Activates or deactivates the statistical evaluation for all measurements.

Remote command:
`MEASurement<mg>:STATistics[:ENABle]` on page 1169

**Clear results**
Deletes the statistical results for all measurements including the histogram, and starts a new statistical evaluation.

Remote command:
`MEASurement<mg>:STATistics:ARESet` on page 1169
To delete results of individual measurements, use `MEASurement<mg>:CLEar` on page 1155 or `MEASurement<mg>:STATistics:RESet` on page 1155.

**Measure all events**
Normally, only one measurement is performed for each acquired waveform to get best performance. If "Measure all events" is enabled, more than one result is taken from one acquired waveform and the results are included in evaluation. For example, the rise time is measured on all pulses in the waveform, not only on the first.

The result table shows only the first result of the waveform, the following results are used only for evaluation. The number of considered results can be restricted: see Max. No. of events.

Measuring all events is useful when calculating statistics and histograms. It can also be useful when generating tracks; however, it reduces the performance of the instrument.

The setting affects all measurements, it is a global setting. However, if a track is enabled for a selected measurement, all results are considered for this measurement, independent from the global setting.

Remote command:
`MEASurement<mg>:MULTiple` on page 1170

**Max. No. of events**
Sets the maximum number of measured events per acquisition.

Remote command:
`MEASurement<mg>:MNOMeas` on page 1170

### 10.2.10.2 Track

A track shows the measurement results over time for an acquired waveform. The track is the graphical interpretation of all measurement values of a single acquisition. Furthermore, the track reveals trends of change in the analysis and preserves the timing relationship of the measurement results to the signal.

Tracks can be enabled for each active measurement. Enabling the track also enables "Measure all events" for the selected measurement.

To analyze tracks, you can use cursor measurements, zoom, mathematics and spectrum.

#### Track activation

Access: "Menu" > "Measurement" > "Track" tab.

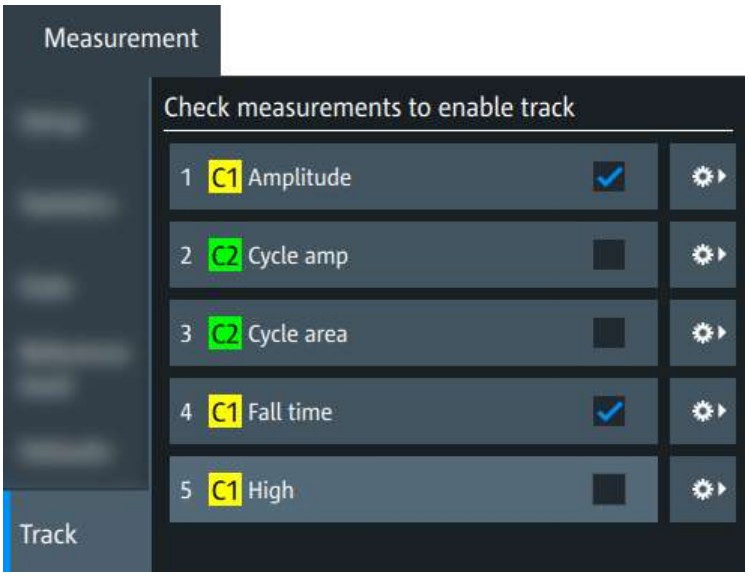



*Figure 10-4: Measurement selection for tracks*

**Check measurements to enable track**
The list shows all active measurements. The list is empty if no measurement is active. Enable the measurements first, then select them for track analysis.

Select the checkbox of a measurement to enable the track for this measurement.

Tap the settings icon to open the track settings.

#### Track settings

Access: "Menu" > "Measurement" > "Track" tab > tap measurement button.

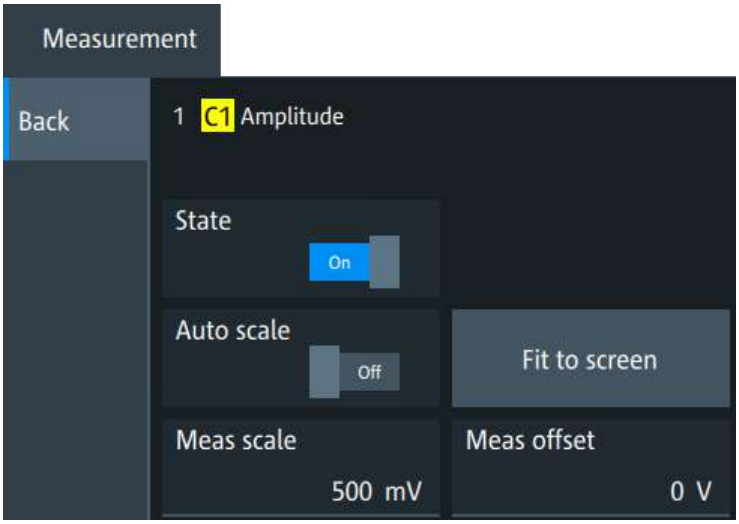



*Figure 10-5: Track settings*

**State**
Enables or disables the track for the selected measurement.

Enabling the track also enables "Measure all events" for the selected measurement.

Remote command:
`MEASurement<mg>:TRACk[:STATe]` on page 1170

**Auto scale**
Performs an automatic scaling whenever the track does not fit in the diagram during the measurement period.

Remote command:
`MEASurement<mg>:TRACk:CONTinuous` on page 1171

**Fit to screen**
Sets the vertical scale and the offset of the track, so that the complete height of the diagram is used.

Remote command:
`MEASurement<mg>:TRACk:AUTO` on page 1171

**Meas scale**
Defines the vertical scale of the track diagram. If "Auto scale" is on, the current scale value is displayed.

You can also select the track waveform and use the vertical [Scale] knob to change the scale.

Remote command:
`MEASurement<mg>:TRACk:SCALe` on page 1172

**Meas offset**
Defines the offset of the track waveform. If "Auto scale" is on, the current offset value is displayed.

You can also select the track waveform and use the vertical [Position] knob to change the offset.

Remote command:
`MEASurement<mg>:TRACk:OFFSet` on page 1171

### Examples of track usage

The examples in this section provide ideas on how you can use tracks for waveform analysis.

#### Tracks with math functions




*Figure 10-6: Tracks used in math functions*

The example is set up as follows:

- Waveform on channel 1
- 3 measurements on C1: positive duty cycle, period and positive pulse width
- Enable track for each measurement: Track 1 (positive duty cycle), Track 2 (period), Track 3 (positive pulse width)
- Math1 = Trk1 * Trk2
- Math2 = Math1 / 100
  Expected values: Period * Pos. duty cycle / 100 = Positive pulse width, i.e. Math2 = Trk3
- Math3 = Math2 - Trk3
  Expected values = 0

### Spectrum on tracks




*Figure 10-7: Track and spectrum on this track. C1 is not in the diagram.*

The example is set up as follows:

- Waveform on channel 1
- Period measurement on C1
- Enable track for the period measurement: Track 1
- Spectrum with "Source" = Track 1
- Enable "Peak list" > "Peak on waveform"

### Zoom on tracks

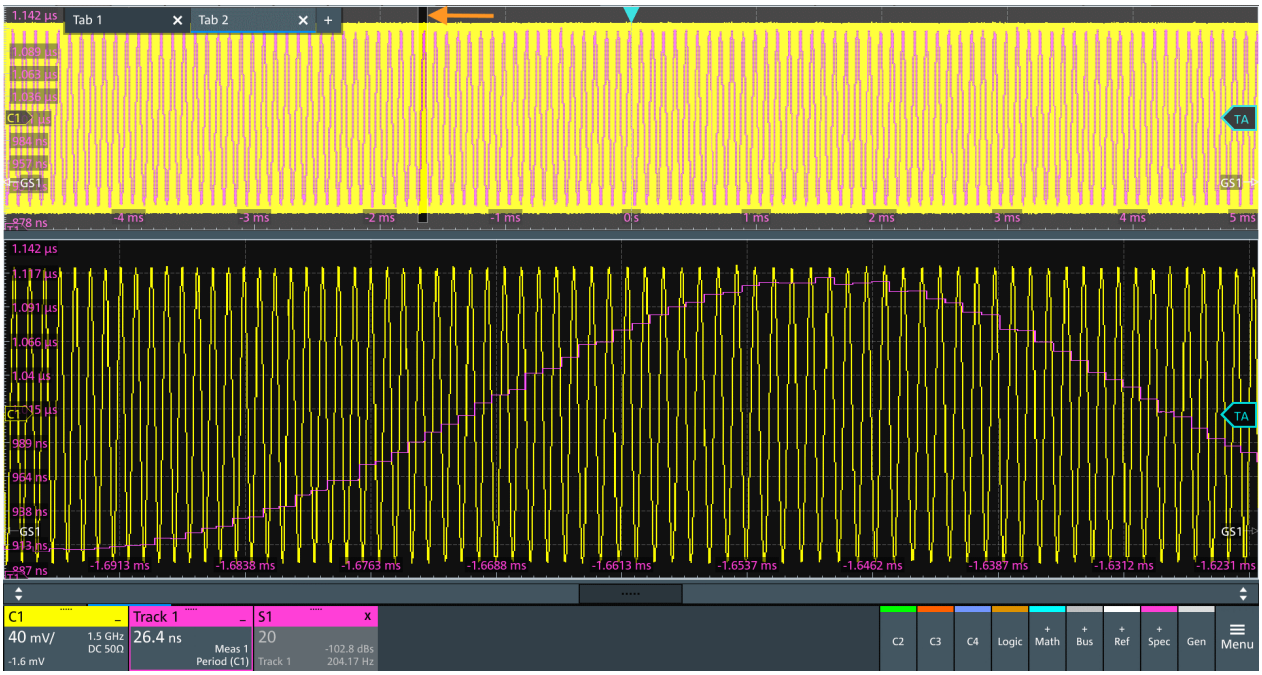



*Figure 10-8: Track and C1 in the upper diagram, zoom below*

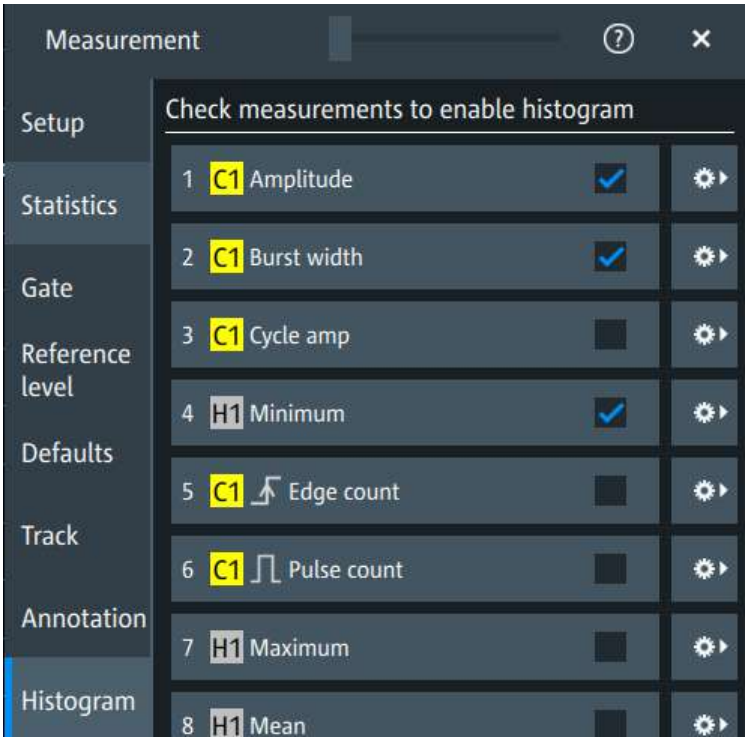
The example is set up as follows:

- Waveform on channel 1
- Period measurement on C1
- Enable track for the period measurement: Track 1.
- Drag and drop the track on the channel waveform.
- Create zoom.

## 10.2.10.3 Histograms on measurement results

Access: "Menu" > "Measurement" > "Histogram" tab.

Histograms are available not only for channel, math and reference waveforms, but also for measurement results. A histogram on measurement results is the cumulative occurrence distribution of measurement results in a graphic. Each histogram is shown in a separate diagram; one histogram per measurement can be created. The source is a measurement, and all events in all acquisitions are measured and included in the histogram.

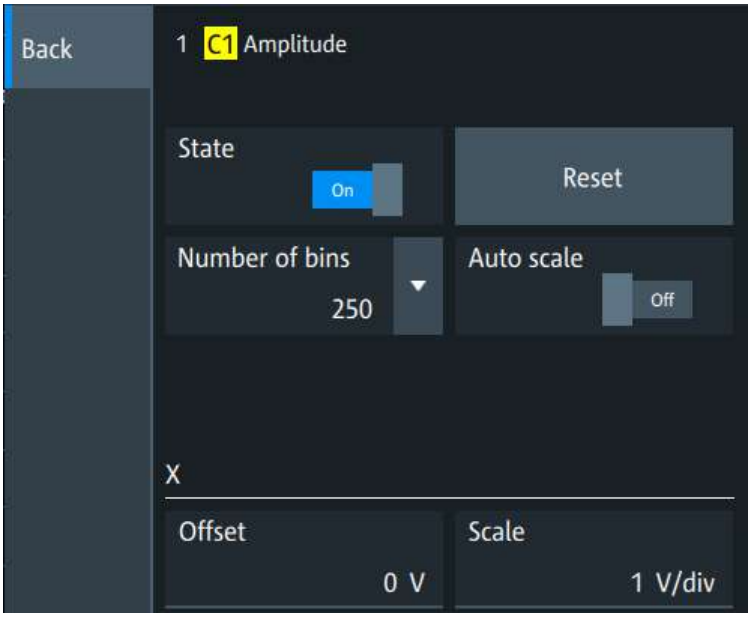



The list shows all active measurements. The list is empty if no measurement is active. Enable and configure the measurements first, then select them for histogram analysis.

1. Select the checkbox of a measurement to enable the histogram for this measurement.
2. Tap the settings icon to open the histogram settings for the selected measurement.

### Histogram settings

Access: "Menu" > "Measurement" > "Histogram" tab > Settings of a measurement




**State**
Enables the histogram for the selected measurement. Enabling the histogram enables also the statistics and "Measure all events".

Remote command:
`MEASurement<mg>:STATistics:HISTogram[:ENABle]` on page 1174

**Reset**
Resets the cumulation of measurement results, all bins are set to zero.

Remote command:
`MEASurement<mg>:STATistics:HISTogram:RESet` on page 1175

**Number of bins**
Sets the number of columns (bins) to be displayed in the histogram. Cumulative histogram values are kept if you change the number of bins.

Remote command:
`MEASurement<mg>:STATistics:HISTogram:HBINs` on page 1174

**Auto scale**
If on, the range on the x-axis is continuously adjusted to the measured values. If off, you can set the offset and the scale of the histogram.

The unit of the x-axis is the unit of the measurement, e.g. Volt for an amplitude measurement.

Remote command:
`MEASurement<mg>:STATistics:HISTogram:AUTOscale` on page 1173

**Offset**
Sets the center value of the histogram on the x-axis if "Auto scale" is off. The offset is the measurement result in the middle of the histogram.

Remote command:
MEASurement<mg>:STATistics:HISTogram:HORizontal:OFFSet on page 1174

**Scale**
Sets the horizontal scale of the histogram if "Auto scale" is off.

Remote command:
MEASurement<mg>:STATistics:HISTogram:HORizontal:SCALe on page 1175

### Analysis of histograms on measurement results

To view the histogram in more detail, use the zoom.

You can export the histogram data to file, one histogram per file. See Section 13.5, "Histogram data export", on page 471.

For remote transfer of histogram data, use EXPort:HMEasurement<m>:DATA:HEADer? and EXPort:HMEasurement<m>:DATA[:VALues]?.

# 11 Spectrum analysis

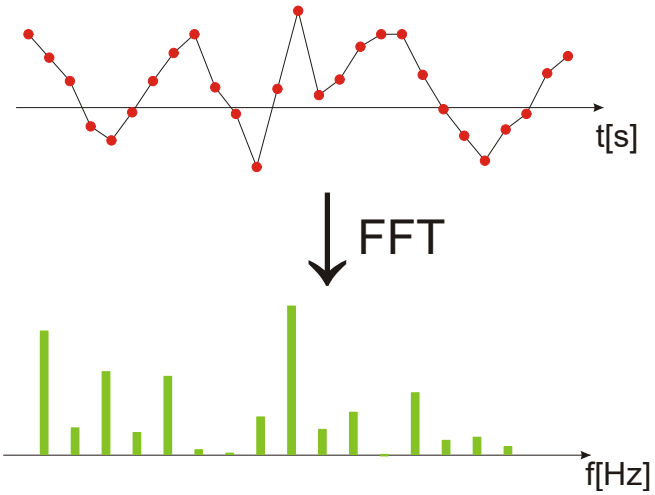
The MXO 3 provides an easy way to set up a spectrum analysis. The spectrum settings are independent of the time domain settings but the time and frequency domains are time-correlated.

You can analyze spectrum traces in various ways:

- Use peak lists.
- Use zoom.
- Perform cursor measurements.
- Export the spectrum data to files.
- Retrieve and transmit spectrum data for further analysis using remote commands.

## 11.1 Fundamentals of spectrum analysis

During spectrum analysis, a signal in the time domain is converted to a spectrum of frequencies. As a result, either the magnitude or the phase of the determined frequencies can be displayed. Spectrum analysis can be restricted to an extract of the original time base, and the results display can be restricted to a specified frequency range.

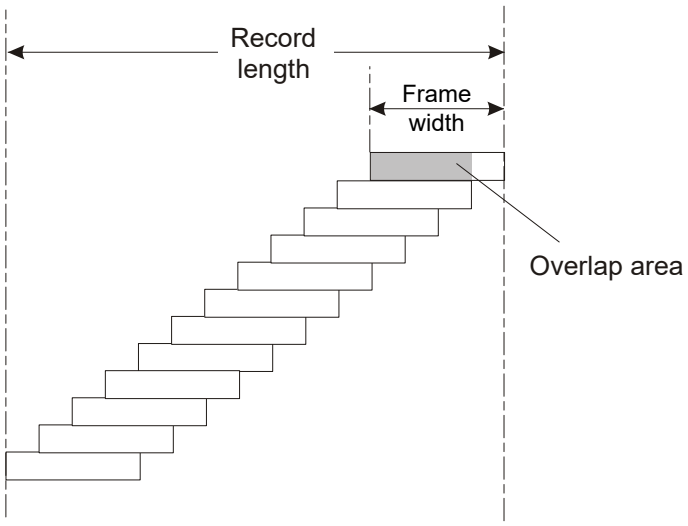



**Frames in FFT**

By default, one FFT per acqusition is calculated. The MXO 3 can calculate multiple FFTs per acquisition by dividing one record length into several *frames*. Thus, the MXO 3 can visualize how the frequency content of a signal changes over time which helps to detect intermittent or sporadic signal details. Furthermore, the MXO 3 allows consecutive frames to overlap. This is especially useful in conjunction with window functions since it enables a gap-free frequency analysis of the signal.

The overlapping factor can be set freely. The higher the overlap factor, the more frames are used. This leads to more individual results and improves detection of transient signal effects. However, it also extends the duration of the calculation. The

size of the frame depends on the number of input signal values (record length), the overlap factor, and the number of samples used for FFT calculation.




### Window functions

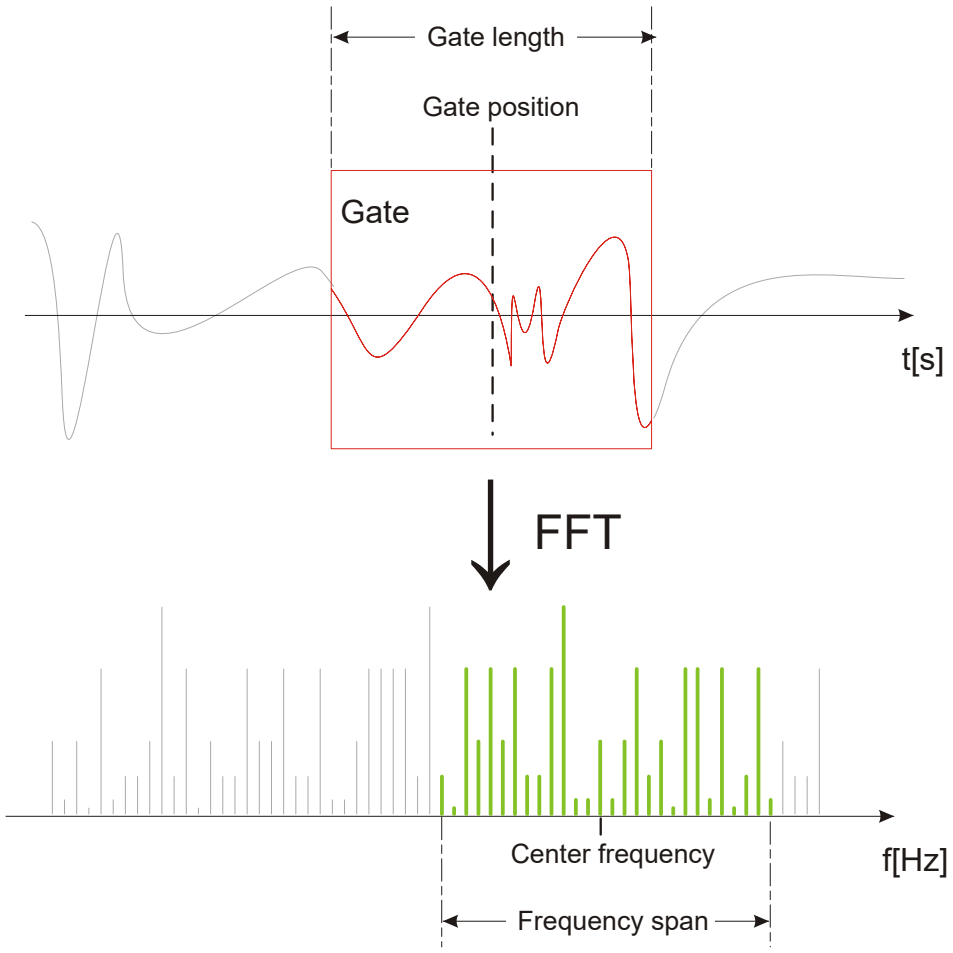
Each frame is multiplied with a specific window function after sampling in the time domain. Windowing helps minimize the discontinuities at the end of the measured signal interval and thus reduces the effect of spectral leakage, increasing the frequency resolution.

There are several window functions that can be used in FFT analysis. Each of the window functions has specific characteristics, including some advantages and some trade-offs. Consider these characteristics carefully to find the optimum solution for the measurement task.

For details, see "Window type" on page 321.

### Gating functions

You can restrict the time base of the input signal for which spectrum analysis is to be performed. You can define start and stop times for the time base extract.




### Restricting the result range

You can restrict the results of the spectrum analysis to a specified frequency range. The frequency range can be defined in two ways:

- Define a center frequency and frequency span
- Define start and stop frequencies

### Dependencies between spectrum parameters

Spectrum analysis in the MXO 3 is highly configurable. Several parameters, including the resolution bandwidth, frequency span and center frequency, can be defined according to your requirements. Note, however, that several parameters are correlated and not all can be configured independently of the others.

The **resolution bandwidth** defines the minimum frequency separation at which the individual components of a spectrum can be distinguished. Small values result in high precision, as the distance between two distinguishable frequencies is small. Higher values decrease the precision, but increase measurement speed.

The minimum achievable RBW depends on the integration time which is equivalent to the number of samples available for calculation. If a higher spectral resolution is

required, the number of samples must be increased by using a higher sample rate or longer record length. To simplify operation, some parameters are coupled and automatically calculated, such as record length and RBW.

The **frequency span** and **center frequency** define the start and stop frequency of the spectral diagram. By default, a suitable frequency range according to the resolution bandwidth is selected, in respect to performance and precision. Span and RBW settings are coupled, so that the parameters can be adjusted automatically as necessary.

With a **Span/RBW ratio** of 100 and a screen resolution of 1000 pixels, each frequency in the spectrum is displayed by 10 pixels. A span/RBW ratio of 1000 provides the highest resolution. For full flexibility, the span/RBW coupling can also be disabled. Note, however, that a higher span/RBW ratio (i.e. low RBW values and large frequency spans) result in large amounts of data and extend the duration of the calculation.

## 11.2 Configuring spectrum waveforms

During spectrum analysis, a signal in the time domain is converted to a spectrum of frequencies. A basic spectrum waveform can be displayed quickly. By defining additional parameters, the waveform can be configured in more detail.

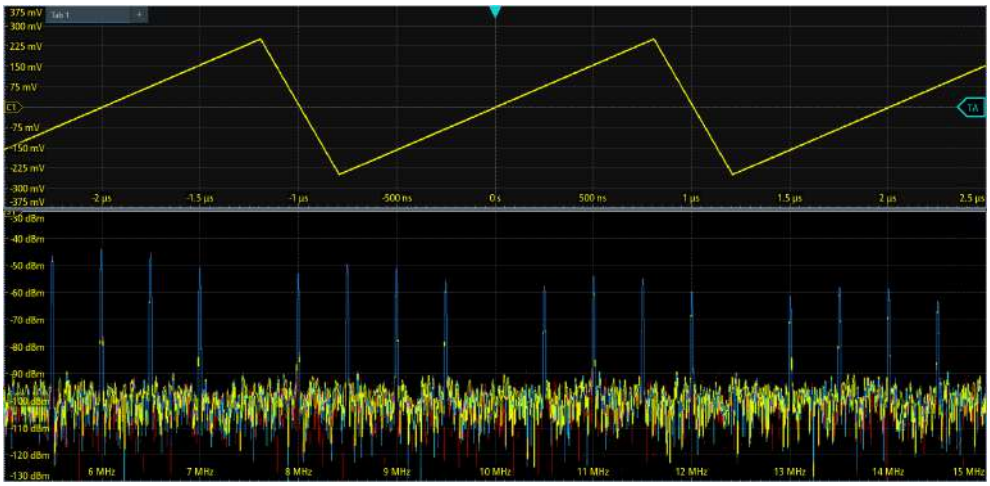
**To display a basic spectrum waveform**

1. Press the [Spec] key.

   The "Setup" tab of the "Spectrum" dialog box opens.

2. Set the "Source" to the input signal.

3. Enable "Display".




4. If necessary, edit the spectrum waveform parameters as described in the following procedures.

**To configure the spectrum**

By default, a suitable frequency range for the expected horizontal values according to the resolution bandwidth is selected, in respect to performance and precision. Span and RBW settings are coupled. If a more precise evaluation is required, for example for postprocessing in a different application, disable the coupling and change the frequency ranges and resolution bandwidth values as required.

1. Press the [Spec] key.

   The "Setup" tab of the "Spectrum" dialog box opens.

2. In the "Setup" tab, specify the frequency range you want to display using one of the following methods:
   - Select "CF span". Enter a "Center" and a "Span" that define the spectrum.
   - Select "Start Stop". Enter a "Start" and "Stop" frequencies that define the spectrum.
   - Tap the "Full span" button to display the complete spectrum resulting from the FFT analysis.

3. Define the resolution bandwidth for the spectrum result.
   The resolution bandwidth defines how precise the results are, i.e. how close together the individual frequencies can be. Small values result in high precision, as the distance between two distinguishable frequencies is small. Higher values decrease the precision, but increase performance.

   You can define the RBW manually, or couple it to other settings. Do one of the following:
   - To couple the RBW to the span, enable "Auto RBW". Define the "Span/RBW" ratio, the smaller the ratio, the higher the RBW becomes to display the same frequency span.
   - Disable "Auto RBW". Enter the "RBW" manually.

4. Tap "Advanced".

5. Select the most suitable "Window type" for your source data. Window functions are multiplied with the input values and thus can improve the spectrum display. For details, see "Window type" on page 321.

**To restrict the input values (gating)**

You can restrict the time range for which the FFT is calculated, resulting in a restricted spectrum.

1. Open "Menu" > "Spectrum" > "Gate" tab.

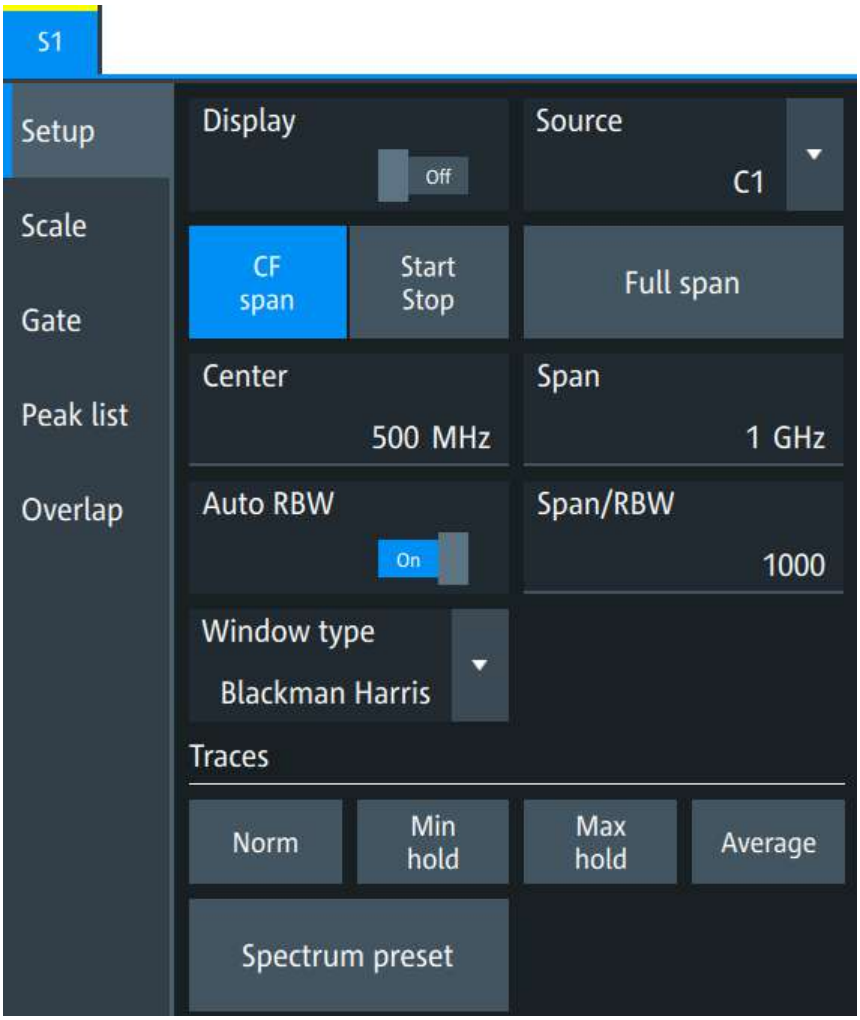
2. Enter the "Start" and the "Stop" times that define the gate area.

3. Set the "Position" and the "Width".

   The spectrum waveform displays the spectrum for the specified time span.

## 11.3 Spectrum setup

Access: "Menu" > "Spectrum" > "Setup" tab.




In this tab, you define the settings for the spectrum window. The display can be restricted to the results for a certain time base extract and to a specified frequency range.

The settings in this dialog are greyed out and cannot be changed, if "Coupling" is enabled in the "Scale" tab.

**Display**
Enables the spectrum.

Remote command:
`CALCulate:SPECtrum<sp>:STATe` on page 1198

**Source**
Selects the source for the spectrum.

Remote command:
CALCulate:SPECtrum<sp>:SOURce on page 1197

**Full span**
Displays the full frequency span.

**CF span, Start Stop**
Selects if the frequency span is defined through a "Center"/"Span" pair or through the "Start"/"Stop".

**Center**
Defines the position of the displayed frequency range, which is (Center - Span/2) to (Center + Span/2). The width of the range is defined using the frequency span setting.

Remote command:
CALCulate:SPECtrum<sp>:FREQuency:CENTer on page 1193

**Span**
The span is specified in Hertz and defines the width of the displayed frequency range, which is (Center - Span/2) to (Center + Span/2). The position of the span is defined using the "Center" setting.

Remote command:
CALCulate:SPECtrum<sp>:FREQuency:SPAN on page 1194

**Start**
Defines the start frequency of the displayed frequency span.

Remote command:
CALCulate:SPECtrum<sp>:FREQuency:STARt on page 1194

**Stop**
Sets the stop frequency of the displayed frequency span.

Remote command:
CALCulate:SPECtrum<sp>:FREQuency:STOP on page 1194

**Auto RBW**
Couples the frequency span to the "RBW" setting.

If span and RBW values are coupled, changing the span also changes the RBW.

Remote command:
CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution]:AUTO on page 1192

**Span/RBW**
Defines the coupling ratio for Span/RBW.

This setting is only available if "Auto RBW" is enabled.

Remote command:
CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution]:RATio on page 1192

**RBW**
Defines the resolution bandwidth. Note that the resolution bandwidth is correlated with the span, record length and acquisition time. If a constant record length is to be used, the RBW can be adapted if the required number of samples cannot be acquired.

Remote command:
`CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution][:VALue]` on page 1193

**Window type**
Selects the window type. Windowing helps minimize the discontinuities at the end of the measured signal interval and thus reduces the effect of spectral leakage, increasing the frequency resolution.

Various different window functions are provided in the MXO 3 to suit different input signals. Each of the window functions has specific characteristics, including some advantages and some trade-offs. Consider these characteristics carefully, to find the optimum solution for the measurement task.

| Window type | Frequency resolution | Magnitude resolution | Measurement recommendation |
|---|---|---|---|
| Rectangular | Best | Worst | Separation of two tones with almost equal amplitudes and a small frequency distance |
| Hamming<br>Hann | Good | Poor | Frequency response measurements, sine waves, periodic signals and narrow-band noise |
| Blackman Harris (default) | Good | Good | Mainly for signals with single frequencies to detect harmonics<br>Accurate single-tone measurements<br>Best compromise of frequency and magnitude resolutions |
| Gaussian | Good | Good | Weak signals and short duration |
| Flattop2 | Poor | Best | Accurate single-tone measurements |
| Kaiser Bessel | Poor | Good | Separation of two tones with differing amplitudes and a small frequency distance |

Remote command:
`CALCulate:SPECtrum<sp>:FREQuency:WINDow:TYPE` on page 1195

**Traces**
Spectrum analysis can only be performed on a maximum number of values at once. If more values must be calculated, the input signal is divided into segments, each of which is calculated separately. The segments need not to be disjunct. In this case, the arithmetic mode defines how the final result is calculated from the individual results.

The following methods are available:

"Norm" The data of only one segment is considered. In effect, no arithmetics are processed.

"Min hold" Determines the minimum result for each input value from the data of the current acquisition and the acquisitions before.

"Max hold" Determines the maximum result for each input value from the data of the current acquisition and the acquisitions before.

"Average" The average is calculated over the number of segments set with "Average count".

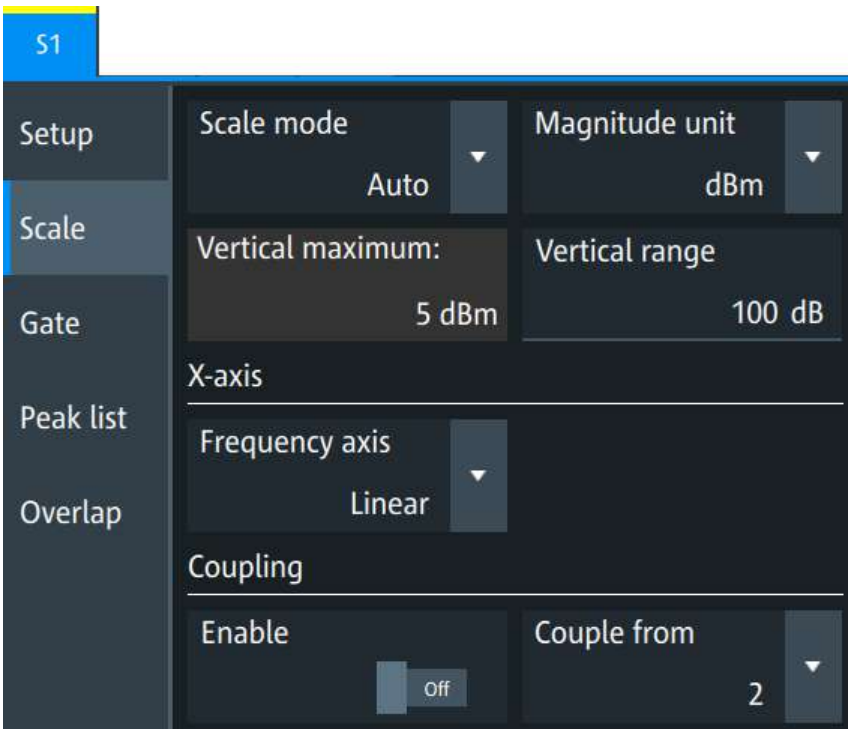
Remote command:
CALCulate:SPECtrum<sp>:WAVeform:AVERage:ENABle on page 1199
CALCulate:SPECtrum<sp>:WAVeform:AVERage:COUNt on page 1199
CALCulate:SPECtrum<sp>:WAVeform:MAXimum:ENABle on page 1199
CALCulate:SPECtrum<sp>:WAVeform:MINimum:ENABle on page 1199
CALCulate:SPECtrum<sp>:WAVeform:NORMal[:ENABle] on page 1200

**Spectrum preset**
Presets the spectrum measurement.

Remote command:
CALCulate:SPECtrum<sp>:PRESet on page 1196

## 11.4 Spectrum scale

Access: "Menu" > "Spectrum" > "Scale".




**Scale mode**
By default, the vertical scale is adapted to the current measurement results automatically to provide an optimal display. However, if necessary, you can define scaling values manually to suit your requirements.

**Note:** When you change the scaling values manually using the [Scale] rotary knob, the scale mode is set to "Manual" temporarily.

"Manual" Enter the required values for "Vertical maximum" and "Vertical range".

"Auto" "Vertical maximum" is read-only.

Remote command:
`CALCulate:SPECtrum<sp>:COUPling:MODE` on page 1191

**Magnitude unit**
Sets the unit for the y-axis.

The display values are calculated for the termination impedance that is set in "Vertical" > "Other" tab > "Impedance".

Remote command:
`CALCulate:SPECtrum<sp>:MAGNitude:SCALe` on page 1196

**Vertical maximum**
Sets the maximum displayed value on the vertical scale.

Remote command:
`CALCulate:SPECtrum<sp>:MAGNitude:LEVel` on page 1196

**Vertical range**
Sets the range of the spectrum values to be displayed.

Remote command:
`CALCulate:SPECtrum<sp>:MAGNitude:RANGe` on page 1196

**Frequency axis**
Defines the scaling method for the frequency axis (x-axis) of the spectrogram.

Available are logarithmic and linear scaling.

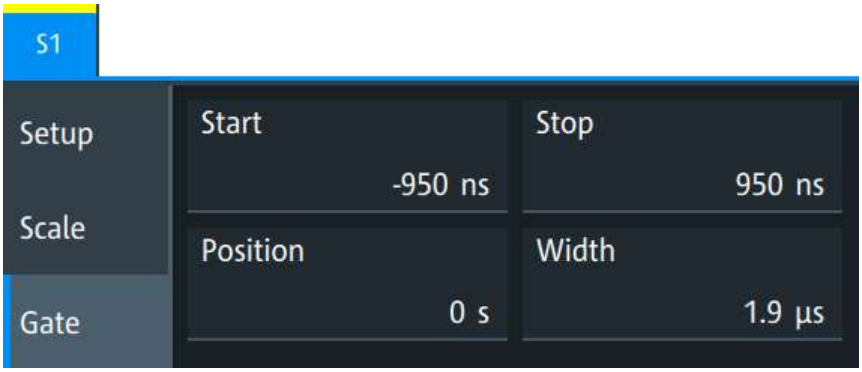
Remote command:
`CALCulate:SPECtrum<sp>:FREQuency:SCALe` on page 1193

**Coupling: Enable, Couple from**
If the coupling is enabled, the settings of the selected "Couple from" spectrum are used in the "Setup" of the current spectrum. Enable "Enable", then select "Couple from".

Remote command:
`CALCulate:SPECtrum<sp>:COUPling[:ENABle]` on page 1191
`CALCulate:SPECtrum<sp>:COUPling:FROM` on page 1191

## 11.5 Spectrum gate

Access: "Menu" > "Spectrum" > "Gate"




Spectrum gating restrict the spectrum analysis to a user-defined region of the captured time domain signal.

**Start**
Sets the starting value for the gate.

Remote command:
`CALCulate:SPECtrum<sp>:GATE:STARt` on page 1200

**Stop**
Sets the end value for the gate.

Remote command:
`CALCulate:SPECtrum<sp>:GATE:STOP` on page 1201

**Position**
Sets the position of the displayed frequency range.

The width of the gate is defined using the "Width" setting.

Remote command:
`CALCulate:SPECtrum<sp>:GATE:POSition` on page 1200

**Width**
Defines the width of the displayed gate.

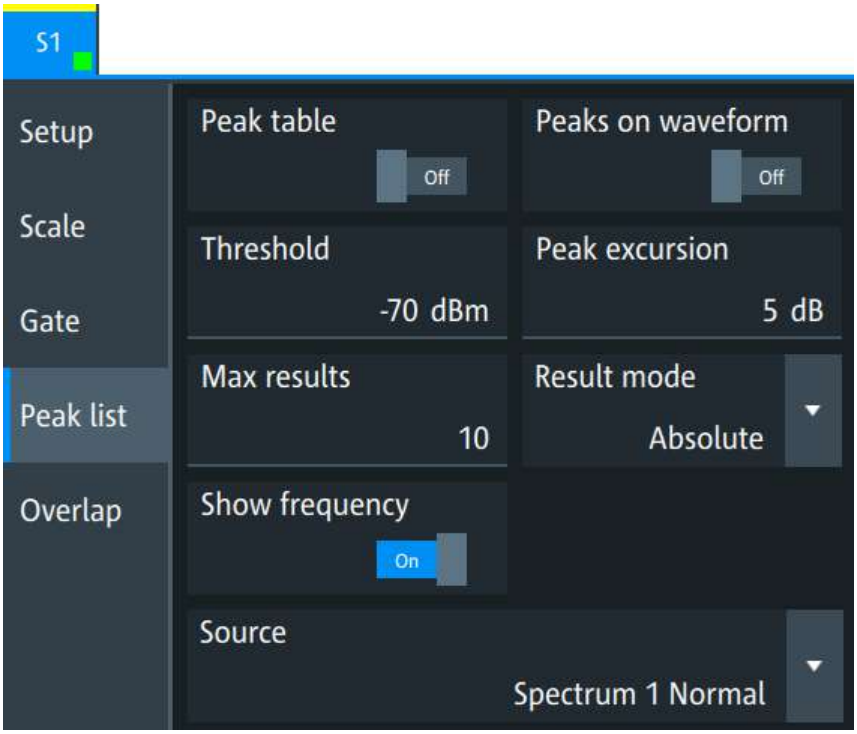
The position of the span is defined using the "Position" setting.

Remote command:
`CALCulate:SPECtrum<sp>:GATE:WIDTh` on page 1201

## 11.6 Spectrum peak list

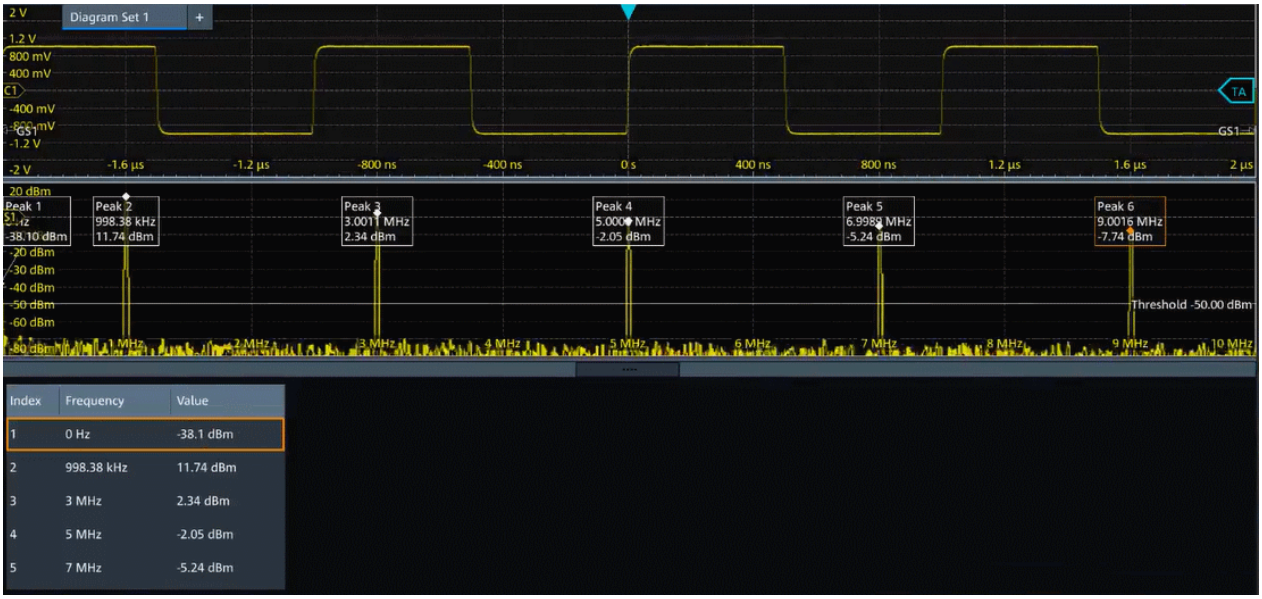
A peak list measurement detects all peaks above a user-definable threshold and optionally indicates the peaks in the spectrum diagram.

Access: "Menu" > "Spectrum" > "Peak list".




In the peak list dialog, you can define various criteria for a peak search. The peaks can be indicated in the frequency diagram by peak boxes, and the measured peak frequencies and magnitudes are listed in the peak list table.

The number of determined peaks can be defined. You can sort the results by frequency or power value, and the peak labels are adjusted accordingly.




Also, you can adjust some of the display settings, see Section 5.3.5, "Peak list", on page 103.

To analyze the peaks in a spectrum, you can use cursors on the spectrum waveform and peak search functions. For details, see Section 10.1.3.3, "Peak search tab", on page 268.

The results in the peak table can be saved to file. See Section 13.6, "Results", on page 475.

Remote commands query peak results and to change the results display:

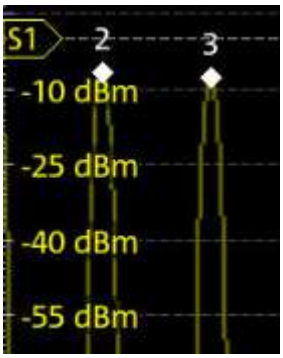
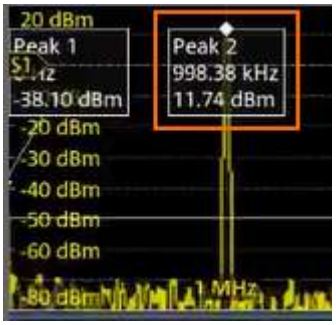
- `CALCulate:SPECtrum<sp>:PLISt:RESult[:VALue]?` on page 1203
- `CALCulate:SPECtrum<sp>:PLISt:SORT:COLumn` on page 1204
- `CALCulate:SPECtrum<sp>:PLISt:SORT:ORDer` on page 1204
- `CALCulate:SPECtrum<sp>:PLISt:LABel:INVert` on page 1205

**Peak table**
Enables the display of the peak table.

Remote command:
`CALCulate:SPECtrum<sp>:PLISt[:STATe]` on page 1202

**Peak on waveform**
Displays a box with a description for each detected peak in the spectrum, including the magnitude. If "Show frequency" is enabled, the frequency values are also displayed.

If this option is disabled, the peaks are marked on the diagram only with a point marker.

| "Peak on waveform" > "Off" | "Peak on waveform" > "On" |
|---|---|
|  |  |





Remote command:
`CALCulate:SPECtrum<sp>:PLISt:SPOW` on page 1203

**Threshold**
Sets an absolute threshold as an additional condition for the peak search. Only peaks that exceed the threshold are detected.

This setting is only available for spectrum waveforms. It is valid for cursor measurements, spectrum measurements and peak search.

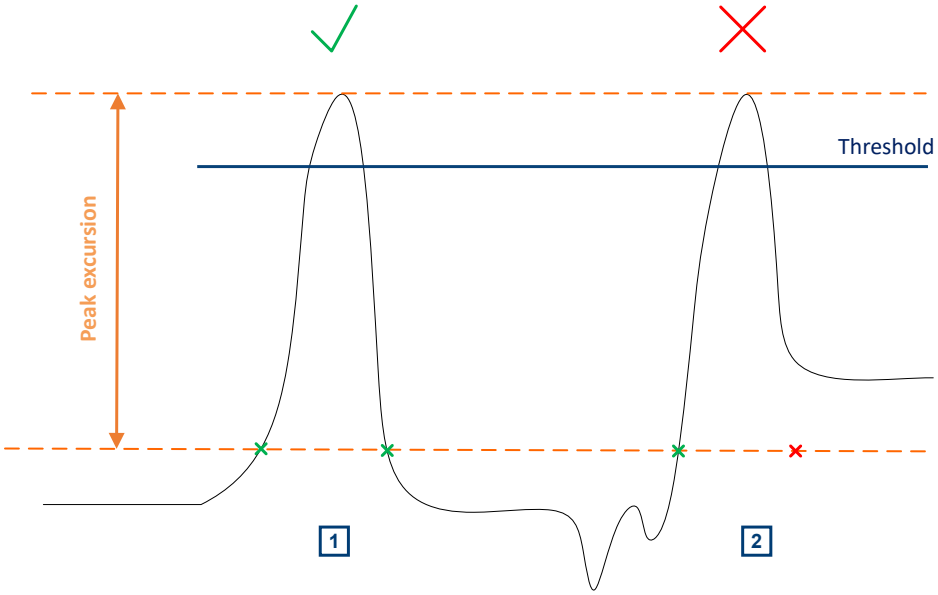
Remote command:
`CALCulate:SPECtrum<sp>:THReshold` on page 1198

**Peak excursion**
Defines a minimum level value by which the waveform must drop left and right of the local maximum to be listed as a peak. Enter a peak excursion value to omit close by peaks and list just the highest peak.

This setting is only available for spectrum waveforms. It is valid for cursor measurements, spectrum measurements and peak search.

**Example:**

In the figure below, 2 peaks are shown, that rise above the defined "Threshold". Peak 1 fulfills the defined "Peak excursion" value and is counted as a peak. Peak 2 does not fulfill the defined "Peak excursion" value on the right and is not counted as a peak.




Remote command:
`CALCulate:SPECtrum<sp>:PEXCursion` on page 1198
`CURSor<cu>:PEXCursion` on page 1189

**Max results**
Sets the maximum number of measurement results that are listed in the result table.

Remote command:
`CALCulate:SPECtrum<sp>:PLISt:MAXCount` on page 1202

**Result mode**
Selects how the measurement results are displayed.

"Absolute" The peaks are shown in absolute value, dBm.

"Relative" The level of the carrier is shown in absolute value dBm. The values of the peaks are shown relatively to the carrier in dBc.

Remote command:
CALCulate:SPECtrum<sp>:PLISt:MODE on page 1202

**Show frequency**
Includes the frequency of the detected peak in the diagram labels.

Remote command:
CALCulate:SPECtrum<sp>:PLISt:LABel:FREQuency[:STATe] on page 1204

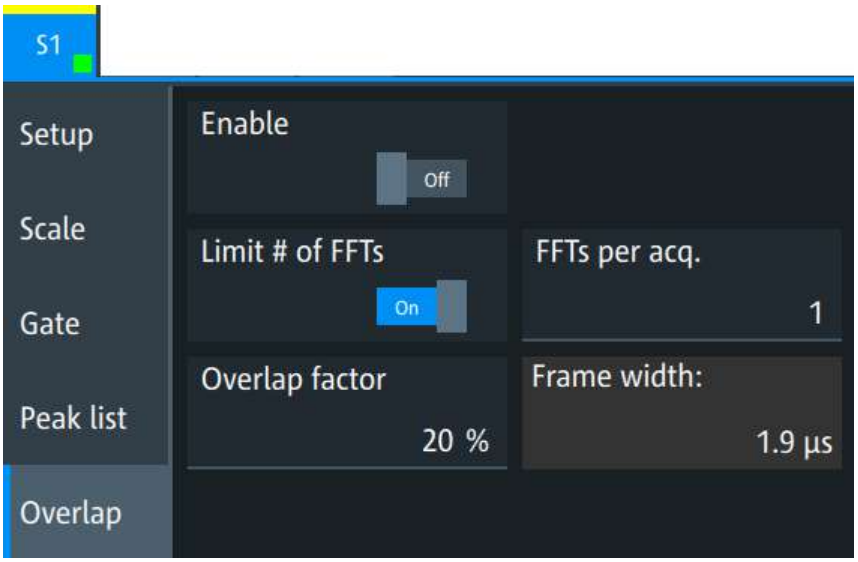
**Source**
Selects the source of the peak table. You can select one of the traces that is enabled with Traces.

Remote command:
CALCulate:SPECtrum<sp>:PLISt:SOURce on page 1203

## 11.7 Spectrum overlap

Access: "Menu" > "Spectrum" > "Setup" tab.

For an description of the basic principles of FFt frames and overlapping, see "Frames in FFT" on page 314.




**Enable**
Activates the use of frames and overlap to create the FFT spectrum.

Remote command:
CALCulate:SPECtrum<sp>:OVERlap:ENABle on page 1206

**Limit # of FFTs**
Limits the number of FFTs that are done per acquisition.

If limitation is off, the firmware sets the number of FFTs automatically to the maximum number that fit in the width of the overview window. It is possible, that some data at the left and the right of the display does not fit in the frames. The gate is set automatically and cannot be changed.

If limitation is active, you can set the number of frames ("FFTs per acq." on page 329). You can change the gate, which changes the frame width, too.

Remote command:
`CALCulate:SPECtrum<sp>:OVERlap:FCOunt:LIMit:ENABle` on page 1207

**FFTs per acq.**
Sets the number of frames to be used for FFT with active limitation. The frame width remains unchanged, and the resulting gate width is *frame width * FFTs per acqu*.

Remote command:
`CALCulate:SPECtrum<sp>:OVERlap:FCOunt:VALue` on page 1207
`CALCulate:SPECtrum<sp>:OVERlap:FCOunt:ADJusted?` on page 1207

**Overlap factor**
Defines the minimum factor by which two neighboring frames overlap. Depending on the settings, the system adjusts the value upward.

The higher the overlap factor, the more frames are used. This leads to more detailed results and improves detection of transient signal effects. However, it also extends the duration of the calculation.

Remote command:
`CALCulate:SPECtrum<sp>:OVERlap:FACTor:VALue` on page 1206
`CALCulate:SPECtrum<sp>:OVERlap:FACTor:ADJusted?` on page 1206

**Frame width**
Shows the current frame width.

Increasing the RBW increases the frame width, and vice versa. With active limitation of the frame number, changing the gate results in adjustment of the frame width.

Remote command:
`CALCulate:SPECtrum<sp>:OVERlap:FWIDth?` on page 1208

# 12 Applications

All available applications are provided in the "Apps" dialog.

► To open an application, press the [Apps] key.

Some applications are described in other sections. See:

- Section 9.4, "Reference waveforms", on page 245
- Section 9.5, "Waveform histograms", on page 254
- Section 16, "Waveform generator (option R&S MXO3-B6)", on page 900
- Section 13.9, "File browser dialog", on page 486
- Section 14, "Protocol analysis", on page 488

Also the "Demo" is available in the "Apps" dialog. It is for demonstration purposes of the functionalities of MXO 3.

## 12.1 Mask testing

### 12.1.1 About mask testing

Masks are used to determine whether the signal remains within specified limits, e.g. to uncover signal anomalies or test compliance and stability of digital signals. The limits are specified as "mask", which is laid over the input signal in the display. Thus you can easily detect where the signal violates the mask.

Mask testing with MXO 3 is hardware accelerated. It is the industry's fastest mask test with hardly any impact on the acquisition rate, thus mask violations are detected fast and reliably.

With MXO 3, you can define your own masks easily. Specific actions can be executed when mask violations occur. For error analysis, you can stop the acquisition on a failed test and use the history to look at the previous waveforms.

**Mask test**

A mask test consists of:

- Mask definition.
- Waveform to be tested.
- Actions to be taken on violation or successful completion.

**Mask definition**

A mask can be created in several ways:

- The individual mask points are defined, either on the touchscreen or as numerical values. This mask type is called *polygon*. A polygon can have several segments.

For details, see Section 12.1.3, "Mask definition: polygon", on page 335.

- The mask is derived from an existing waveform. This mask type is called *auto mask*.
  For details, see Section 12.1.4, "Mask definition: auto mask", on page 337.

**Polygon masks**

A polygon mask can have one or more segments. Each segment has a predefined shape: rectangle, square, diamond, hexagon or octagon. You can change the shape by moving the edge points on the screen or change the numerical values. You can also add and remove single edge points.

A mask segment can have only one range in y direction at a certain time point. If a shape is not valid, the invalid point is marked red, and an attention symbol is shown at the segment name.

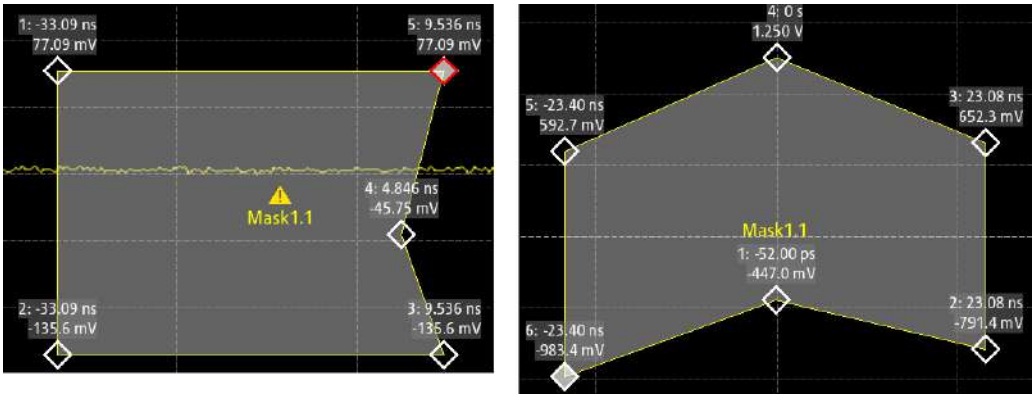
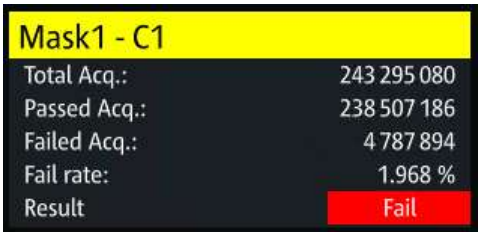



***Figure 12-1: Invalid mask segment (left) and valid mask segment (right)***

To check the validity in remote control, use `MTESt<m>:SEGMent<n>:VALid?` and `MTESt<m>:SEGMent<n>:POINt<o>:VALid?`.

### 12.1.1.1 Results of a mask test

A mask test shows the following test results:




**Total acq.**
Number of tested acquisitions.

Remote command:
`MTESt<m>:RESult:COUNt:WAVeforms?` on page 1221

**Passed acq.**
Number of acquisitions that passed the test.

Remote command:
`MTESt<m>:RESult:COUNt:PWAVeforms?` on page 1221

**Failed acq.**
Number of failed acquisitions.

Remote command:
`MTESt<m>:RESult:COUNt:FWAVeforms?` on page 1221

**Fail rate**
Ratio of failed acquisitions to the number of tested acquisitions.

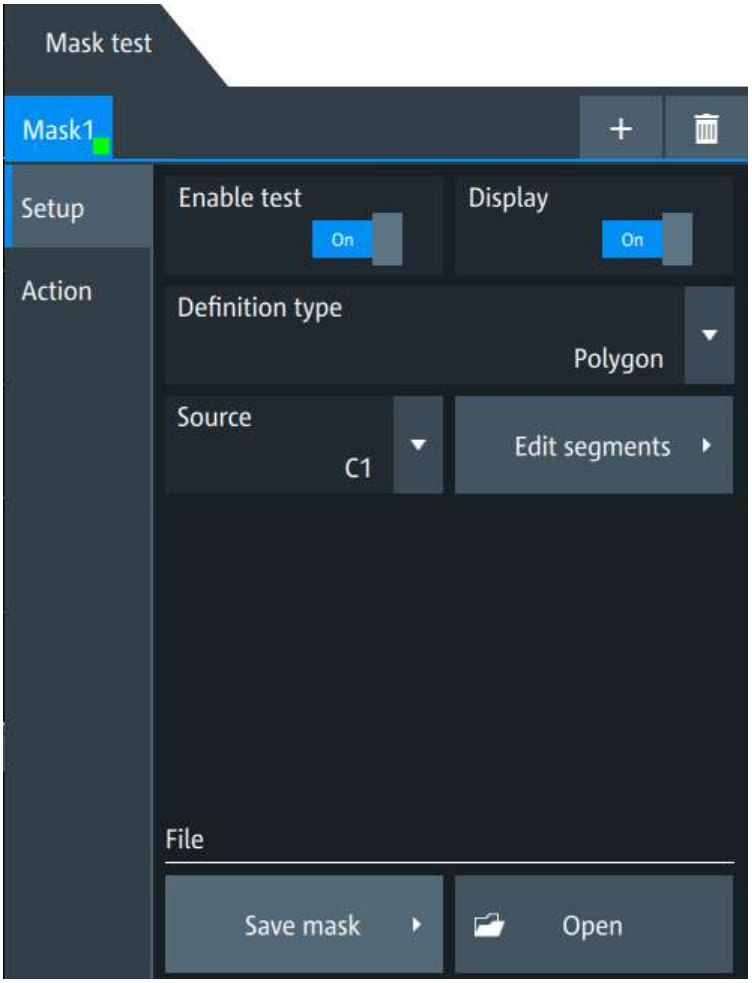
Remote command:
`MTESt<m>:RESult:FRATe?` on page 1221

**Result**
A test has failed if at least one acquisition violated the mask.

Remote command:
`MTESt<m>:RESult[:RESult]?` on page 1222

### 12.1.2 Mask test setup: general settings

Access: "Menu" > "Apps" > "General" tab > "Masks" > "Setup" tab




**Add mask**
Adds a mask test and opens the subtab to configure the new test.

Remote command:
`MTESt<m>:ADD` on page 1209

**Enable test**
Activates or deactivates the mask test.

If the acquisition is running, the test starts immediately. Otherwise, the test starts when acquisition is started. The testing is stopped when acquisition is stopped, or if a stop action is configured.

Remote command:
`MTESt<m>:STATe` on page 1211

**Display**
Displays all mask segments of the selected mask in the diagrams, or hides them.

Remote command:
`MTESt<m>:VISible` on page 1211

**Definition type**
Selects the mask type.

- Polygon: define mask segments by setting their corner points. See Section 12.1.3, "Mask definition: polygon", on page 335.
- Auto mask: define the mask based on an active waveform. See Section 12.1.4, "Mask definition: auto mask", on page 337.

Remote command:
None, each mask type has its own commands for definition.

**Source**
Selects the waveform to be tested against the mask. For auto mask definition, it is also the waveform from which the mask is created.

All analog channel waveforms, math waveforms, XY-plots and norm traces of spectrums can be tested.

Remote command:
`MTESt<m>:SOURce` on page 1211

**Edit segments**
Opens a dialog to define the mask, depending on the selected "Definition type" on page 334.

**Save mask**
Opens a dialog to define the storage location and filename, and save the mask.

| | |
|---|---|
| "<Folder>" | Opens a file explorer where you can select the directory where the file is saved. |
| "File name base" | Sets a name for the file, without extension. The name is extended with a time stamp when the file is saved. Thus, multiple "Save" actions are possible without changing the filename. |
| "File extension" | The format of the mask file is always `.xml`. |
| "Save" | Saves the file in the defined folder using the "File name base". If the specified file already exists, it is overwritten with the new data. |
| "Save As" | Opens a file explorer where you select the folder, the file type, enter the filename and save the file. |

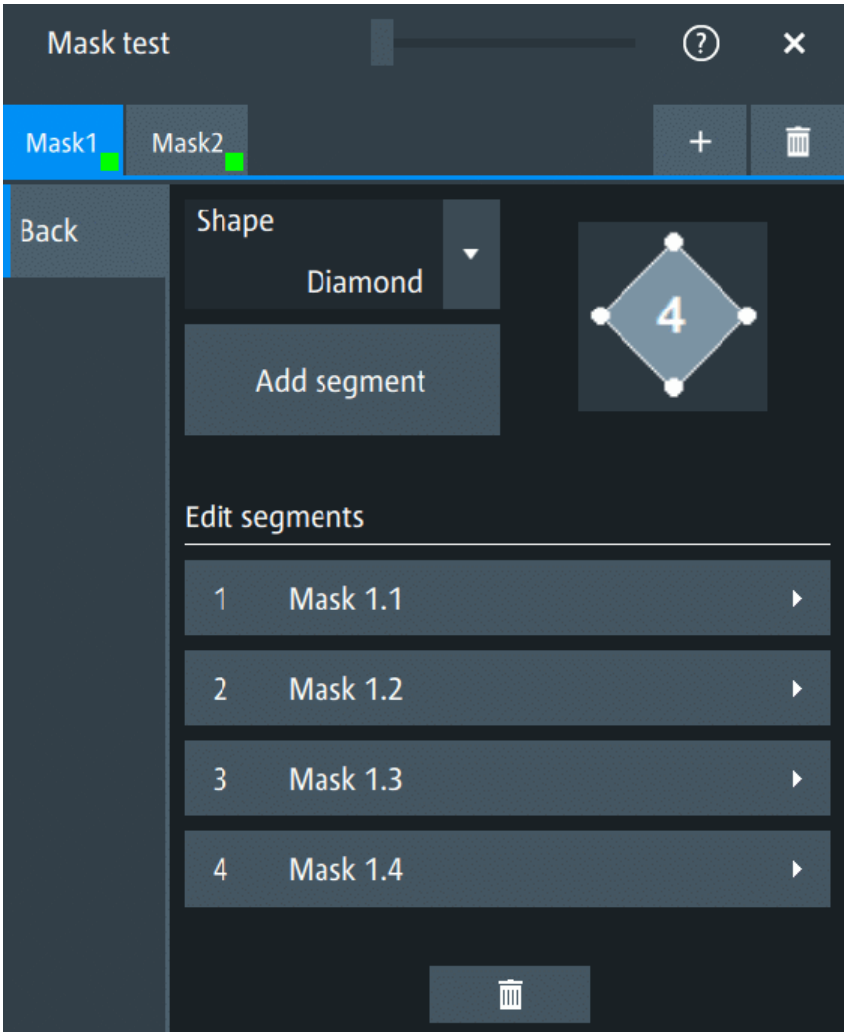
Remote command:
`MTESt<m>:IMEXport:NAME` on page 1210
`MTESt<m>:IMEXport:SAVE` on page 1210

**Open**
Opens a file explorer to select a previously stored mask file. The selected mask is loaded and used for test.

Remote command:
`MTESt<m>:IMEXport:NAME` on page 1210
`MTESt<m>:IMEXport:OPEN` on page 1210

### 12.1.3 Mask definition: polygon

"Menu" > "Apps" > "General" tab > "Masks" > "Setup" tab > select "Definition type" = "Polygon" > "Edit segments"

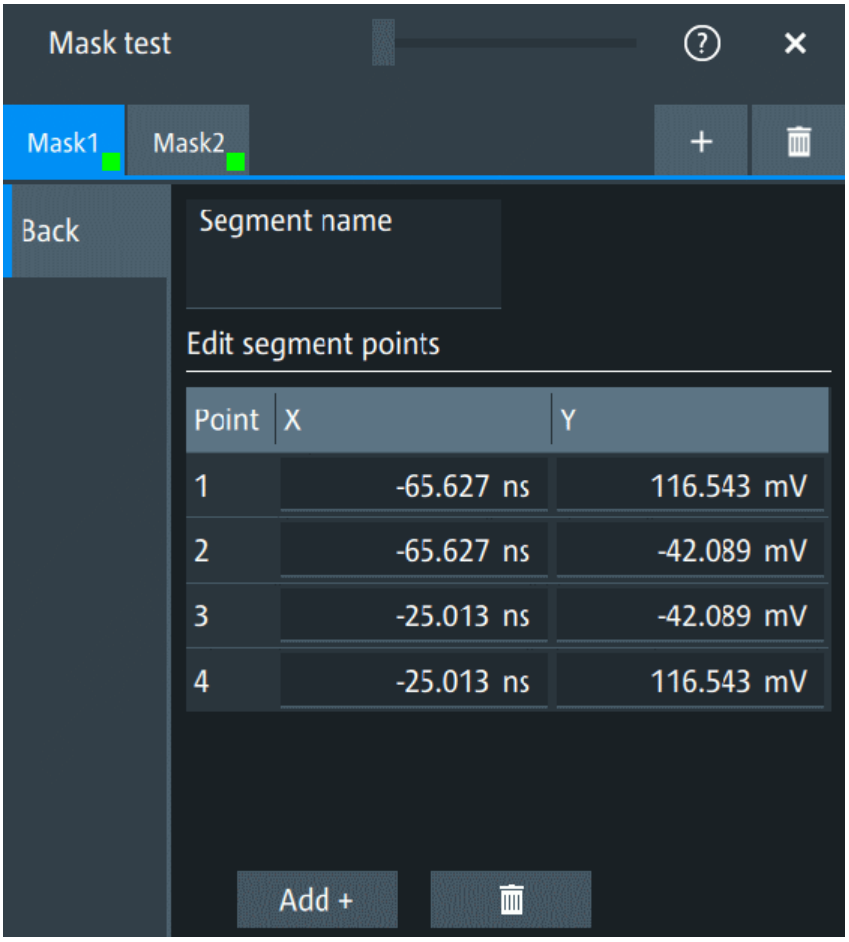



**Shape**
Selects the shape of the user-defined mask: rectangle, diamond, hexagon, octagon.

**Add segment**
Adds a new segment with the selected shape to the mask.

Remote command:
`MTESt<m>:SEGMent<n>:ADD` on page 1212

**Edit segments**
Dialog to edit the mask segment. For each segment, you define the coordinates of the corner points. You can add and remove points and define the shape as needed.




**Point, X, Y ← Edit segments**
The table shows the horizontal and vertical positions of the corner points. You can change each value.

Remote command:
`MTESt<m>:SEGMent<n>:POINt<o>:X` on page 1214
`MTESt<m>:SEGMent<n>:POINt<o>:Y` on page 1214
`POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:X` on page 1298
`POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:Y` on page 1299

**Add + ← Edit segments**
Appends a new point to the segment. You cannot change the order of the points.

Remote command:
`MTESt<m>:SEGMent<n>:POINt<o>:ADD` on page 1212
`MTESt<m>:SEGMent<n>:POINt<o>:COUNt?` on page 1213

**Segment name ← Edit segments**
Enter a name for the selected mask segment.

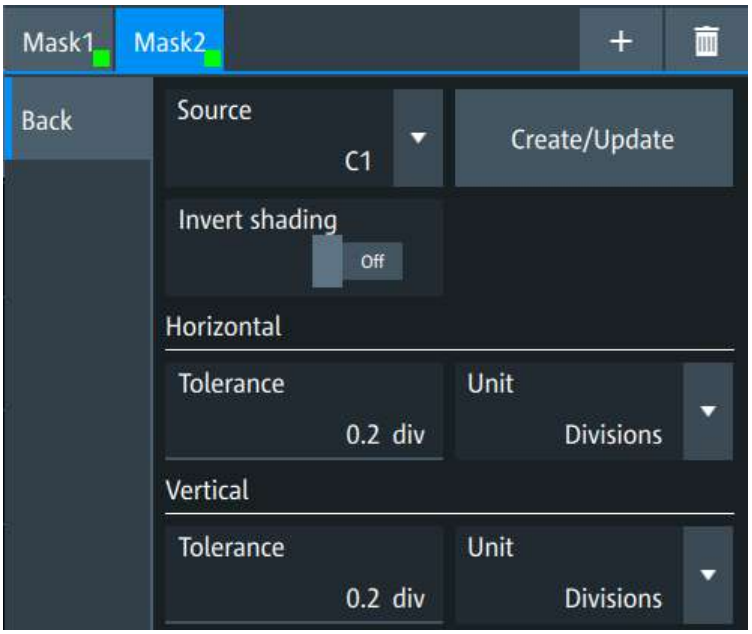
**Delete, Delete all**
switches to the delete mode. You can delete single segments or single points by selecting the delete icon in the table row. You can also delete all segments or points with "Delete all". To leave the delete mode, tap the blue delete icon.

Remote command:
`MTESt<m>:SEGMent<n>:POINt<o>:REMove` on page 1213
`MTESt<m>:SEGMent<n>:REMove` on page 1215

### 12.1.4 Mask definition: auto mask

"Menu" > "Apps" > "General" tab > "Masks" > "Setup" tab > select "Definition type" = "Auto mask" > "Edit segments"

An auto mask, also known as waveform mask, is created from an existing waveform. The waveform envelope builds the upper and lower limit line of the mask, and the limits are moved by a given value from the original waveform. The result is a tolerance tube around the waveform that is used as mask.




**Source**
See "Source" on page 334.

**Create/Update**
Creates or updates the upper and lower mask limit from the envelope of the selected "Source" waveform by copying the waveform data.

Remote command:
`MTESt<m>:AUTomask:UPDate` on page 1216

**Invert shading**
If enabled, the area inside the mask is highlighted. If disabled, the area outside the mask limits is highlighted.

**Tolerance (Horizontal)**
Sets the width of the mask in horizontal direction. The value is added to the left and the right of the x-values of the copied source waveform. The overall mask width is twice the specified horizontal tolerance. Depending on the selected "Unit" on page 338, you can enter the tolerance in divisions or in the unit of the waveform's x-axis.

Remote command:
`MTESt<m>:AUTomask:XTOLerance:DIVisions` on page 1216
`MTESt<m>:AUTomask:XTOLerance:PHYSical` on page 1216

**Tolerance (Vertical)**
Sets the height of the mask in vertical direction. The value is added to and subtracted from y-values of the copied source waveform. The overall mask height is twice the specified vertical tolerance. Depending on the selected "Unit" on page 338, you can enter the tolerance in divisions or in the unit of the waveform's y-axis.

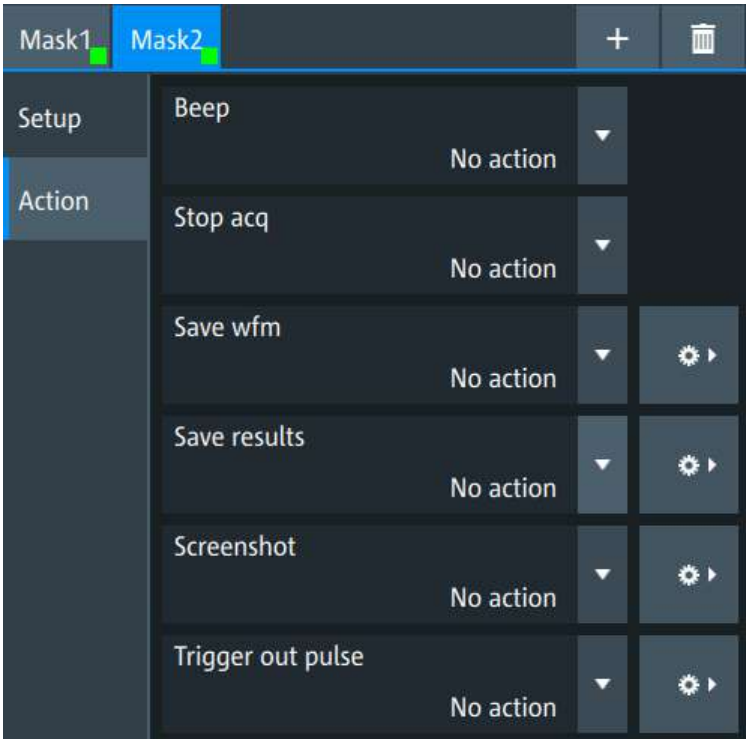
Remote command:
`MTESt<m>:AUTomask:YTOLerance:DIVisions` on page 1217
`MTESt<m>:AUTomask:YTOLerance:PHYSical` on page 1217

**Unit**
Set the unit of the mask tolerance in horizontal and vertical direction: divisions or in the unit of the corresponding axis.

Remote command:
`MTESt<m>:AUTomask:XTUNit` on page 1217
`MTESt<m>:AUTomask:YTUNit` on page 1217

### 12.1.5 Actions on mask test results

Access: "Menu" > "Apps" > "General" tab > "Masks" > "Action" tab




The action settings define what happens when an acquisition passes the mask test successfully, or when it hits the mask. All available actions can be initiated at the same time.

**Beep**
Generates a beep sound.

Remote command:
`MTESt<m>:ONViolation:BEEP` on page 1218
`POWer<m>:SOA:MTESt<n>:ONViolation:BEEP` on page 1300

**Stop acq**
Stops the acquisition on completion or violation.

Remote command:
`MTESt<m>:ONViolation:STOP` on page 1219
`POWer<m>:SOA:MTESt<n>:ONViolation:STOP` on page 1300

**Save wfm**
Saves the waveform data to a file according to the folder, filename and extension settings in the "Menu" > "Save/recall" > "Save" > "Waveform" dialog.

To select the waveforms and adjust the file path, tap the settings icon .

See also Section 13.4.2, "Saving waveforms", on page 460.

Remote command:
`MTESt<m>:ONViolation:WFMSave` on page 1220
`POWer<m>:SOA:MTESt<n>:ONViolation:WFMSave` on page 1301

**Save results**
Saves the results to a file according to the settings in "Menu" > "Save/recall" > "Save" > "Results".

To select the waveforms and adjust the file path, tap the settings icon.

See also Section 13.6, "Results", on page 475.

Remote command:
`MTESt<m>:ONViolation:SRESults` on page 1219

**Screenshot**
Saves a screenshot according to the settings in the "Menu" > "Save/recall" > "Save" > "Screenshot" dialog.

To adjust the screenshot settings, tap the settings icon.

See also Section 13.8, "Screenshots", on page 481.

Remote command:
`MTESt<m>:ONViolation:SCReenshot` on page 1219
`POWer<m>:SOA:MTESt<n>:ONViolation:SCReenshot` on page 1300

**Trigger out pulse**
Selects, if a pulse is provided to the [Trigger Out] connector on the rear panel. The trigger-out signal is used to synchronize the measurements of other instruments.

To adjust the outgoing signal, tap the settings icon.

Remote command:
`MTESt<m>:ONViolation:TRIGgerout` on page 1220
`POWer<m>:SOA:MTESt<n>:ONViolation:TRIGgerout` on page 1301

## 12.1.6 Working with masks

This section explains step by step how mask tests are configured. For the explanation of the individual settings, see Section 12.1.2, "Mask test setup: general settings", on page 332.

### 12.1.6.1 Working with masks using the toolbar

**To create a polygon mask graphically using the toolbar**

1. Select the "Add mask" icon on the toolbar. If the icon is not visible, configure the toolbar as described in Section 4.4.2, "Configuring the toolbar", on page 63.



2. In the "Add mask" overlay menu, select the "Definition type" of the mask: "Polygon".



3. In the overlay menu, select the channel that you want to test against the mask.
4. In the overlay menu, the "Add mask" function is selected. Draw the diagonal of the rectangle on the screen.
5. Now, the "Add segment" function is selected in the overlay menu. Draw another segment on the screen.
6. If you need another mask, select "Add mask". Draw the first segment of the new mask.
7. To finish the mask definition, close the overlay menu.

   The mask test is enabled by default. If the acquisition is running, the mask test starts automatically and immediately. Otherwise, start the acquisition to start the test.

**To modify a polygon mask using the toolbar**

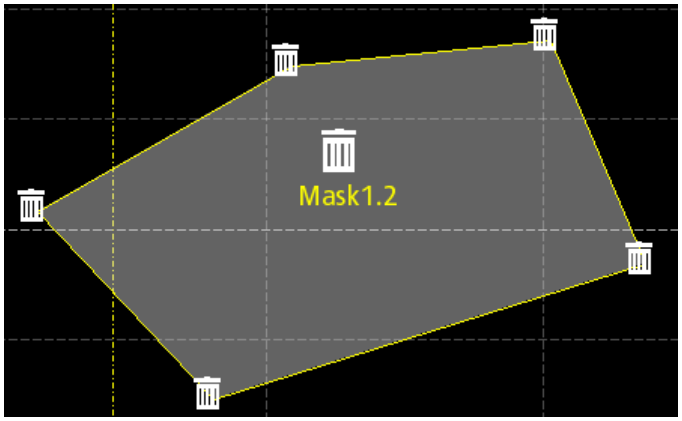
You already have created a mask and want to change its shape and add segments.

1. Select the mask segment in the diagram.

   The segment is marked and the overlay menu opens. One of the points is selected by default, and its coordinates are shown.




2. In the diagram, you can change the shape of the mask segment:
   - To move the segment, tap in the segment, hold and move it.
   - To move an edge, tap and hold the edge and move it.
   - To move a point, tap and hold the point and move it.
3. The overlay menu provides more functions.
   To add a point to the selected segment:
   a) Select "Add point" in the overlay menu.
   b) Tap outside the segment, hold and move the point to the required position.
4. To delete a segment or point:
   a) Select "Delete" in the overlay menu.



b) To delete a point, tap its delete icon.
c) To delete the segment, tap the "Delete" icon inside the segment.

5. To enter the exact coordinates of a point:
   a) Select the mask segment and the point in the diagram.
   b) In the overlay menu, change the X- and Y-values.
6. To change the waveform to be tested, select "Source" in the overlay menu.

**To create an auto mask using the toolbar**

1. Select the "Add mask" icon on the toolbar. If the icon is not visible, configure the toolbar as described in Section 4.4.2, "Configuring the toolbar", on page 63.



2. In the "Add mask" overlay menu, select the "Definition type" of the mask: "Auto mask".
3. In the overlay menu, select the channel that you want to use for mask definition.
4. In the overlay menu, select "Copy".
   The mask is created with default settings, and the selected source is tested against the mask immediately.
5. To change the mask tube, select "Setup" in the overlay menu.

### 12.1.6.2 Working with masks using the dialog

1. Open the "Mask test" dialog: "Menu" > "Apps" > "General" tab > "Masks" > "Setup" tab.
   If there is already at least one defined mask, tap its result table to open the dialog.
2. Select "Delete mask" to create the first mask. To create more masks, use the + icon.
3. Select the "Source", the waveform to be tested.
4. Select "Edit segments".
5. Select the "Shape" of the new segment.
6. Select "Add segment".
   The new segment is listed in the dialog and shown in the diagram.
7. Under "Edit segments", select the segment.
8. Change the X- and Y-values as needed.
9. To name the segment, select "Segment name". Enter the name.
10. To add more points, use "Add +".

11. To delete points:
    a) Select .
    b) To delete one point, select the delete icon in its table row.
    c) To delete all points, select "Delete all"
    d) To leave the delete mode, tap the blue delete icon.
12. Select "Back" twice to finish the segment setup.
13. In the "Setup" tab, enable the test if it is off.
14. If the acquisition is running, the mask test starts automatically and immediately. Otherwise, start the acquisition to start the test.

#### 12.1.6.3 Setting up a mask test

In addition to the mask definition, you can define the actions to be taken if a test has failed or completed successfully.

1. Open the "Mask test" dialog: "Menu" > "Apps" > "General" tab > "Masks".
2. Select the "Action" tab.
3. For each action, select when the action will be executed:
   - No action
   - On successful completion
   - On violation of the mask
4. For the actions "Save wfm" and "Trigger out pulse", tap the icon and configure the settings.

#### 12.1.6.4 Running a mask test

You can perform continuous testing with "Run / stop" or test a specified number of acquisitions with "Run single". If the test is enabled in the "Setup" tab, the test starts when the acquisition starts. The test runs until the acquisition is stopped.

#### 12.1.6.5 Saving and loading masks

Mask definitions are remained until they are changed or deleted, or a preset is performed. If you want to keep a mask definition, you can save and reload it.

**To save a mask**

1. Open "Menu" > "Apps" > "General" tab > "Masks" > "Setup" tab.
2. Select "Save mask".
3. Select "Save as".
4. In the file browser, select the directory and enter the filename. The format of the mask file is always `.xml`.

5. Select "Save". If the specified file already exists, it is overwritten with the new data.

**To load a mask**

1. In the "Masks" >"Setup" tab, select "Open".
2. In the file browser, select the mask file.
3. Select "Open".

### 12.1.6.6 Mask testing on history waveforms

In the same way as for running acquisitions, you can set up and perform the mask testing on history waveforms.

1. Perform and finish the acquisition.
2. Define the mask, and enable the test. See Section 12.1.6.1, "Working with masks using the toolbar", on page 340 and Section 12.1.6.2, "Working with masks using the dialog", on page 342.
3. Set up the test actions. See Section 12.1.6.3, "Setting up a mask test", on page 343.
4. Select "Menu" > "Acquisition" > "History".
5. Activate "Show history".
6. Select "Play".

## 12.2 Frequency response and impedance analyses (options R&S MXO3-K36/K337)

**Frequency response analysis**

The frequency response analysis (FRA) option lets you perform low-frequency response analysis on your oscilloscope. It characterizes the frequency response of various electronic devices, including passive filters and amplifier circuits. For switching power supplies, it measures the control loop response (CLR) and power supply rejection ratio (PSRR).

For more basic information, see Understanding Bode plots.

**Impedance analysis**

With the impedance analysis option, you can measure the impedance of, for example, power distribution networks (PDN) over frequency. The analysis helps to improve the power integrity, reduce EMI, characterize components, and much more.

### General notes for both options

Both options use the oscilloscope's built-in waveform generator (option R&S MXO3-B6) to create stimulus signals ranging from 10 mHz to 100 MHz. The stimulus signal and the output signal of the DUT are measured at each test frequency. The oscilloscope calculates the gain, phase and impedance and plots the resulting curves in the plot diagram (Bode plot).

Seeing both time and frequency domain signals in parallel, you can monitor if the injected signal causes distortion that leads to errors in the measurement.

The setup of both options is very similar. They use the same measurement mode and dialog. Most settings are common for both options. Restrictions to one of the options are mentioned in the descriptions.

## 12.2.1 Results and plot of analysis

### FRA window

The FRA window has several sections. They are mainly the same for frequency response analysis and impedance analysis. Frequency response analysis delivers the results for gain and phase, and impedance analysis the impedance. Margin values are only available for frequency response analysis.




*Figure 12-2: Display of frequency response analysis*

1 = Toolbar
2 = Progress bar
3 = Plot results table
4 = Marker value table
5 = Margin value table
6 = Exit application
7 = Inactive waveforms
8 = Active waveforms and plots
9 = Controls
10 = Plot diagram, gain: blue color; phase: red color; amplitude: green color; reference: white color
11 = Channel diagram




***Figure 12-3: Display of impedance analysis (shunt through measurement): same sections as for frequency response analysis, except for margin value table***

#### 12.2.1.1 Progress bar

Displays the current status of the measurement: how many points have already been displayed and the total set number of points.

#### 12.2.1.2 Plot diagram

The FRA plot diagram has the frequency presented on the x-axis. The gain or impedance (blue color waveform in the image) and phase (red color waveform) scales are on the y-axis. The gain plot represents the ratio between input and output. The phase plot shows the phase shift between input and output.

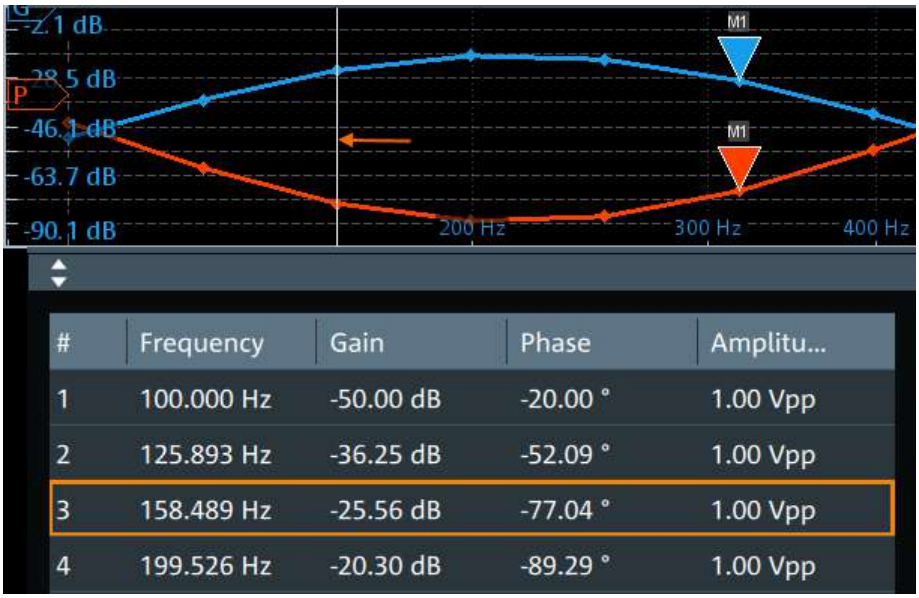
For better visibility and distinguishability, FRA waveforms and their reference waveforms have their own color category in "Settings" > "Appearance" > "Colors" > "Cate-

gory". You can change the color assignment and assign another color or a color table to FRA waveforms. See Section 5.3.1, "Colors", on page 96 for details.

#### 12.2.1.3 Plot results table

In the plot results table for frequency response analysis, all sample points are displayed with their values for the frequency, gain, phase and amplitude. The plot results table for impedance analysis lists the sample points with frequncy, impedance and amplitude values.

If you select a sample in the table, the respective points in the FRA plot are highlighted by a white vertical helper line on the gain and the phase curve.




| # | Frequency | Gain | Phase | Amplitu... |
|---|---|---|---|---|
| 1 | 100.000 Hz | -50.00 dB | -20.00 ° | 1.00 Vpp |
| 2 | 125.893 Hz | -36.25 dB | -52.09 ° | 1.00 Vpp |
| 3 | 158.489 Hz | -25.56 dB | -77.04 ° | 1.00 Vpp |
| 4 | 199.526 Hz | -20.30 dB | -89.29 ° | 1.00 Vpp |

#### 12.2.1.4 Marker value table

There are two markers available for the plot, which visualize the points of the margin measurement in frequency response analysis. They are highlighted on the plot diagram by a white line and the marker number 1 or 2. You can move the markers as needed.

In the marker table you can see the frequency, phase and gain (or impedance) values for both markers.

Remote commands:

- `FRANalysis:MARKer<m>:FREQuency` on page 1238
- `FRANalysis:MARKer<m>:GAIN?` on page 1238
- `FRANalysis:MARKer<m>:PHASe?` on page 1238

#### 12.2.1.5 Margin value table

The margin value displays the values of gain margin and the phase margin and their frequencies. Higher margin values are an indicator for higher stability of the system.

- Gain margin is measured at the frequency where the phase shift equals -180°. Gain margin indicates the distance from the measured gain to a gain of 0 dB.
- Phase margin is measured at the frequency where the gain equals 0 dB (also known as crossover frequency). Phase margin indicates the distance from the measured phase to a phase shift of -180°.

The margin value table is only available for frequency response analysis.

Remote commands:

- `FRANalysis:MARGin:STATe` on page 1237
- `FRANalysis:MARGin:GAIN:FREQuency?` on page 1237
- `FRANalysis:MARGin:GAIN:VALue?` on page 1237
- `FRANalysis:MARGin:PHASe:FREQuency?` on page 1237
- `FRANalysis:MARGin:PHASe:VALue?` on page 1237

#### 12.2.1.6 Vertical position and size of the curves

To set the position and the vertical scaling of the gain, phase, amplitude and impedance waveforms:

► Select the waveform and use the vertical [Scale] and [Position] (upper knob) knobs.
Or: Use the settings in the "Scale" tab.

The remote commands for setting the position and scales of the are described in Section 18.17.2.4, "Diagram settings", on page 1232.

### 12.2.2 Using frequency response and impedance analysis

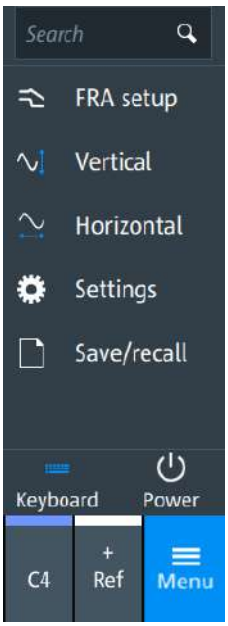
**Connecting the test setup**

To avoid measurement uncertainties, make sure that the cables connecting the input and the output of your DUT to the oscilloscope are of similar length.

**Starting the analysis**

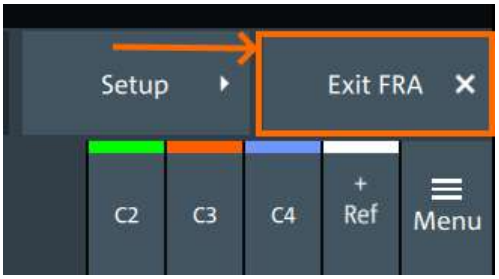
1. Press the [Apps] key.
2. In the "General" tab, tap "FRA" or "Impedance".

   The FRA window opens. Only the settings relevant for the FRA setup are included in the "Menu". The menu provides access to the following dialogs: "Setup", "Vertical", "Horizontal", "Settings", "Save/recall".




**Closing the FRA window**

► Tap the "Exit FRA" button at the bottom-right corner of the display.




### 12.2.3 Dependencies of the measurement setup

When performing the analysis, consider the following dependencies of vertical parameters.

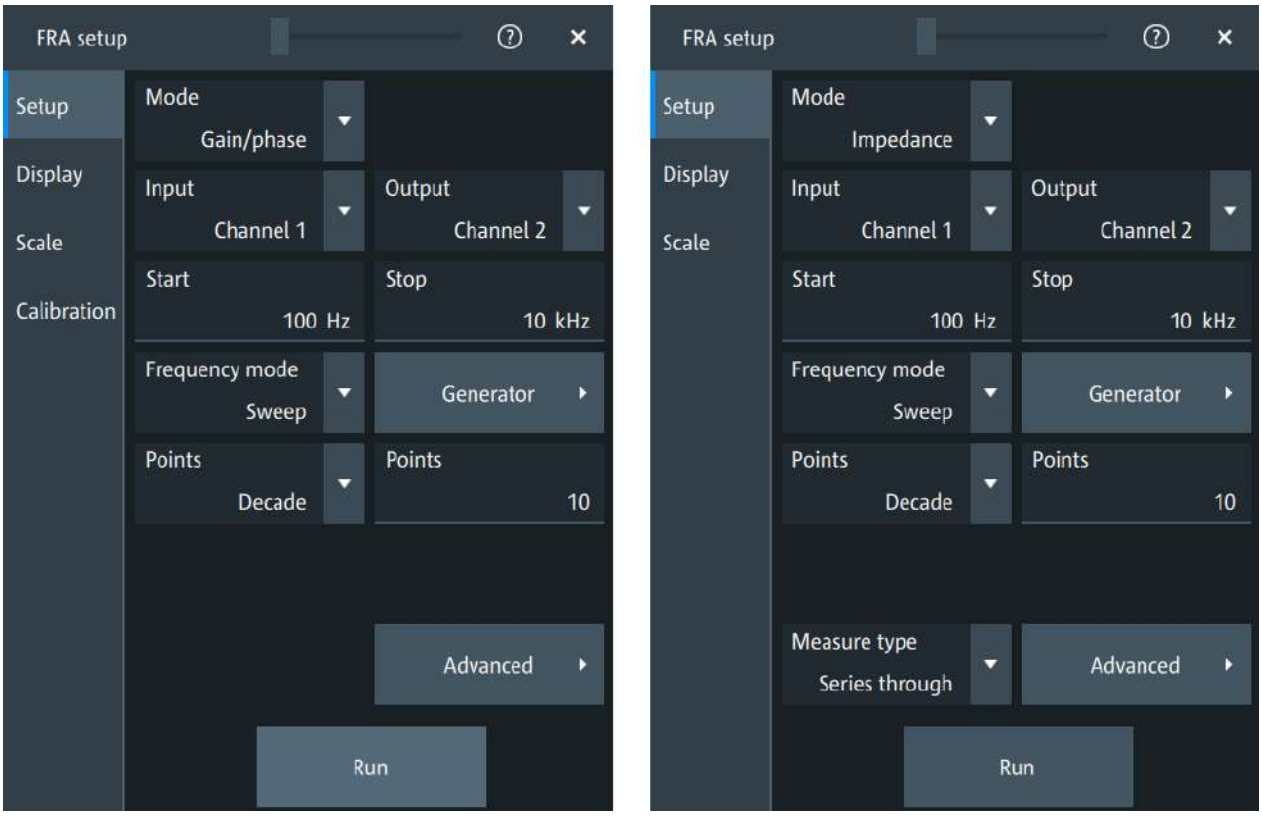
- Coupling:
  - For frequencies smaller than 10 Hz, the "Coupling" on page 137 is set to DC.
  - For frequencies higher than 10 Hz, you can select a Coupling manually. The default value is AC.
- The bandwidth value is changed to a higher value, if the selected bandwidth is smaller than double the stop frequency.
- If the bandwidth of the connected probe is lower than the "Stop" frequency, a warning is shown.

### 12.2.4 Settings for frequency response and impedance analysis

The following sections describe the settings that you can define for the frequency response analysis and impedance analysis.

#### 12.2.4.1 Setup

Access: [Apps] > "General" > "FRA" or "Impedance" > "Setup".




The "Setup" tab provides the analysis settings and access to the generator setup and advanced settings.

**Mode**
Selects the application mode.

- "Gain/phase": Frequency response analysis (option R&S MXO3-K36)
- "Impedance": Impedance analysis (option R&S MXO3-K337)

The mode is set according to your selection in the "Apps" menu. If you have obtained both options, you can change the analysis mode directly in the "Setup" dialog.

Remote command:
`FRANalysis:MODE` on page 1227

**Input**
Sets the channel for the input signal of the DUT.

Remote command:
FRANalysis:INPut[:SOURce] on page 1225

**Output**
Sets the channel for the output signal of the DUT.

Remote command:
FRANalysis:OUTPut[:SOURce] on page 1227

**Start, Stop**
Set the start and stop frequency of the sweep.

Remote command:
FRANalysis:FREQuency:STARt on page 1224
FRANalysis:FREQuency:STOP on page 1224

**Frequency mode**
Selects between single frequency and sweep analysis.

| | |
|---|---|
| "Sweep" | Set the start and stop frequency, the points measurement mode and the number of points. |
| "Single" | Set the signal frequency in "Start", and the number of points. The points define the scale of the horizontal axis. |

Remote command:
FRANalysis:FREQuency:MODE on page 1224

**Generator**
See Section 12.2.4.2, "Generator settings", on page 352.

**Points**
Selects if the number of points is measured as total or per decade and sets the number of points.

Remote command:
FRANalysis:POINts:TOTal on page 1228
FRANalysis:POINts:MODE on page 1228
FRANalysis:POINts:LOGarithmic on page 1228

**Measure type**
Selects the measurement method for the impedance analysis. Select the measurement method according to the impedance magnitude of your DUT.

| | |
|---|---|
| "Shunt through" | The shunt through method can measure very low impedance. |
| "Series through" | The series-through method is most effective when measuring high impedance values: the 10% accuracy range is about 5 Ω to 20 kΩ or roughly one decade higher than the reflection method. |
| "Voltage current" | This setup is very flexible and especially suitable to measure input impedance and output impedance of active circuits, such as DC/DC converter. |

Remote command:
FRANalysis:IMPedance[:MODE] on page 1225

**Run**
Starts the frequency response analysis.

Remote command:
FRANalysis:STATe on page 1229

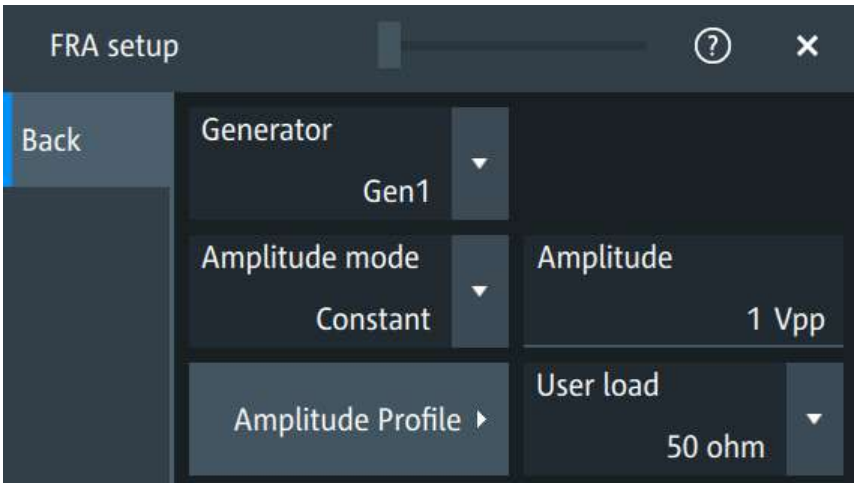
**Repeat**
Repeats the measurement, using the same parameters.

Only available in the FRA control menu.

Remote command:
FRANalysis:REPeat on page 1229

### 12.2.4.2 Generator settings

Access: [Apps] > "General" > "FRA" > "Setup" > "Generator".




The dialog provides the settings for the generator.

**Generator**
Selects the built-in generator to start a frequency sweep for a defined frequency range.

Remote command:
FRANalysis:GENerator[:CHANnel] on page 1230

**Amplitude mode**
Selects, if the amplitude is a constant value ("Amplitude") or is defined as an amplitude profile.

If the amplitude profile is selected, a green-colored waveform of the amplitudes for the different frequencies is displayed in the analysis plot, and scale settings for this waveform are available.

Remote command:
FRANalysis:AMPLitude:MODE on page 1229
FRANalysis:AMPLitude:ENABle on page 1233

**Amplitude**
Sets a fixed amplitude for the generated signal.

Remote command:
FRANalysis:GENerator:AMPLitude on page 1230

**Amplitude profile**
Opens a dialog to set the amplitude profile. See Section 12.2.4.3, "Amplitude profile", on page 353.

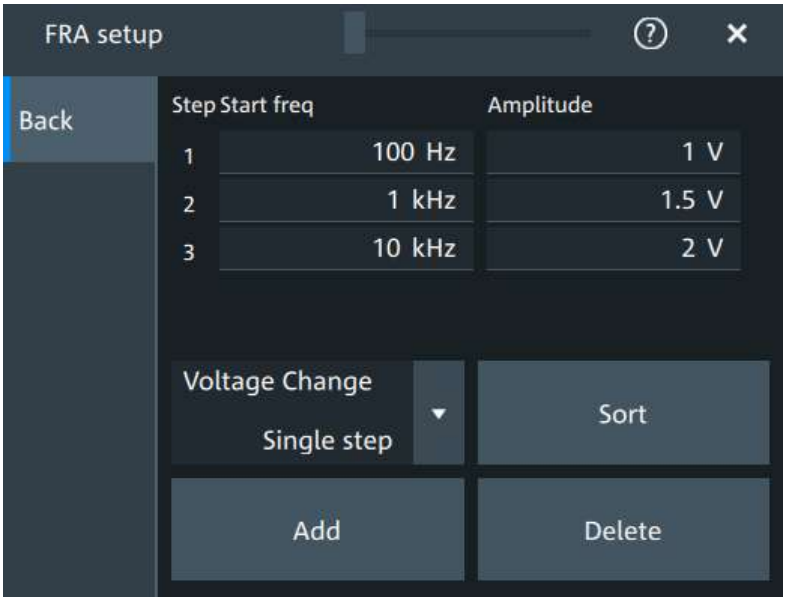
You can define specific amplitudes for different frequencies. The amplitude profile is useful when testing sensitive circuits, where the amplitude gets too high. In this case, distortion can occur.

**User load**
Selects the generator voltage display for 50Ω or high impedance load.

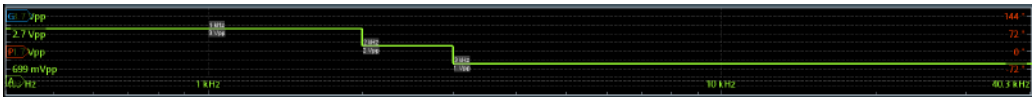
Remote command:
FRANalysis:GENerator:LOAD on page 1230

### 12.2.4.3 Amplitude profile

Access: [Apps] > "General" > "FRA" > "Setup" > "Generator" > "Amplitude profile".




Opens a dialog to set the amplitude profile. You can then define different amplitudes for different frequency, and also the number of points (steps) in the profile. The amplitude profile is useful when testing sensitive circuits, where the amplitude gets too high. In this case distortion can occur.



**Step start freq, Amplitude**
Set the frequency and amplitude values for the selected step.

Remote command:
`FRANalysis:AMPLitude:PROFile:POINt<m>:AMPLitude` on page 1231
`FRANalysis:AMPLitude:PROFile:POINt<m>:FREQuency` on page 1231

**Voltage change**
Selects if the voltage change is done as a single step or as a ramp.

Remote command:
`FRANalysis:AMPLitude:PROFile:MODE` on page 1231

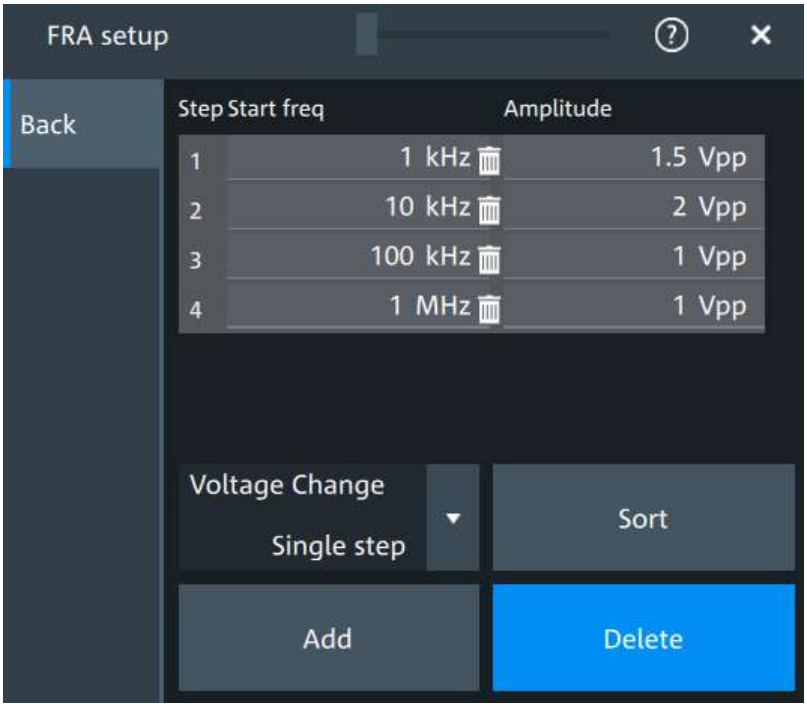
**Add**
Adds a new step to the amplitude profile.

Remote command:
`FRANalysis:AMPLitude:PROFile:APOint` on page 1231
`FRANalysis:AMPLitude:PROFile:COUNt` on page 1231

**Sort**
Sorts the steps in the amplitude table by frequency, starting with the lowest frequency.

Remote command:
`FRANalysis:AMPLitude:PROFile:SORT` on page 1232

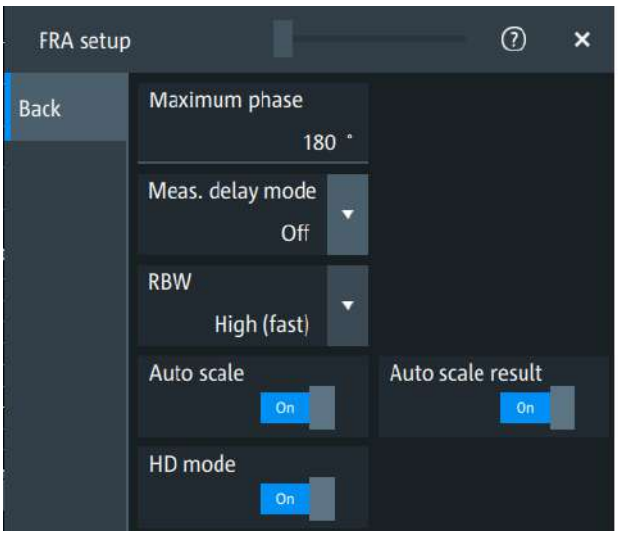
**Delete**
If enabled, a delete icon appears next to each step in the amplitude profile table. Tap the icon to delete the step.

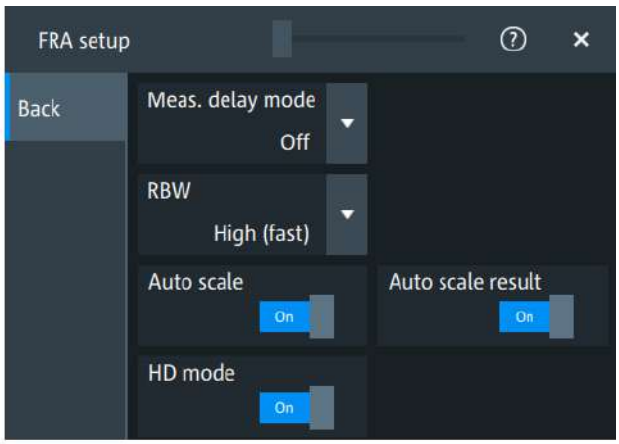



Remote command:
`FRANalysis:AMPLitude:PROFile:POINt<m>:REMove` on page 1232

### 12.2.4.4 Advanced

Access: [Apps] > "General" > "FRA" > "Setup" > "Advanced".




**Maximum phase**
Sets the upper boundary of the vertical phase window.

The lower boundary is given by "Maximum phase" - 360°.

By default, the "Maximum phase" is set to 180° for a phase window ranging from -180° to 180° accordingly.

Only available for frequency response analysis.

Remote command:
FRANalysis:PHASe:MAXimum on page 1227

**Meas. delay mode, Delay time, Delay period**
Selects the type of delay that the system waits before measuring the next point of the plot. Delay is helpful in systems that need more time to adapt to the new frequency, for example if filters with significant time group delays are present. You can select a time delay and set the "Delay time", or a period delay and set the "Delay period".

Remote command:
FRANalysis:MEASurement:DELay:STATe on page 1225
FRANalysis:MEASurement:DELay:MODE on page 1225
FRANalysis:MEASurement:DELay[:TIME] on page 1226
FRANalysis:MEASurement:DELay:PERiod[:NUMBer] on page 1226

**RBW**
Sets the resolution bandwidth, which determines the number of measurements that are used for creating the plot.

| | |
|---|---|
| "High (fast)" | A high RBW value is useful for obtaining a general understanding of a system's behavior over a wide frequency range. The accuracy of the measurement is not as good, but the measurement is completed fast. |
| "Medium" | A medium RBW value is a compromise between the measurement time and measurement accuracy. |
| "Low (slow)" | A low RBW value allows for a more detailed analysis of the system. Since more data needs to be collected, the measurement time can increase significantly. |

Remote command:
FRANalysis:MEASurement:RBW on page 1227

**Auto scale**
If enabled, the vertical scaling of input and output channel is done automatically for every new measurement frequency point.

That guarantees a non-disturbed signal and a maximum of measurement resolution.

For Coupling = "DC", the channel offset is also changed to compensate the signal offset.

The function is disabled if calibration is active.

Remote command:
FRANalysis:AUToscale on page 1223

**Auto scale result**
If enabled, the scaling of the plot diagram is adjusted when new measurement results are determined. Thus, all values are included in the plot.

Remote command:
FRANalysis:AUToscale:RESult[:ENABle] on page 1223

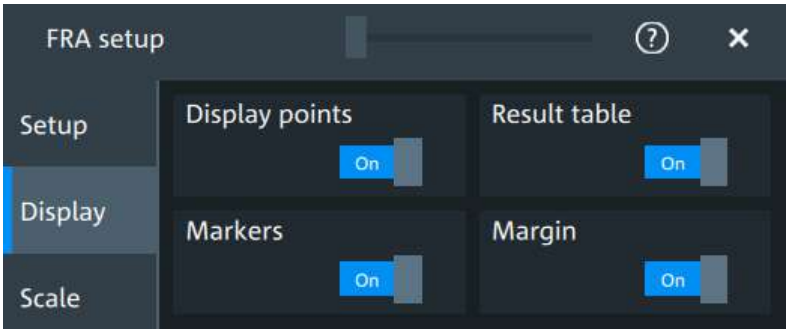
**HD mode**
Disables the HD mode, which is active by default. In particular, disable the HD mode if you analyze switching peaks.

The frequency in HD mode must be at least 3 times the current frequency. The bandwidth filter of the channel is adjusted automatically to the generator frequency.

Remote command:
`FRANalysis:HDEFinition[:ENABle]` on page 1224

### 12.2.4.5 Display

Access: [Apps] > "General" > "FRA" > "Display".




In this dialog, you can select which elements are displayed in the FRA diagram.

**Display points**
Enables the display of the measurement points for the frequency response analysis.

Remote command:
`FRANalysis:MEASurement:POINt[:DISPlay]` on page 1226

**Result table**
Enables the display of the result table for the FRA.

Remote command:
`FRANalysis:RESult:STATe` on page 1240

**Markers**
Activates the display of the marker table and the markers in the plot.

See also: Section 12.2.1.4, "Marker value table", on page 347.

Remote command:
`FRANalysis:MARKer<m>:STATe` on page 1240

**Margin**
Enables the display of the margin table for the frequency response analysis.

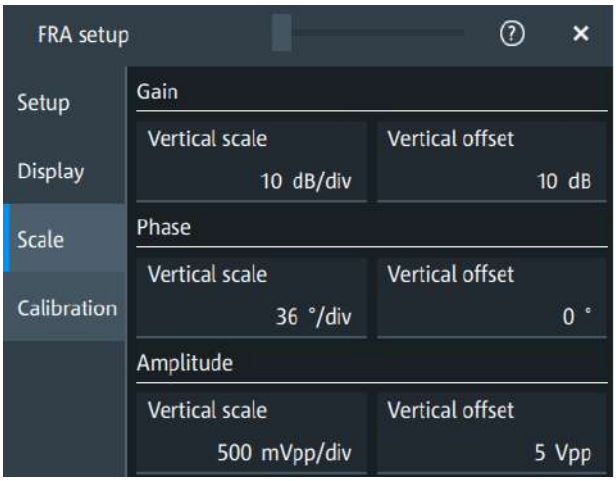
Available only for frequency response analysis. See also: Section 12.2.1.5, "Margin value table", on page 348.

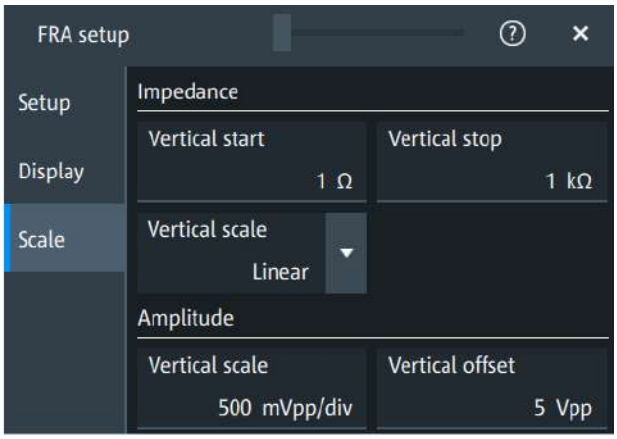
Remote command:
`FRANalysis:MARGin:STATe` on page 1237

### 12.2.4.6 Scale

Access: [Apps] > "General" > "FRA" > "Scale".

The settings are different for frequency response analysis and impedance analysis.




**Vertical scale**
Sets the scale of the y-axis in the function diagram for gain, phase and amplitude.

Remote command:
`FRANalysis:PHASe:SCALe` on page 1235
`FRANalysis:GAIN:SCALe` on page 1234

**Vertical offset**
Sets a voltage offset to adjust the vertical position on the screen for gain, phase and amplitude.

Remote command:
`FRANalysis:PHASe:OFFSet` on page 1235
`FRANalysis:GAIN:OFFSet` on page 1233

**Vertical start, Vertical stop**
Set the lower and upper value of the vertical scale for the impedance trace.

Remote command:
`FRANalysis:IMPedance:STARt` on page 1234
`FRANalysis:IMPedance:STOP` on page 1234

**Vertical scale**
Sets the scale mode for the impedance trace: linear or logarithmic.

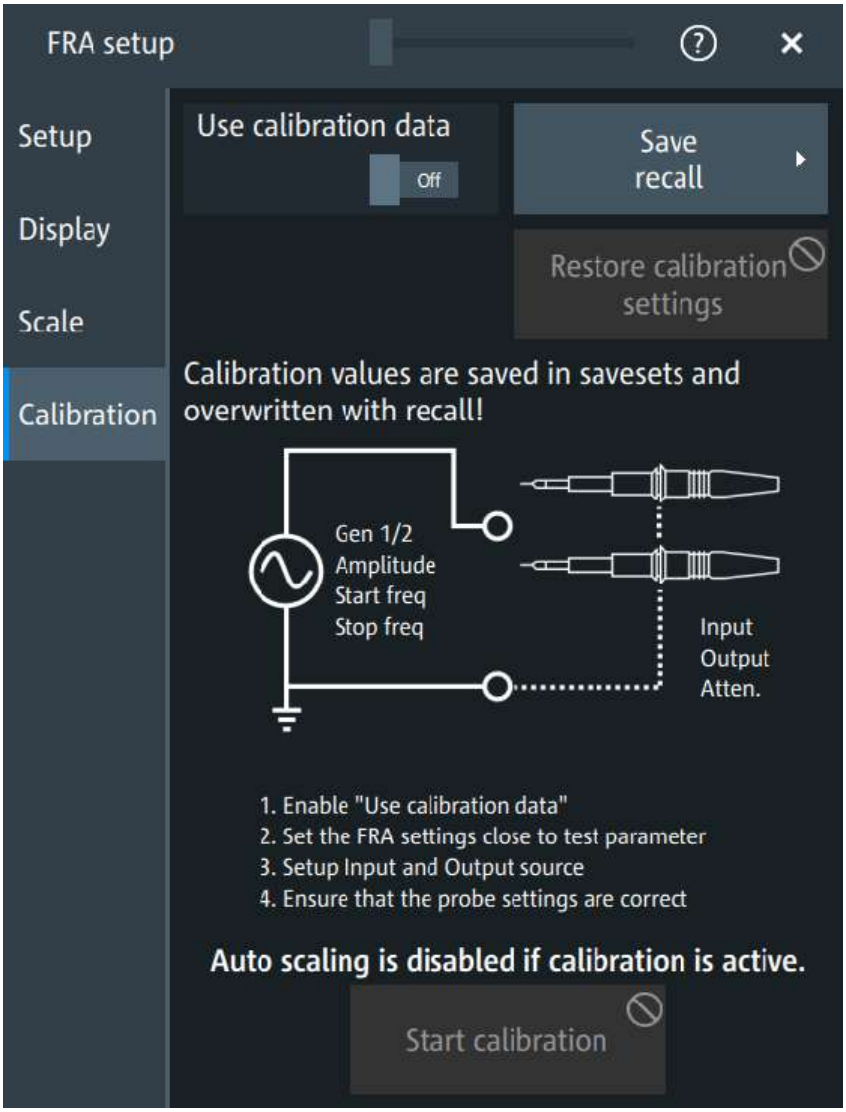
Remote command:
`FRANalysis:IMPedance:SCALe` on page 1234

### 12.2.4.7 Calibration

Calibration is used to compensate the electrical parameter of probes, cables, connectors or fixtures.

Access: [Apps] > "General" > "FRA" > "Calibration".

Calibration is only available for frequency response analysis.




**Test setup considerations**

Example of a calibration test setup shows an example of a test setup, where the input probe is connected to [C1] and the output probe is connected to [C2]. The input is also connected to the waveform generator.

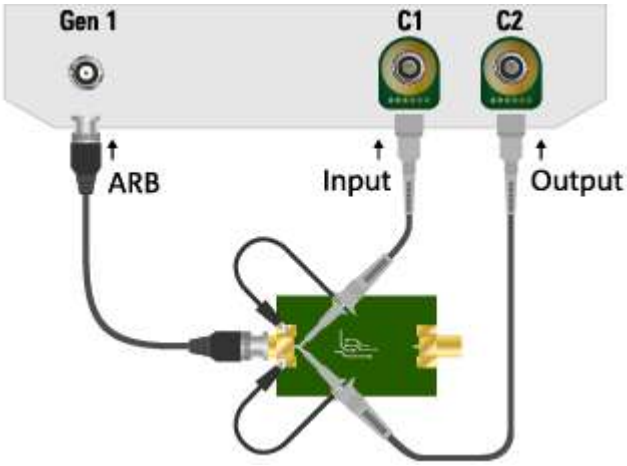



*Figure 12-4: Example of a calibration test setup*

Input = [C1], [Gen1]
Output = [C2]

Consider also the following:

- Calibration for FRA is only possible with probes connected to the input and output channels.
- For best FRA calibration results, shorten the DUT and place the probe tips as near as possible to each other.
- If there is a fixture for the DUT, include it in the calibration setup.
- If the DUT needs a termination, conduct it with an external termination directly on the DUT output or input.

### Calibration state

Once the calibration is started, a label is shown in the toolbar.




The calibration label displays the status of the calibration. The following states are available:

- "Valid": calibration values are used
- "Invalid": device settings changed
- "Running": a calibration cycle is running
- "None": not performed or no data

### Settings influencing calibration

Calibration is only valid for the settings that are active during its execution. If one of the following settings is changed, the calibration becomes "Invalid":

- "Input" and "Output" channel
- "Channel coupling" and "Bandwidth"
- "Channel scaling"

- "Generator"
- "User load"
- "Amplitude mode"
- "Amplitude profile"
- "Amplitude profile" > "Voltage change"
- "Start" frequency
- "Stop" frequency
- "Points" mode and total points
- "Meas. delay mode"
- "Delay time"
- "RBW"

If calibration is active, "Auto scale" is disabled.

**To perform a calibration**

1. Enable "Use calibration data".
2. Set the FRA settings close to the test parameters.
3. Select the Input and Output source.
4. Ensure that the probe settings are correct.

The following settings are available for the calibration:

**Use calibration data**
If enabled, the user calibration data is used for the frequency response analysis.

Remote command:
`FRANalysis:CALibration:STATe` on page 1236

**Save/recall**
Opens the "Save/recall" dialog. You can save the instrument setting in a saveset, and load an existing saveset. For details, see Section 13.3, "General and measurement settings: savesets", on page 451.

The saveset includes the calibration data. If you load a saveset, the current calibration values are overwritten.

**Restore calibration settings**
Restores the FRA settings to the values that were selected during the calibration process. Restore allows you to repeat measurements with a valid calibration state. See also "Settings influencing calibration" on page 360.

To keep and restore calibration settings in remote control, set up and calibrate the frequency analysis, and save the instrument settings in a saveset. When you load this saveset later, the calibration data is restored together with the instrument settings.

**Start calibration**
Starts the calibration. The button is only enabled if there is a valid test setup.

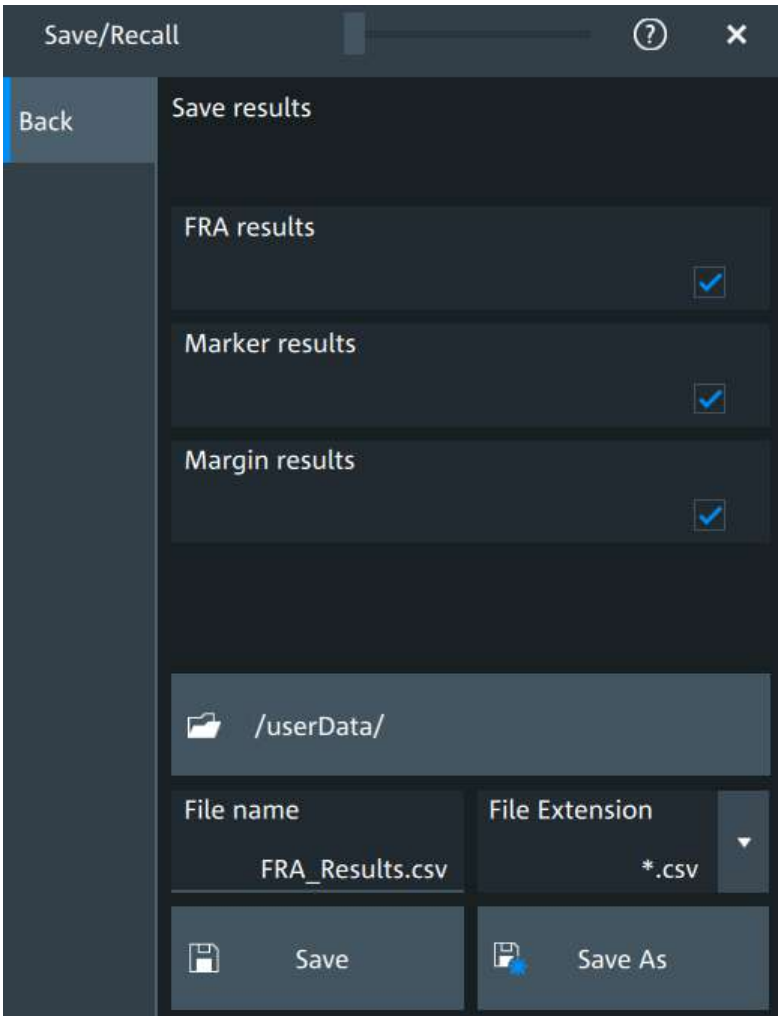
The state of the calibration process is displayed in the calibration label in the toolbar.

After a calibration is performed successfully, the calibration values are saved automatically.

Remote command:
`FRANalysis:CALibration:CALibration` on page 1236
`FRANalysis:CALibration:RESult?` on page 1236

### 12.2.5 Saving analysis results

Access: "Menu" > "Save/recall" > "Results".




**FRA results**
In this dialog, you can select which result values you want to include in your FRA export results file.

The following options are available:
- "FRA results": includes the frequency, gain, phase and amplitude
- "Marker results": includes the marker frequency and gain

- "Margin results": includes the margin gain and phase

Remote command:
`EXPort:RESult:SELect:FRA:MARGin` on page 1245
`EXPort:RESult:SELect:FRA:MARKer` on page 1245
`EXPort:RESult:SELect:FRA:RESult` on page 1245

## 12.2.6 Reference waveforms of FRA curves

To compare the results of frequency resonse analysis and analyze differences between them, you can use up to 4 specific reference waveforms. These references can be created from gain and phase curves, and from other references. The type of the source waveform is included in the reference waveform data.

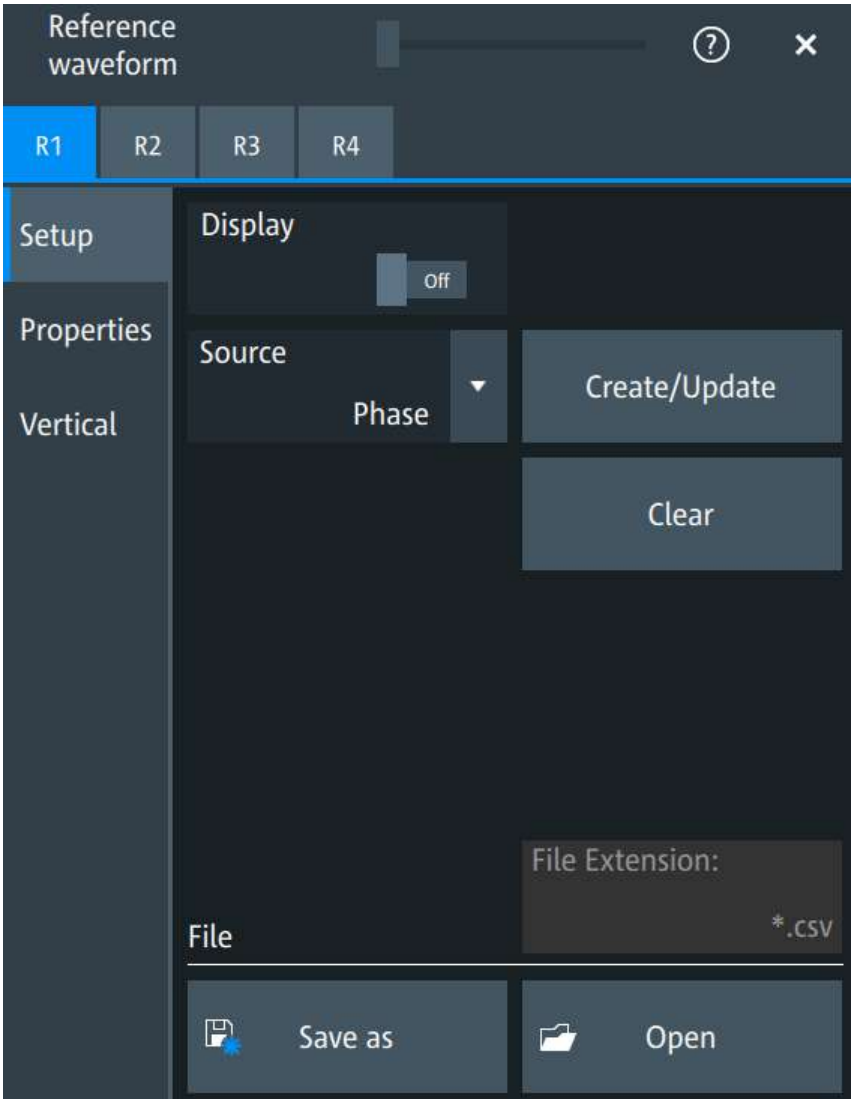
You can save an unlimited number of reference waveforms and load them for further use. Markers are also set on these reference waveforms.

The display of a reference waveform is independent from the display of the source waveform; you can move, stretch and compress the curve vertically. A loaded reference waveform is restored when you leave the FRA mode and enter it again with no preset inbetween. Reference waveforms are aligned horizontally to the actual start and stop position in the waveform diagram.

When a FRA reference waveform is selected, you can change scale and position using the knobs on the front panel.

### 12.2.6.1 Reference waveform setup

Access: perform a frequency response analysis and display resulting waveforms > select the "+ Ref" signal activator > "Setup" > select reference waveform




In the "Setup" tab, you select the source of the reference waveform and create the reference.

**Display**
Enables the display of the reference waveform in the diagram. Before you can display it, create the reference waveform.

Remote command:
`FRANalysis:REFCurve<rc>:ENABle` on page 1241

**Source**
Selects the source waveform of the reference from the existing frequency response waveforms, e.g. gain, phase or an other reference.

Remote command:
`FRANalysis:REFCurve<rc>:SOURce` on page 1243

**Create/Update**
Copies the selected source waveform with all its settings to the memory of the reference waveform. If there is a previously defined reference waveform in this memory, it is updated by the current source waveform.

A progress bar informs you about the process.

Remote command:
`FRANalysis:REFCurve<rc>:UPDate` on page 1244

**Clear**
Deletes the selected reference waveform. It disappears from the display, and its memory is deleted.

Remote command:
`FRANalysis:REFCurve<rc>:CLEar` on page 1241

**File extension**
Shows the file format. Currently, only CSV files are supported.

**Save as**
Opens a file explorer where you select the folder, enter the filename and save the file. The complete reference waveform is stored in a CSV file and can be reloaded later.

A progress bar informs you about the process.

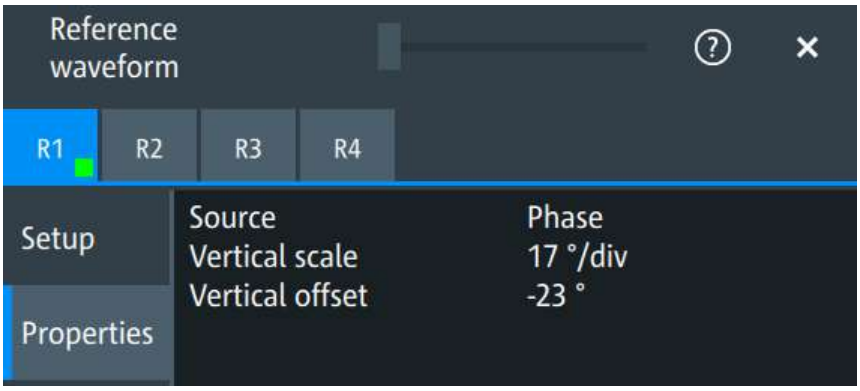
Remote command:
`FRANalysis:REFCurve<rc>:NAME` on page 1242
`FRANalysis:REFCurve<rc>:SAVE` on page 1243

**Open**
Opens a file selection dialog box and loads the selected reference file.

Remote command:
`FRANalysis:REFCurve<rc>:OPEN` on page 1242

### 12.2.6.2 Reference waveform properties

Access: perform a frequency response analysis and display resulting waveforms > select the "+ Ref" signal activator > "Properties" tab > select reference waveform



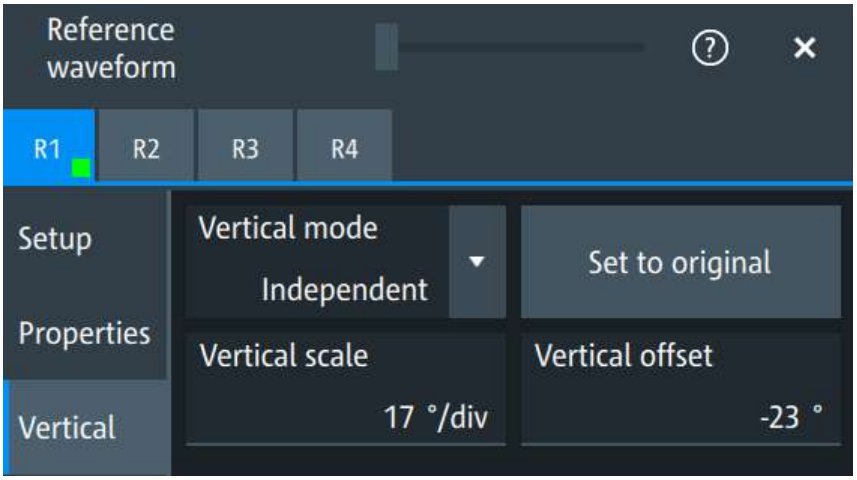
A FRA reference waveform can be vertically scaled and positioned in the diagram. The "Properties" tab shows the original settings of the reference waveform, which are stored together with the waveform data.

The following settings are displayed:

- Type of the source waveform
- Vertical scale and offset that are defined in the "Scale" tab (Section 12.2.4.6, "Scale", on page 358)

### 12.2.6.3 Vertical settings for reference waveforms

Access: perform a frequency response analysis and display resulting waveforms > select the "+ Ref" signal activator > "Vertical" tab > select reference waveform




**Vertical mode**
Selects the type of vertical settings:

"Coupled to source" Vertical position and scale of the source are used.

"Independent" Scaling and position can be set specific to the reference waveform.

Remote command:
`FRANalysis:REFCurve<rc>:VMODe` on page 1244

**Set to original**
Available, if "Vertical mode" = "Independent".

Restores the original vertical settings of the reference waveform.

Remote command:
`FRANalysis:REFCurve<rc>:TOORiginal` on page 1243

**Vertical scale**
Available, if "Vertical mode" = "Independent".

Sets the vertical scale, which defines the displayed amplitude of the selected waveform.

The unit of the reference is the same as the unit of the source waveform.

Remote command:
`FRANalysis:REFCurve<rc>:SCALe` on page 1243

**Vertical offset**
The vertical offset moves the reference waveform vertically. Enter a value with the unit of the waveform.

Remote command:
`FRANalysis:REFCurve<rc>:OFFSet` on page 1242

## 12.2.7 Example of impedance analysis

The following screenshots show a series through measurement of a lowpass filter that has its maximum attenuation near 5 MHz. The logarithmic scale is used for the impedance curve.




*Figure 12-5: Initial impedance measurement*




*Figure 12-6: Reference waveform was created from the impedance curve*




*Figure 12-7: New impedance measurement around 5 MHz to analyze the peak, with the reference waveform of the previous measurement*

# 12.3 Power analysis (option R&S MXO3-K31)

With the MXO 3 and option R&S MXO3-K31, you can perform power analysis measurements.

For best measurement results, we recommended setting channel Bandwidth = 20 MHz.
If using the HD mode, we recommended setting Bandwidth = 100 MHz.

## 12.3.1 Power analysis initial settings

In the main "Power analysis" dialog, you select the required power measurements, align the channels and set the offset compensation to get correct measurements.

### 12.3.1.1 Select power measurements

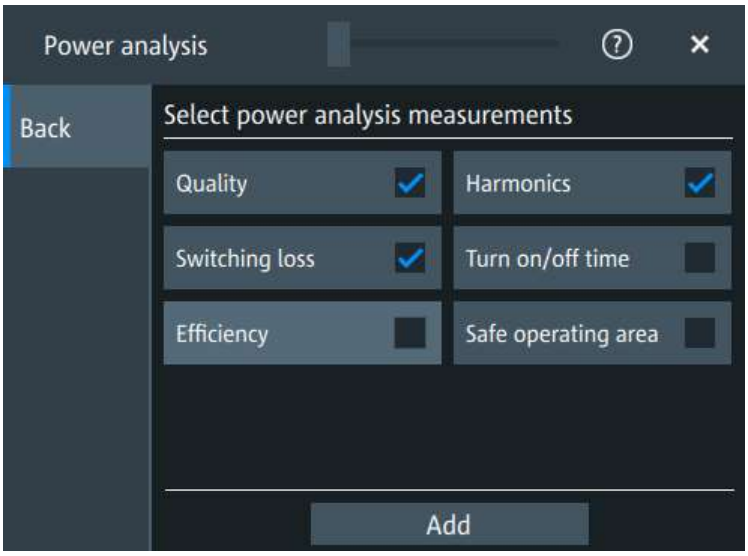
In the "Analysis" tab, you select the power analysis measurements.

You can add up to 3 power analysis measurements, including several instances of the same measurement.

From here, you can access the settings for the enabled power analysis measurements.

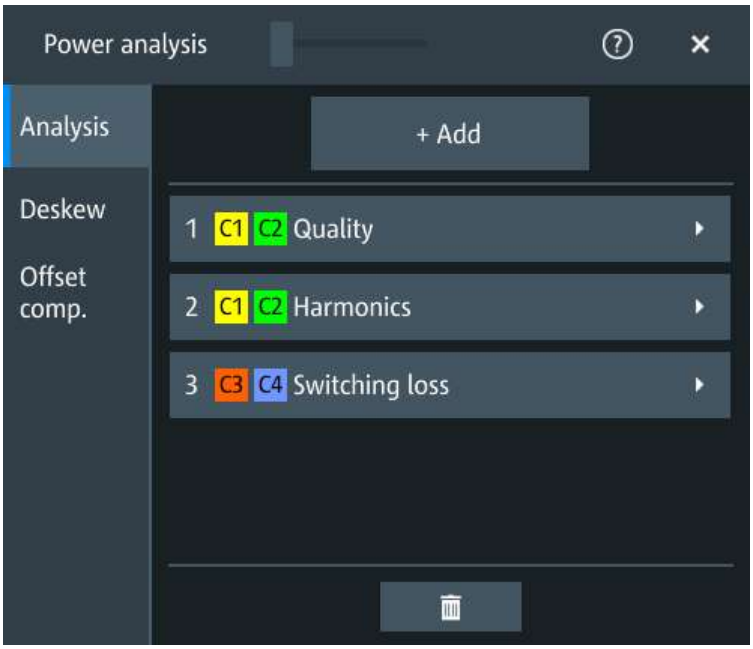
**To add a new power measurement**

1. Select "Menu" > "Apps" > "General" tab > "Power".
2. In the "Analysis" tab, select "+ Add".




3. Select a power measurement on the list to enable it.
4. Select "+ Add" to add the selected measurements.

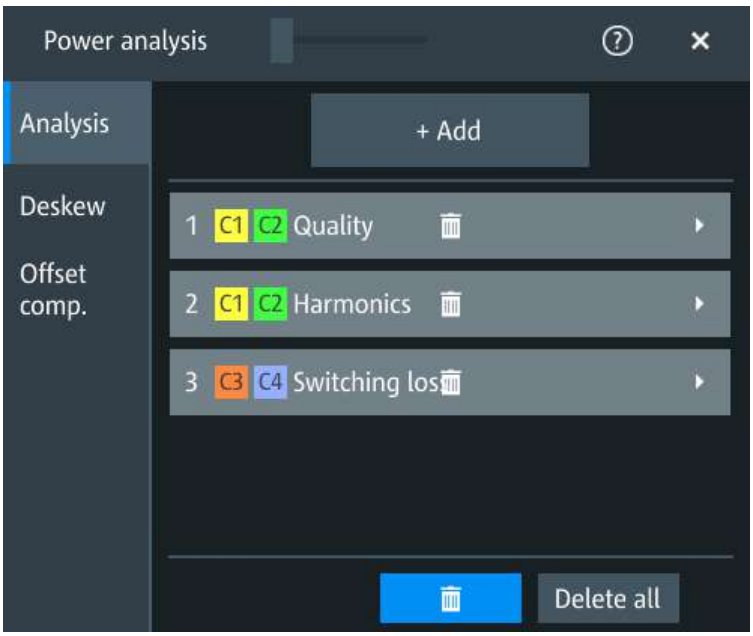
All selected measurements are enabled.




**To delete a measurement**

1. Select the "Menu" > "Apps" > "General" tab > "Power", to open the power measurement dialog.
2. In the "Analysis" tab, select .

   A delete icon appears in the selection button of each measurement.




3. Select the power measurement that you want to delete. Alternatively, select "Delete All" to delete all measurements.

**+ Add**
Opens a dialog to select the power analysis measurements. See "To add a new power measurement" on page 369.

Remote command:
`POWer<m>[:ENABle]` on page 1248
`POWer<m>:TYPE` on page 1248

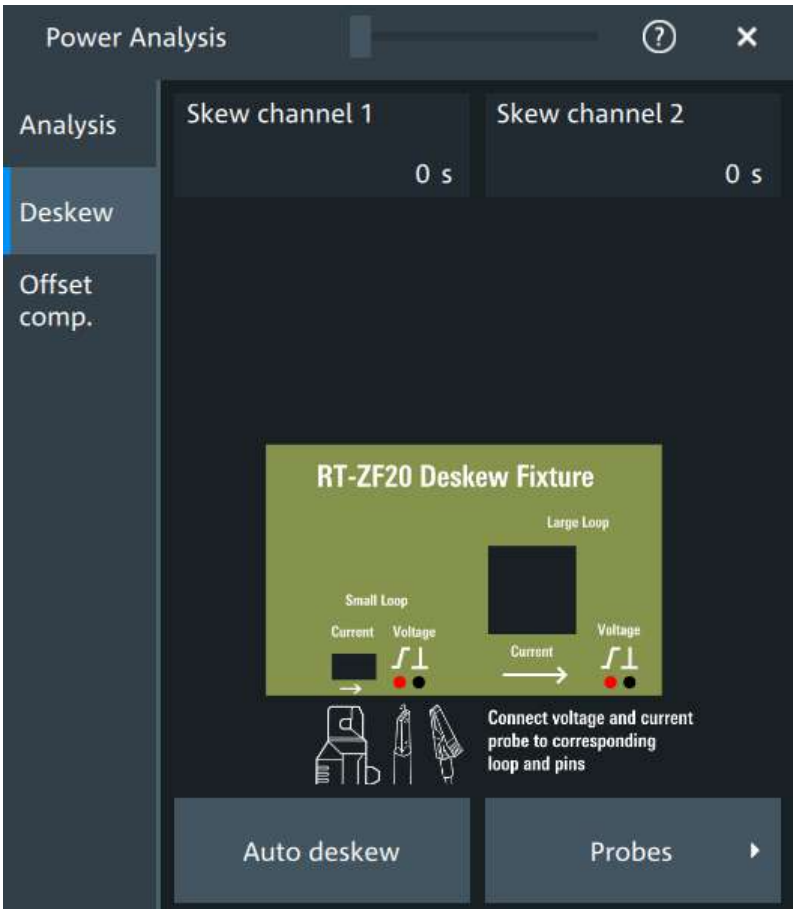
### 12.3.1.2 Deskew

In the dialog box you can perform a deskew of your current and voltage probes.

Required equipment:

- R&S RT-ZF20 power deskew fixture
- Rohde & Schwarz voltage probe
- Rohde & Schwarz current probe

#### Deskew settings




**Skew channel**
Sets user-defined skew settings for the channel connected to the current probe.

Alternatively, you can start an "Auto deskew".

Remote command:
CHANnel<ch>:SKEW:TIME on page 1002

**Auto deskew**
Starts an auto deskew. It is used to determine the trigger and scaling properties.

For an auto deskew, only Rohde & Schwarz probes are supported.

Make sure that the probes are configured correctly before you start the deskewing.

After the deskew, the result values are written in the "Skew" value of the corresponding channel.

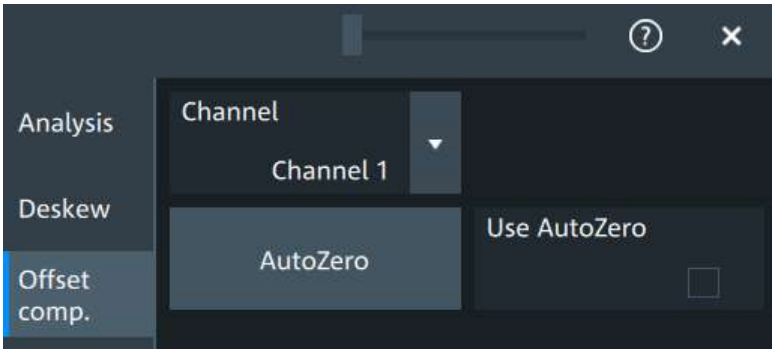
If there are more than two active channels, you can perform automatic deskew step by step and exchange the probes connected fixture between the steps.

Remote command:
POWer<m>:ASKew[:EXECute] on page 1249

**Probes**
Opens the vertical setup dialog, where you find the probe settings.

### 12.3.1.3 Offset compensation

The offset compensation adjustments are applied to all 3-phase power analysis measurements.




**Channel**
Select the channel for the offset compensation.

Remote command:
PROBe<ch>:SETup:OFFSet:ZADJust on page 1020

**AutoZero, Use AutoZero**
Voltage probes can have a zero error. The zero error is the voltage appearing at the probe output when nothing is connected to the probe. If a zero error occurs, it results in an external offset and the waveform is not displayed around 0 V.

To correct the zero error of voltage probes, short the signal pin and the ground pin together. Then select "AutoZero". While the alignment is running, the instrument switches to DC coupling to display the waveform correctly.

To correct the zero error, enable "Use AutoZero".

"AutoZero" uses the current scale and attenuation settings during alignment. If you change these settings, e.g. the vertical scale, repeat "AutoZero".

If a current probe is connected, the function demagnetizes the probe's sensor head and sets the waveform to zero position.

Remote command:
`PROBe<ch>:SETup:OFFSet:AZERo` on page 1008
`PROBe<ch>:SETup:OFFSet:USEautozero` on page 1008

## 12.3.2 Power quality

In an electric circuit power is a measure for the rate of flow of energy at a certain point of the circuit. The active power of a circuit is the portion of energy that is transferred in one direction over a complete cycle of the AC waveform. The active power is the energy that can be used for work. In AC circuits, however, inductive and capacitive elements can store energy temporarily. This portion of the power flow known as reactive power is then returned to the source without doing any work.

The power quality analysis measurements include the active power, the reactive power, the apparent power and the power factor. The crest factors and the phase angle between the current and voltage are also measured. These properties describe the power transfer in the system and allow you to characterize the power quality of the system.

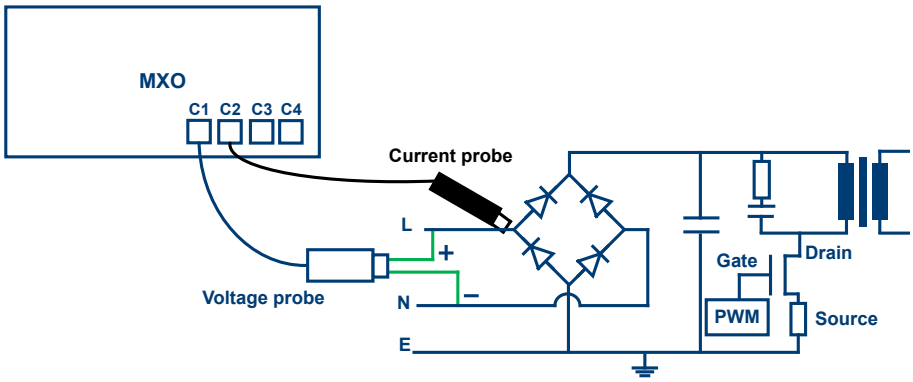
Required probes:

- Differential voltage probe
- Current probe

### 12.3.2.1 Configuring power quality

For details of the configuration settings, see Section 12.3.1.1, "Select power measurements", on page 369.

1. Add a new power quality measurement as described in "To add a new power measurement" on page 369.
2. Connect the differential voltage probe and the current probe to the oscilloscope.
3. Deskew the probes as described in Section 12.3.1.2, "Deskew", on page 371.
4. Connect the probes to the DUT:




a) Connect the positive (+) signal socket of the differential voltage probe to the line of the AC input.
b) Connect the negative (-) signal socket of the differential voltage probe to the neutral of the AC input.
c) Connect the current probe to the line of the AC input.

5. In the "Analysis" tab, tap the "Quality" measurement.

   The "Quality" setup dialog opens.

6. Select the correct channels for the "Voltage" and the "Current" probes.

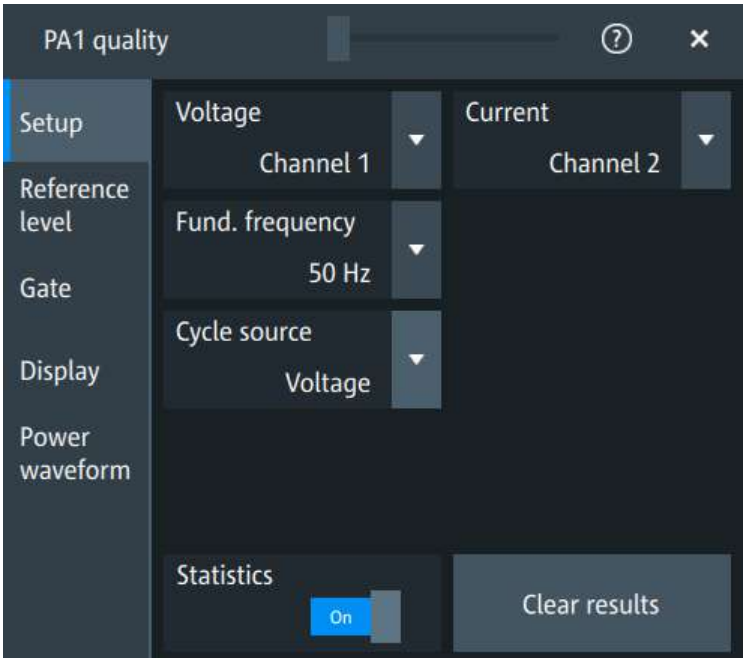
7. Set the "Fund. frequency" according to your signal.

8. If necessary, enable "Statistics".

   On the screen, you can see the measurement waveforms of the current, the voltage and the power. Also, the result table with numeric measurement results is shown. For details, see Section 12.3.2.7, "Power quality results", on page 378.

#### 12.3.2.2 Power quality settings

Access: "Menu" > "Apps" > "Power" > "+ Add" > "Quality" > select "Quality" > "Setup".




In the "Setup" tab, you configure the power quality measurement: select the sources, the fundamental frequency, and enable statistics.

**Voltage**
Selects the voltage source waveform. Analog channels and math waveforms can be used.

Remote command:
`POWer<m>:QUALity:SOURce[:VOLTage]` on page 1251

**Current**
Selects the current source waveform. Analog channels and math waveforms can be used.

Remote command:
`POWer<m>:QUALity:SOURce:CURRent` on page 1251

**Fund. frequency**
Selects the input frequency of the source signal.

Remote command:
`POWer<m>:QUALity:FREQuency[:VALue]` on page 1251
`POWer<m>:QUALity:FREQuency:USER` on page 1252

**Cycle source**
Selects the source of the period measurement. This period is the reference for related measurements and calculations.

Remote command:
`POWer<m>:QUALity:SOURce:CYCLe` on page 1251

**Statistics**
Activates or deactivates the statistical evaluation for the selected power measurement. If statistics are enabled, the following results are calculated:

| Label | Description |
|---|---|
| Current | Current value |
| Max | Maximum value |
| Min | Minimum value |
| Mean | Average |
| RMS | Root mean square |
| σ (S-dev) | Standard deviation |
| Event count | Number of measured events |
| Wave count | Number of waveforms (acquisitions) |

Remote command:
`POWer<m>:QUALity:STATistics[:ENABle]` on page 1249
`POWer<m>:QUALity:STATistics:WFMCount?` on page 1250
`POWer<m>:SWITching:STATistics:ENABle` on page 1249
`POWer<m>:SWITching:STATistics:WFMCount?` on page 1250
`POWer<m>:EFFiciency:STATistics[:ENABle]` on page 1249
`POWer<m>:EFFiciency:STATistics:WFMCount?` on page 1250

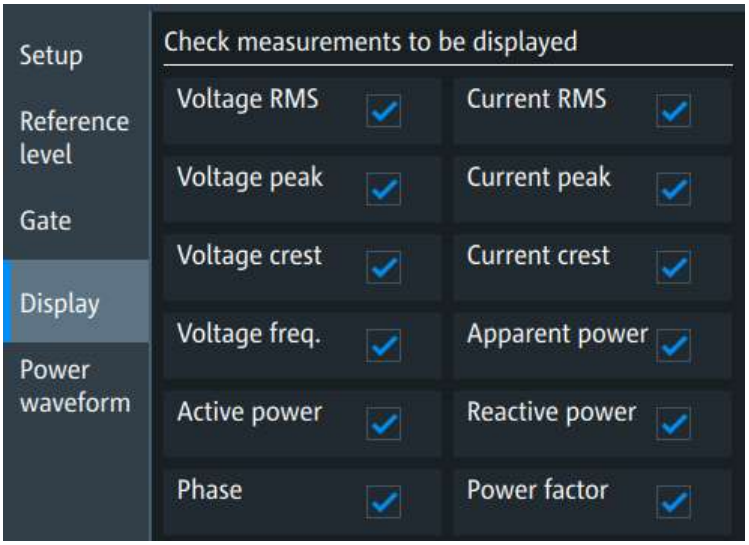
**Clear results**
Resets the statistical results for the respective power measurements, and starts a new statistical evaluation if the acquisition is running.

Remote command:
`POWer<m>:QUALity:STATistics:RESet` on page 1250
`POWer<m>:SWITching:STATistics:RESet` on page 1250
`POWer<m>:EFFiciency:STATistics:RESet` on page 1250

#### 12.3.2.3 Power quality display

Access: "Menu" > "Apps" > "Power" > "+ Add" > "Quality" > select "Quality" > "Display".

In this dialog, you select the power quality analysis measurements that you want to display. For more details on the different measurements, see Section 12.3.2.7, "Power quality results", on page 378.




Remote commands

- `POWer<m>:QUALity:DISPlay:CURRent:CREStfactor` on page 1252
- `POWer<m>:QUALity:DISPlay:CURRent:PEAK` on page 1253
- `POWer<m>:QUALity:DISPlay:CURRent:RMS` on page 1253
- `POWer<m>:QUALity:DISPlay:POWer:ACTive` on page 1253
- `POWer<m>:QUALity:DISPlay:POWer:APParent` on page 1253
- `POWer<m>:QUALity:DISPlay:POWer:PFACtor` on page 1254
- `POWer<m>:QUALity:DISPlay:POWer:PHASe` on page 1254
- `POWer<m>:QUALity:DISPlay:POWer:REACtive` on page 1254
- `POWer<m>:QUALity:DISPlay:VOLTage:CREStfactor` on page 1254
- `POWer<m>:QUALity:DISPlay:VOLTage:FREQuency` on page 1255
- `POWer<m>:QUALity:DISPlay:VOLTage:PEAK` on page 1255
- `POWer<m>:QUALity:DISPlay:VOLTage:RMS` on page 1255

#### 12.3.2.4 Reference level

See Section 12.3.8.1, "Reference level", on page 410.

#### 12.3.2.5 Gate

See Section 12.3.8.2, "Gate settings for power measurements", on page 414.

#### 12.3.2.6 Power waveform

For a description of the power waveform scaling, see Section 12.3.8.3, "Power waveform", on page 415.

### 12.3.2.7 Power quality results




The results of power quality measurements are provided in two ways:

- The diagram shows the graphical presentation of:
  - The voltage waveform
  - The current waveform
  - The power waveform, which is the product of the current and voltage waveforms.
- The result table displays the numeric measurement results:

#### Voltage and current results

The voltage and current results are defined as follows:

| Result | Description |
|---|---|
| Current RMS, voltage RMS | Square root of the mean of the square of the current or voltage averaged over N cycles |
| Voltage frequency | Frequency of the signal |
| Current cycle crest factor, voltage cycle crest factor | Peak value / RMS value |

#### Power results

The power in a system is described by several physical quantities:

| Result | Unit | Formula | Description |
|---|---|---|---|
| Apparent power, \|S\| | VA | $\|S\| = V_{RMS} \cdot I_{RMS}$ | \|S\| is the magnitude of the vector sum of active and reactive power (the complex power S). |
| Reactive power, Q | VAR (Volt-Ampere reactive) | $Q = \|S\| \sin\varphi$ | Power flow that is temporally stored in a system because of the inductive and capacitive elements.<br>For the individual phase values, Q<0 indicates a capacitive load, Q>0 indicates an inductive load. The total reactive power is always positive. |
| Power factor, $P_{Factor}$ | - | $P_{Factor} = P / \|S\|$ | Measure of the system efficiency. The value varies between -1 and 1. |
| Active power, P | W | $P = V_{INSTANTENEOUS} \cdot I_{INSTANTENEOUS}$ | Real power that can be converted into work. The result can be positive or negative. Positive values indicate active power consumption. Negative values indicate active power generation. |
| Phase, φ | ° | $\varphi = acos\ (P_{Factor})$ | Phase angle between the current and the voltage sine waves.<br>φ<0 indicates a capacitive load, φ>0 indicates an inductive load.<br>The total phase angle has the same sign convention as Q. |

Remote commands: "Power quality results" on page 1255.

### 12.3.3 Power harmonics

Current harmonics appear in an electric power system due to nonlinear electric loads. The harmonics can be ejected back into the AC line and disturb other equipment on the grid. To avoid this disturbance, there are often standards of compliance that consumer or industry end-products must meet.

The "Harmonics" analysis tests the devices according to the precompliance standards EN 61000-3-2, MIL-STD-1399 and RTCA DO-160.

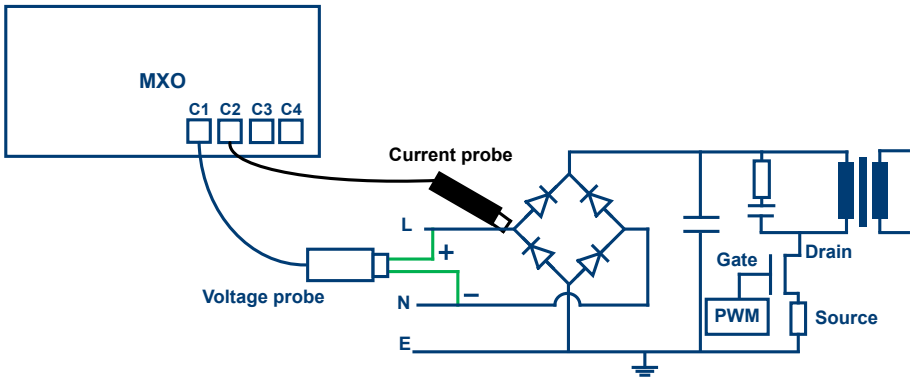
Required probes:

- Differential voltage probe
- Current probe

#### 12.3.3.1 Configuring power harmonics

For details of the configuration settings, see Section 12.3.3.2, "Power harmonic settings", on page 381.

1. Add a new power quality measurement as described in Section 12.3.1.1, "Select power measurements", on page 369.
2. Connect the differential voltage probe and the current probe to the oscilloscope.
3. Deskew the probes as described in Section 12.3.1.2, "Deskew", on page 371.
4. Connect the probes to the DUT:




   a) Connect the positive (+) signal socket of the differential voltage probe to the line of the AC input.
   b) Connect the negative (-) signal socket of the differential voltage probe to the neutral of the AC input.
   c) Connect the current probe to the line of the AC input.
5. In the "Analysis" tab, tap the "Harmonics" measurement.

   The "Harmonics" setup dialog opens.
6. Select the correct channels for the "Voltage" and the "Current" probes.
7. Select the "Standard".
8. Select the "Fund. frequency" according to your signal.

   On the screen, you can see the measurement of the current, the voltage and the power. Also there is a table giving information about important measurement parameters. For details, see Section 12.3.3.6, "Power harmonic results", on page 385.

### 12.3.3.2 Power harmonic settings

In this tab, you configure the current harmonic measurement parameters and display settings.

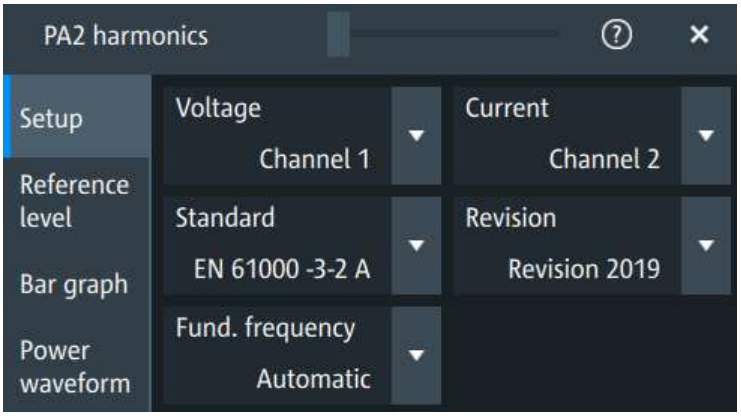



*Figure 12-8: Power harmonic setup for EN 61000-3-2 A / EN 61000-3-2 B*

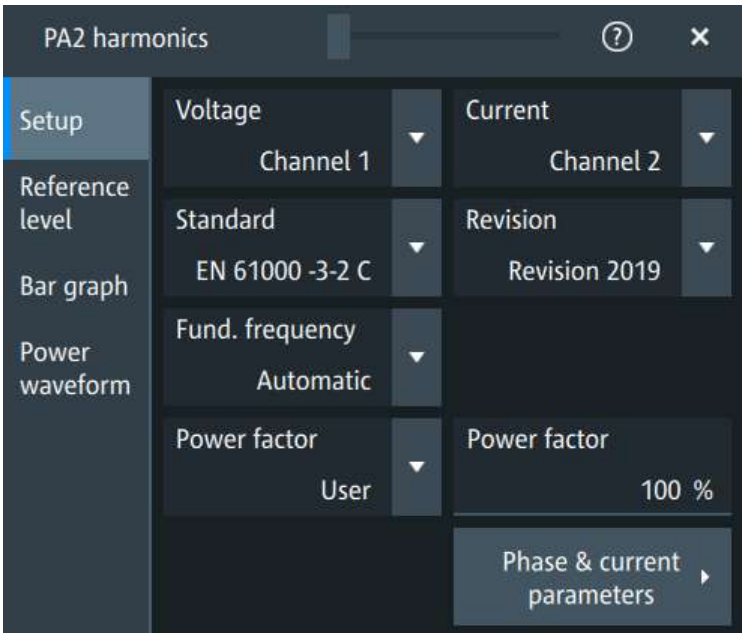



*Figure 12-9: Power harmonic setup for EN 61000-3-2 C*

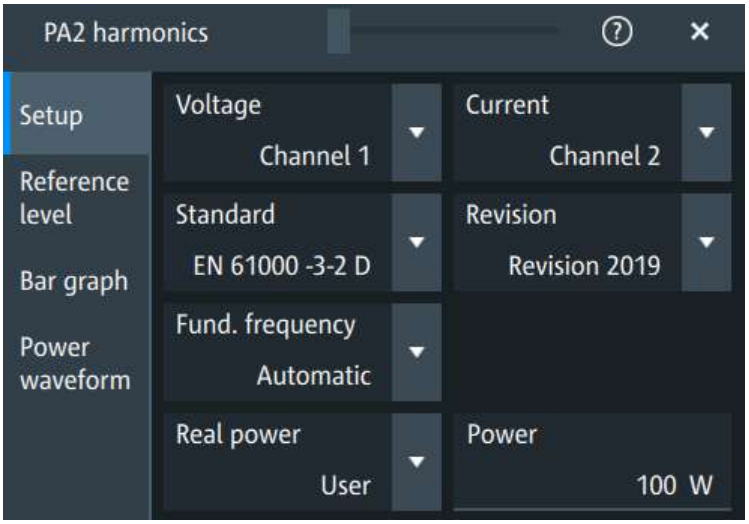



*Figure 12-10: Power harmonic setup for EN 61000-3-2 D*

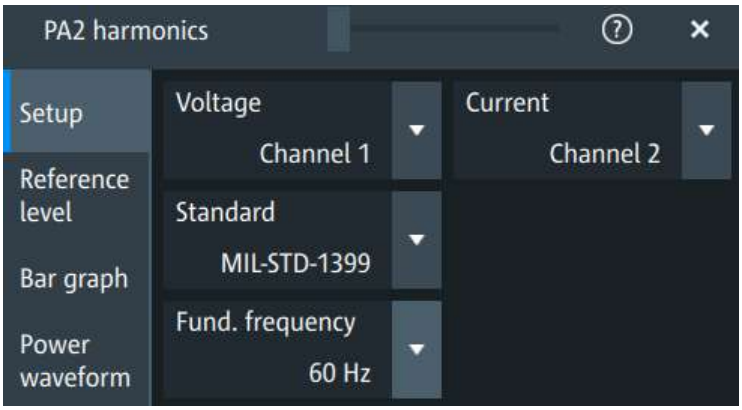



*Figure 12-11: Power harmonic setup for MIL-STD-1399/ RTCA DO-160*

**Voltage**
Selects the voltage source waveform. Analog channels and math waveforms can be used.

Remote command:
`POWer<m>:HARMonics:SOURce[:VOLTage]` on page 1264

**Current**
Selects the current source waveform. Analog channels and math waveforms can be used.

Remote command:
`POWer<m>:HARMonics:SOURce:CURRent` on page 1263

**Standard**
Select the standard in use.

For EN 61000-3-2 you can also select which "Revision" of the standard is used, "Revision 2011" or "Revision 2019".

*Table 12-1: Current harmonic precompliance standards*

| Standard | Application |
|---|---|
| EN 61000-3-2 Class A | Balanced 3-phase equipment, household appliances (excluding equipment identified as class D), tools (excluding portable tools), dimmers for incandescent lamps, audio equipment |
| EN 61000-3-2 Class B | Portable tools, not professional arc welding equipment |
| EN 61000-3-2 Class C[1)] | Lighting equipment |
| EN 61000-3-2 Class D | PC, PC monitors, radio, or TV receivers with an input power less than or equal to 600W |
| MIL-STD-1399 | Military shipboard user equipment |
| RTCA DO-160 | Environmental tests of avionics hardware |
| [1)] The EN 61000-3-2 defines different limits for EN 61000-3-2 Class C equipment with an input power smaller or equal than 25W either. The limits that are implemented in the MXO 3 firmware comply with the limits of table 3 (column two) of the EN 61000-3-2 standard. | |

Remote command:
`POWer<m>:HARMonics:STANdard` on page 1264
`POWer<m>:HARMonics:REVision` on page 1264

**Fund. frequency**
Selects the frequency of the input signal. The available frequencies depend on the selected Standard.

Remote command:
`POWer<m>:HARMonics:FREQuency:EN` on page 1265
`POWer<m>:HARMonics:FREQuency:MIL` on page 1265
`POWer<m>:HARMonics:FREQuency:RTCA` on page 1265

**Power factor**
Available only for "Standard" = EN 61000-3-2 Class C.

Selects if the power factor is defined automatically, or a user-defined value ("Power factor") is used.

Remote command:
`POWer<m>:HARMonics:PFACtor[:MODE]` on page 1268
`POWer<m>:HARMonics:PFACtor:USER` on page 1267

**Phase & current parameters**
Shows a graphical explanation of the relative phase angle and current parameters for "Standard" = EN 61000-3-2 Class C.

**Real power, Power**
Available only for "Standard" = EN 61000-3-2 Class D.

Selects if the real power is defined automatically, or a user-defined value ("Power") is used.

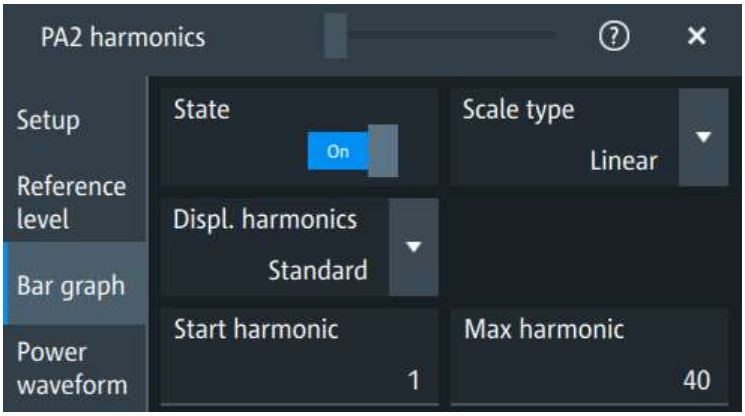
Remote command:
POWer<m>:HARMonics:RPOWer[:MODE] on page 1267
POWer<m>:HARMonics:RPOWer:USER on page 1267

#### 12.3.3.3 Reference level

See Section 12.3.8.1, "Reference level", on page 410.

#### 12.3.3.4 Bar graph

In this tab you can configure the bar graph display.




**State**
Shows the bar graph in the diagram.

**Scale type**
Selects a logarithmic or linear scale for the display for the harmonics bar graph.

Remote command:
POWer<m>:HARMonics:DISPlay:BARGraph:VERTical:SCALe:TYPE
on page 1266

**Displ. harmonics**
Selects which harmonics are displayed in the bar graph: all, odd even or depending on the standard definition.

Remote command:
POWer<m>:HARMonics:DISPlay:HARMonics[:DISPlay] on page 1266

**Start harmonic, Max harmonic**
Set the first and last harmonic to be shown in the bar graph.

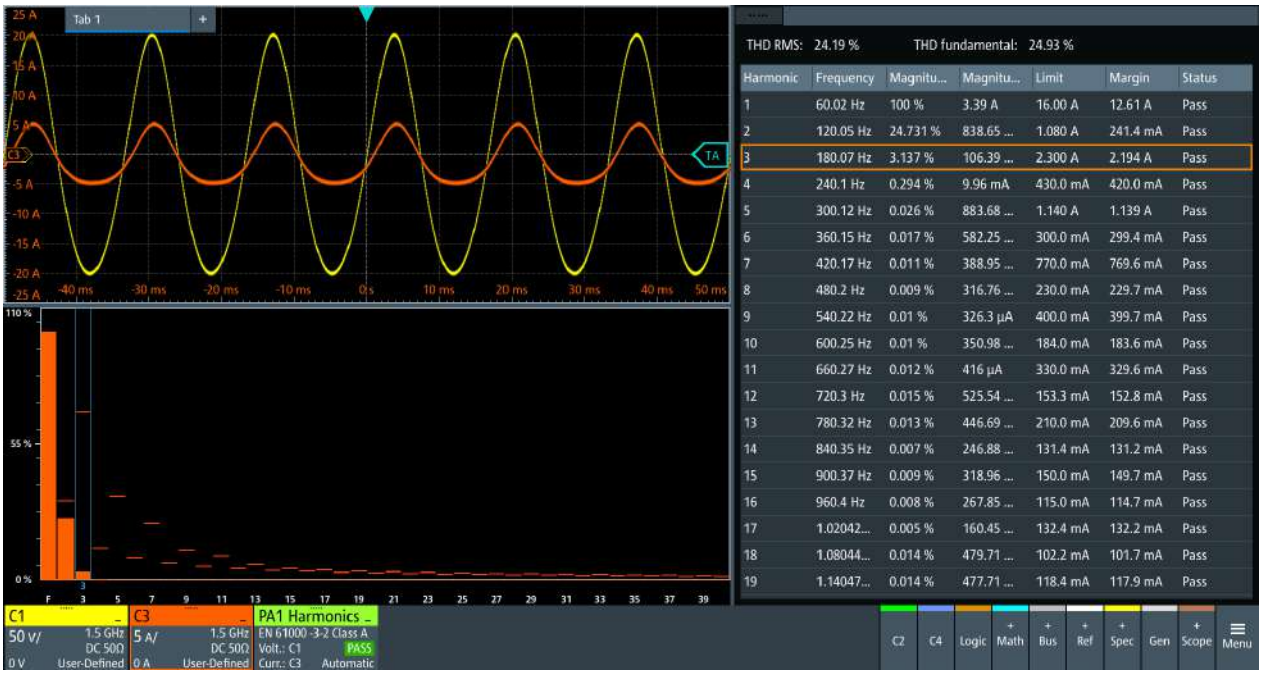
Remote command:
POWer<m>:HARMonics:DISPlay:HARMonics:STARt[:VALue] on page 1266
POWer<m>:HARMonics:DISPlay:HARMonics:STOP[:VALue] on page 1266

### 12.3.3.5 Power waveform

A power waveform is available if "Standard" is set to EN 61000-3-2 Class C or EN 61000-3-2 Class D.

For a description of the power waveform scaling, see Section 12.3.8.3, "Power waveform", on page 415.

### 12.3.3.6 Power harmonic results




The results of the harmonic measurements are provided in two ways:

- The diagram shows the graphical presentation of the voltage waveform, the current waveform and the power waveform.
- The results are a bar chart and a table with the numerical measurement results. The number of displayed harmonics depends on the standard definition. When you tap on a harmonic in the bar chart, the corresponding row in the result table is highlighted.

The current harmonic results displayed in the result table are defined as follows:

| Result table | Bar chart match | Description |
|---|---|---|
| THD RMS | - | Total harmonic distortion relative to the RMS amplitude |
| THD fundamental | - | Total harmonic distortion relative to the fundamental |
| Harmonic index | Value of the x-axis | The harmonic order |
| Frequency | - | The frequency value of the signal |

| Result table | Bar chart match | Description |
|---|---|---|
| Magnitude (%) | Value of the y-axis | The present value of the current harmonic |
| Magnitude (RMS) | | The measured RMS value of the magnitude |
| Limit | Dark orange line | The maxim allowed value according to the selected standard |
| Margin | | Sets the upper limit for the display of the Y scale. |
| Status | Bar for magnitude below limit has the color of the channel (pass)<br>Bar for magnitude outside limit is orange (fail) | Displays, if the harmonics is within the defined limit (pass) or not (fail). |

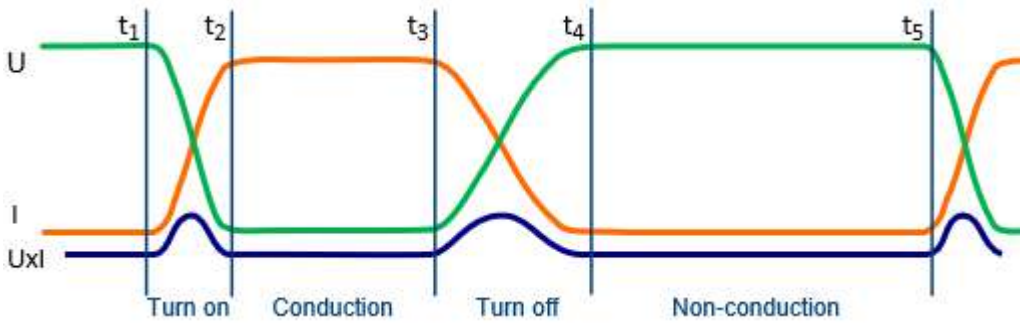
### 12.3.4 Switching loss

The switching loss analysis measures the power and energy losses of a switching device. The losses occur during the switching phases and the conduction phase of the switching transistor.

Required probes:

- Differential voltage probe
- Current probe

**Phases of the switching cycle**



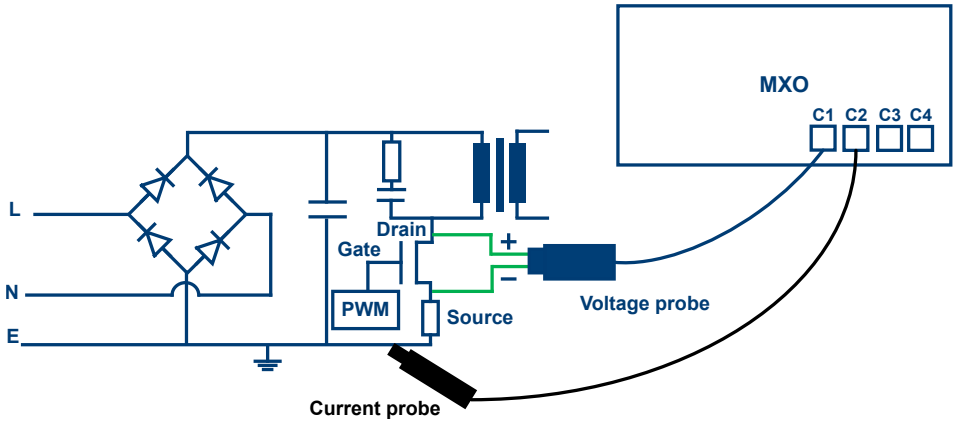

| Phase | Definition Points | Description |
|---|---|---|
| Turn on | The area between "$t_1$" and "$t_2$" | The time after switching on the device, during which the current rises until it reaches the saturation current level. |
| Conduction | The area between "$t_2$" and "$t_3$" | The time during which the voltage is at the transistors saturated minimum and the current flows. |

| Phase | Definition Points | Description |
|---|---|---|
| Turn off | The area between "$t_3$" and "$t_4$" | The time during which after a short delay the voltage rises until it reaches its final value. |
| Non conduction | The area between "$t_4$" and "$t_5$" | The time when the current does not flow. The losses during this period should be theoretically zero. |

#### 12.3.4.1 Configuring switching loss measurements

For details of the configuration settings, see Section 12.3.4.2, "Switching loss setup", on page 388.

1. Add a switching loss measurement as described in Section 12.3.1.1, "Select power measurements", on page 369.
2. Connect the differential voltage probe and the current probe to the oscilloscope.
3. Deskew the probes as described in Section 12.3.1.2, "Deskew", on page 371.
4. Connect the probes to the DUT:



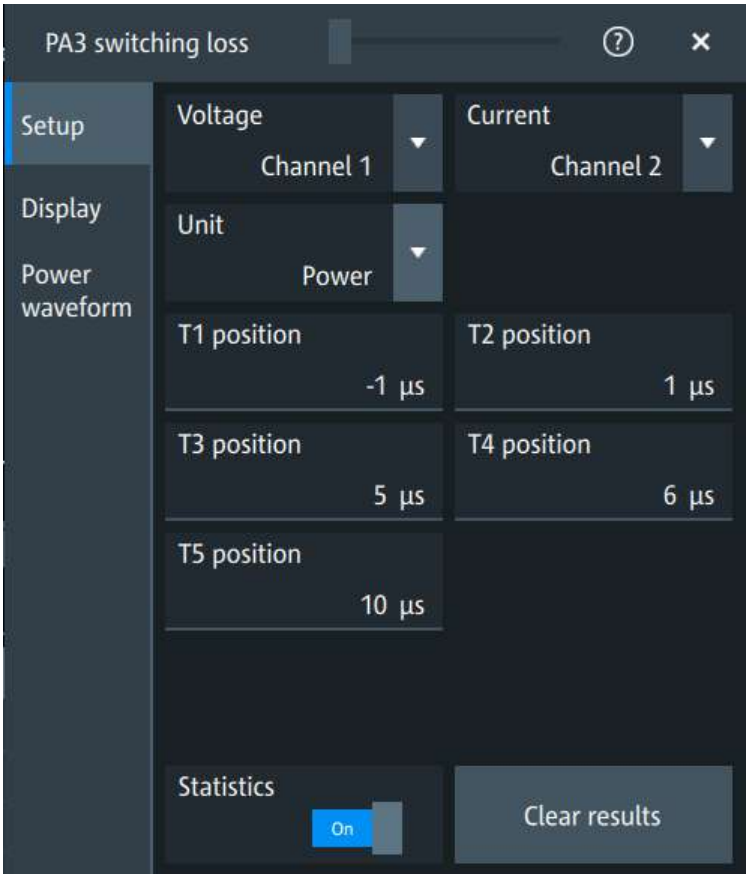

   a) Connect the positive (+) signal socket of the differential voltage probe to the drain of the transistor.
   b) Connect the negative (-) signal socket of the differential voltage probe to the source of the transistor.
   c) Connect the current probe to the source of the transistor.
5. Measure the time points of the switching cycle, for example, using cursor measurements.
6. In the signal bar, tap twice the signal icon of the switching loss power measurement to open its settings dialog.
7. Select the correct channels for the "Voltage" and the "Current" probes.
8. Set the "Unit" for the trace: power or energy.

9. Set the "T<n> position" values.
10. Enable statistics.
11. Select the "Display" tab.
12. Select the measurements that you want to see. You can also change the color of the partial measurements.
13. Select the "Power waveform" tab.
14. Adjust the scale and offset of the resulting trace if necessary.

    On the screen, you can see the measurement waveforms of the current, the voltage and the power. Also, the result table with numeric measurement results is shown. For details, see Section 12.3.4.5, "Switching loss results", on page 391.

### 12.3.4.2 Switching loss setup

Access: "Menu" > "Apps" > "Power" > "+ Add" > "Switching loss" > select "Switching loss" > "Setup".




In the "Setup" tab, you configure the switching loss measurement: select the sources, the time points of the switching loss cycle, and if you want to measure either power or energy.

**Voltage**
Selects the voltage source waveform. Analog channels and math waveforms can be used.

Remote command:
POWer<m>:SWITching:SOURce[:VOLTage] on page 1273

**Current**
Selects the current source waveform. Analog channels and math waveforms can be used.

Remote command:
POWer<m>:SWITching:SOURce:CURRent on page 1272

**Unit**
Selects the measurement type: power or energy.

Remote command:
POWer<m>:SWITching:DISPlay:TYPE on page 1273

**T<n> position**
T1 to T5 define the time points in the switching cycle. You can use cursor measurements to get the time values.

- T1: Start of the turn-on area, and start of the total switching cycle.
- T2: End of the turn-on area, and start of the conduction area.
- T3: End of the conduction area, and start of the turn-off area.
- T4: End of the turn-off area, and start of the non-conduction area.
- T5: End of non-conduction area, and end of the total switching cycle.

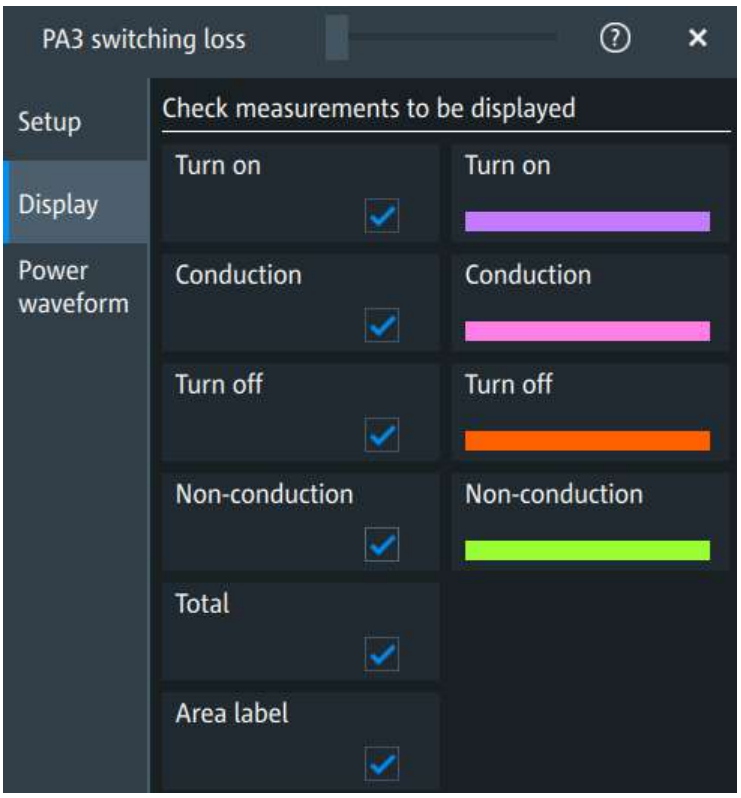
See "Phases of the switching cycle" on page 386.

Remote command:
POWer<m>:SWITching:REGion:TOTal[:STARt] on page 1273 (T1)
POWer<m>:SWITching:REGion:TON[:STARt] on page 1273 (T1)
POWer<m>:SWITching:REGion:TON:STOP on page 1273 (T2)
POWer<m>:SWITching:REGion:CONDuction[:STARt] on page 1273 (T2)
POWer<m>:SWITching:REGion:CONDuction:STOP on page 1274 (T3)
POWer<m>:SWITching:REGion:TOFF[:STARt] on page 1274 (T3)
POWer<m>:SWITching:REGion:TOFF:STOP on page 1274 (T4)
POWer<m>:SWITching:REGion:NCONduction[:STARt] on page 1274 (T4)
POWer<m>:SWITching:REGion:NCONduction:STOP on page 1275 (T5)
POWer<m>:SWITching:REGion:TOTal:STOP on page 1275 (T5)

**Statistics, Clear results**
See "Statistics" on page 376 and "Clear results" on page 376.

#### 12.3.4.3 Switching loss display

Access: "Menu" > "Apps" > "Power" > "+ Add" > "Switching loss" > select "Switching loss" > "Display".




In the "Display" dialog, you select which measurements are performed. The results are displayed in the result table. You can change the colors of the cycle areas in the diagram.

**Turn on**
Enables the measurements of the turn-on area, and selects the display color.

Remote command:
`POWer<m>:SWITching:DISPlay:TON[:ENABle]` on page 1275
`POWer<m>:SWITching:DISPlay:TON:COLor` on page 1276

**Conduction**
Enables the measurements of the conduction area, and selects the display color.

Remote command:
`POWer<m>:SWITching:DISPlay:CONDuction[:ENABle]` on page 1275
`POWer<m>:SWITching:DISPlay:CONDuction:COLor` on page 1276

**Turn off**
Enables the measurements of the turn-off area, and selects the display color.

Remote command:
`POWer<m>:SWITching:DISPlay:TOFF[:ENABle]` on page 1275
`POWer<m>:SWITching:DISPlay:TOFF:COLor` on page 1276

**Non-conduction**
Enables the measurements of the non-conduction area, and selects the display color.

Remote command:
POWer<m>:SWITching:DISPlay:NCONduction[:ENABle] on page 1275
POWer<m>:SWITching:DISPlay:NCONduction:COLor on page 1276

**Total**
Enables the measurements of the total switching cycle.

Remote command:
POWer<m>:SWITching:DISPlay:TOTal[:ENABle] on page 1276

**Area label**
Displays the names of the switching cycle areas.

Remote command:
POWer<m>:SWITching:DISPlay:LABel[:ENABle] on page 1276

#### 12.3.4.4 Power waveform

For a description of the power waveform display and scaling, see Section 12.3.8.3, "Power waveform", on page 415.

#### 12.3.4.5 Switching loss results

The results of switching loss measurements are:

- Voltage waveform
- Current waveform
- Power waveform, which is the product of the current and voltage waveforms.
- Numeric measurement results of the selected measurement type (power or energy). The results are given for each phase of the switching cycle.

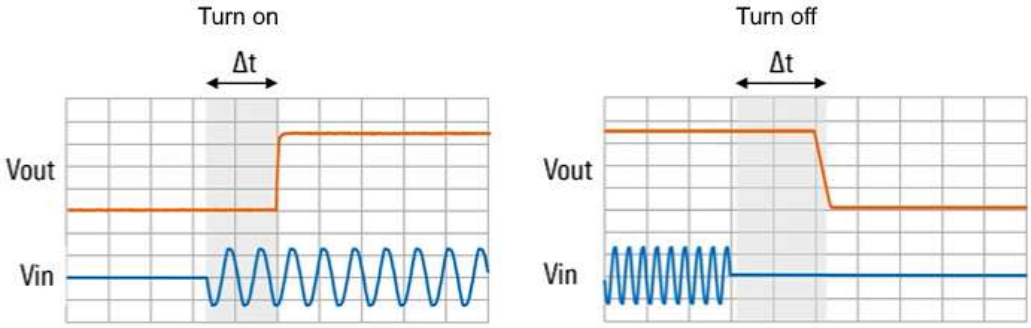
See "Phases of the switching cycle" on page 386 for an explanation of the switching cycle.




Remote commands are described in "Switching loss results" on page 1277.

### 12.3.5 Turn on/off time

The turn-on/off time analysis measures the time that a power supply takes to reach a certain output threshold when turned on or turned off. The output thresholds are usually a percentage of the steady state output level: 90% for turn-on, and 10% for turn-off time.




Required probes:

- AC input: differential voltage probe
- DC input: passive or differential voltage probe
- DC output: passive or differential voltage probe
  You can measure several outputs in parallel. The maximum number of output connections depends on the number of channels on the instrument: 3 outputs (4-channel instrument) or 7 outputs (8-channel instrument).

#### 12.3.5.1 Configuring turn on/off time measurements

For details of the configuration settings, see Section 12.3.5.2, "Turn on/off setup", on page 394.

1. Add a turn on/off time measurement as described in Section 12.3.1.1, "Select power measurements", on page 369.
2. Connect all voltage probes to the oscilloscope.
3. Connect the probes to the DUT:

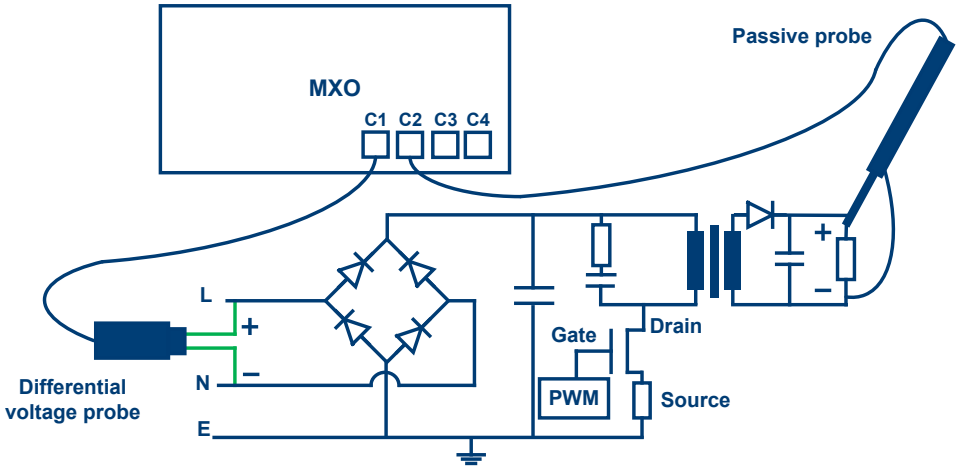



*Figure 12-12: Connection with AC input*

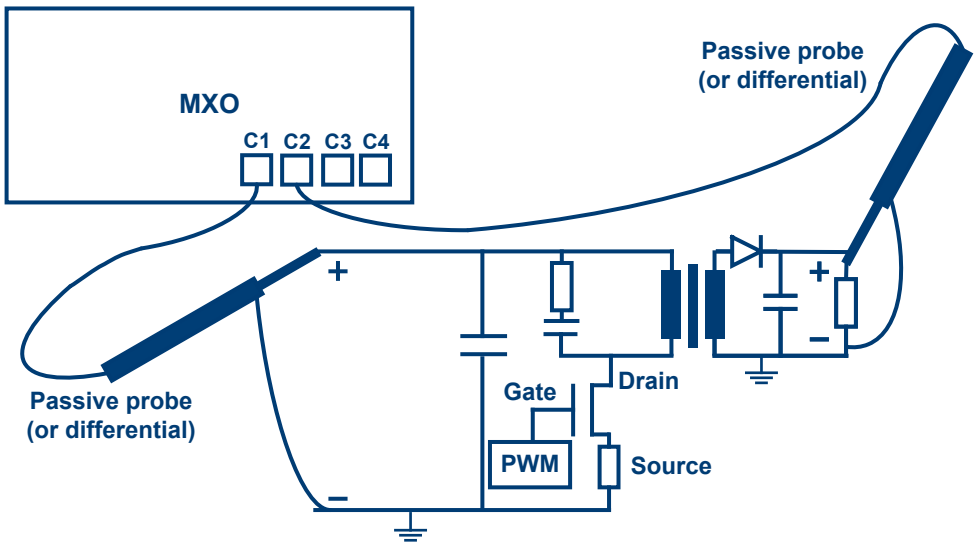



*Figure 12-13: Connection with DC input*

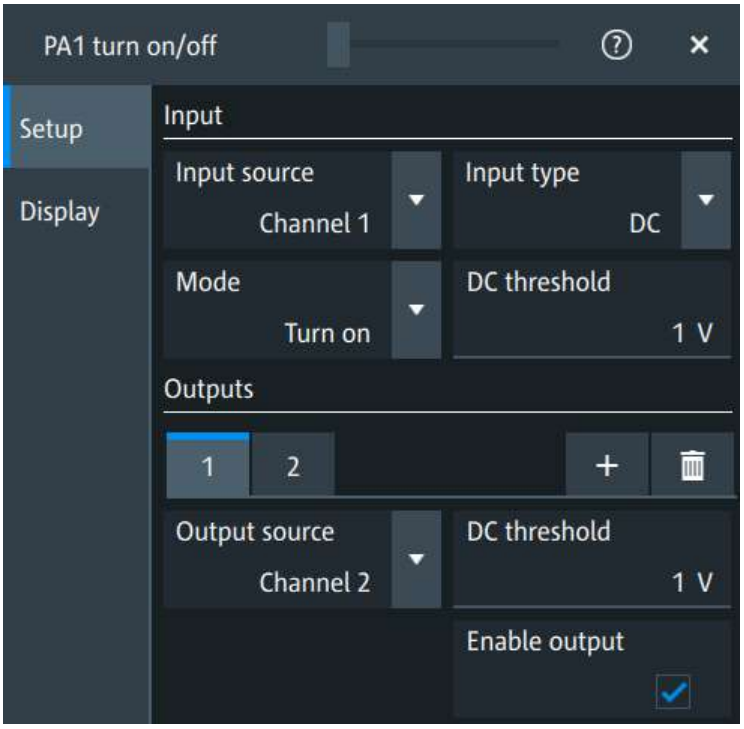
a) AC input:
Connect the positive (+) signal socket of the differential voltage probe to the line (L) of the AC input.
Connect the negative (-) signal socket of the differential voltage probe to the neutral (N) of the AC input.
b) DC input:
Connect the passive or differential voltage probe to the DC input of the power supply.
c) Outputs:
Connect the passive or differential voltage probes to the DC outputs of the power supply.

4. In the "Analysis" tab, tap the "Turn on/off time" measurement.
5. Configure the input settings:
   a) Select the correct channel for the "Input source".
   b) Select the "Mode" (turn on or turn off) and the "Input type" (AC or DC).
   c) Enter the threshold. The measurement starts when the input signal crosses this threshold.
6. Use the + to add the required number of outputs
7. Configure the outputs. For each output:
   a) Select the correct channel for the "Output source".
   b) Enter the threshold. The measurement stops when the output signal crosses this threshold.
   c) Enable the output signal to see the waveform and the results.
8. Select the "Display" tab.
9. Select the outputs for which you want to see the result lines.

   On the screen, you can see the waveforms of the input and the outputs. Also, the result table with numeric measurement results is shown. For details, see Section 12.3.5.4, "Turn on/off results", on page 397.

#### 12.3.5.2 Turn on/off setup

Access: "Menu" > "Apps" > "Power" > "+ Add" > "Turn on/off time" > select "Turn on/off time" > "Setup".




In the "Setup" tab, you configure the turn-on/off time measurement: select the sources, the thresholds, and the measurement.

**Input source**
Selects the channel of the input signal. Analog channels and math waveforms can be used.

Remote command:
`POWer<m>:ONOFf:INPut[:SOURce]` on page 1284

**Input type**
Selects whether the input signal is AC or CD.

Remote command:
`POWer<m>:ONOFf:INPut:TYPE` on page 1283

**Mode**
Selects the turn-on or turn-off measurement.

Remote command:
`POWer<m>:ONOFf[:TYPE]` on page 1285

**DC threshold**
Sets the threshold for the DC input signal.

The time starts when the first edge exceeds a positive threshold value, or falls below a negative threshold value.

Remote command:
`POWer<m>:ONOFf:INPut:DC:ABSolute[:VALue]` on page 1283

**AC threshold**
Sets the threshold for the AC input signal. You set the absolute value, the instrument uses the positive and negative values of the given absolute value.

The time starts when the signal crosses either the positive threshold value, or the negative value - the first crossing point is considered.

Remote command:
`POWer<m>:ONOFf:INPut:AC:ABSolute[:VALue]` on page 1283

**Output source**
Selects the channel of the output signal. Analog channels except for the input channel and math waveforms can be used.

You can set up several outputs in parallel. The maximum number of output connections depends on the number of channels on the instrument.

Remote command:
`POWer<m>:ONOFf:OUTPut<n>[:SOURce]` on page 1285

**DC threshold**
Sets the threshold for the selected output signal.

When the output signal crosses this threshold, the turn-on time ends, or the turn-off time starts.

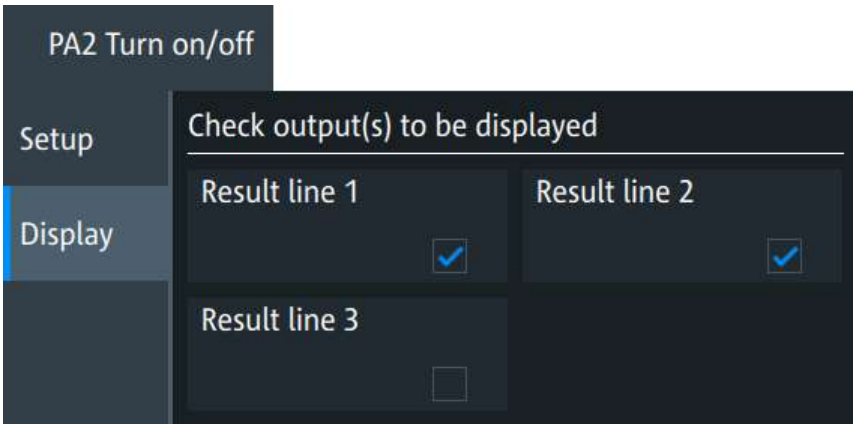
Remote command:
`POWer<m>:ONOFf:OUTPut<n>:DC:ABSolute[:VALue]` on page 1284

**Enable output**
Activates the selected output line.

Remote command:
`POWer<m>:ONOFf:OUTPut<n>:DISPlay:RESult[:ENABle]` on page 1284

### 12.3.5.3 Turn on/off display

For better understanding, you can display result lines for each output separately.



**Result line <n>**
Enables the display of result lines and threshold for the selected output. Result lines are only visible if the threshold is within the signal amplitude. Start and stop time are displayed with vertical dashed lines.

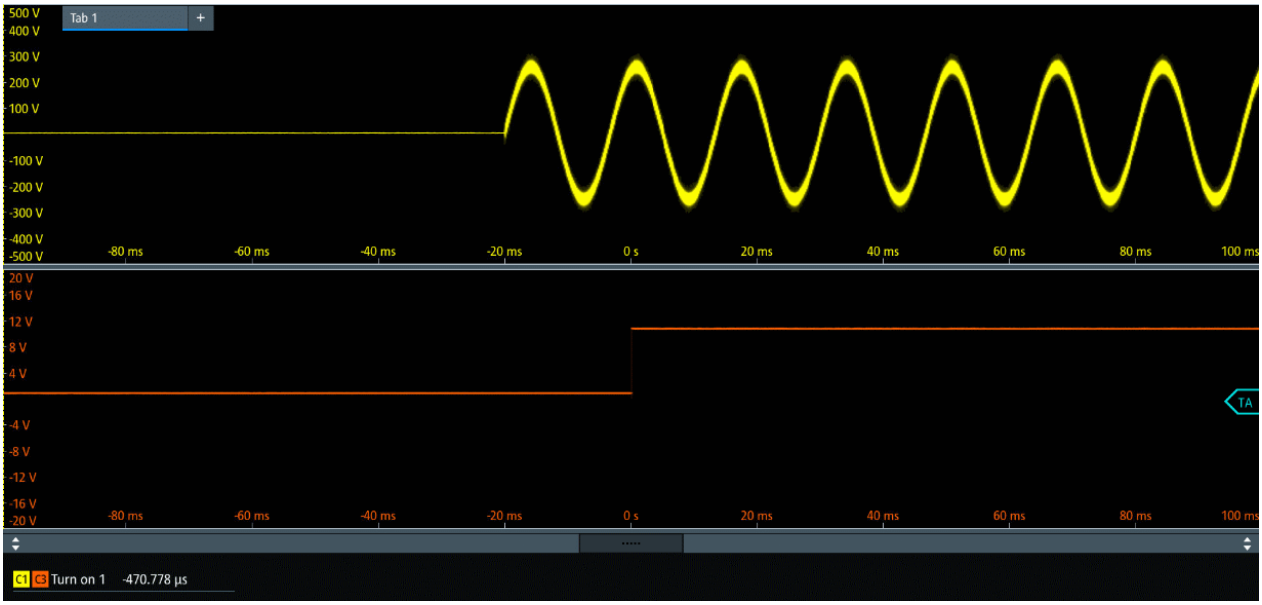
Remote command:
`POWer<m>:ONOFf:OUTPut<n>:DISPlay:RLINe[:ENABle]` on page 1285

### 12.3.5.4 Turn on/off results

The results of turn-on/off time measurements are:

- Input waveform
- Output waveforms
- Measured values of turn-on or turn-off time for each input-output pair.

You can visualize the measured points on the waveform, see Section 12.3.5.3, "Turn on/off display", on page 396.




Remote commands are described in "Turn on/off results" on page 1286.

## 12.3.6 Power efficiency

Power efficiency analysis assesses how effectively a device converts input power into useful output power. It measures the input voltage and current and the output voltage and current, calculates the input and the output power, and the ratio of output power to input power. This ratio percentage is the power efficiency.

Required probes:

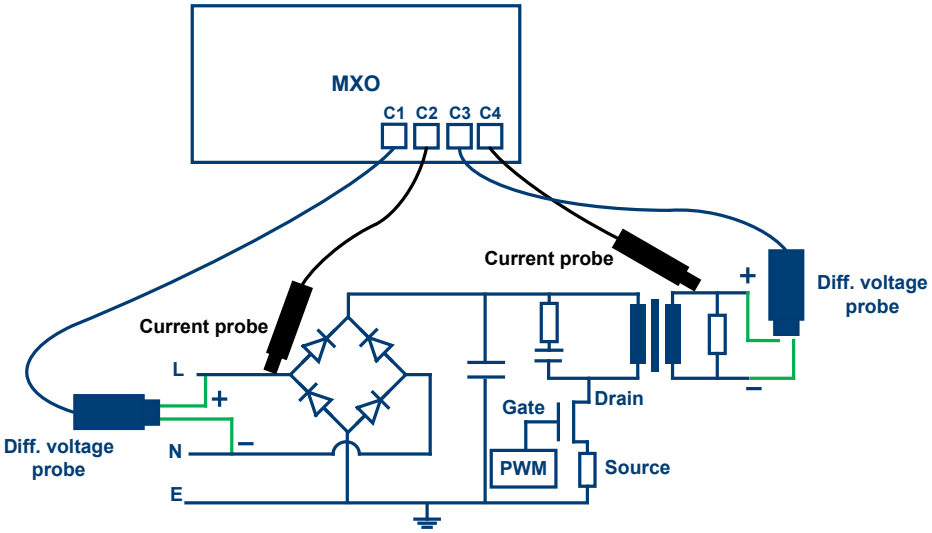
- Input: current probe and differential voltage probe
- Output: current probe and differential voltage probe

If your MXO oscilloscope has more than 4 channels, you can measure several outputs in parallel and obtain the total power and efficiency of the DUT. The maximum number of output lines is *Number of channels / 2 - 1*.

### 12.3.6.1 Configuring power efficiency measurements

For details of the configuration settings, see Section 12.3.6.2, "Power efficiency setup", on page 399.

1. Add a power efficiency measurement as described in Section 12.3.1.1, "Select power measurements", on page 369.
2. Connect the differential voltage probes and the current probes to the oscilloscope.
3. Deskew the probes as described in Section 12.3.1.2, "Deskew", on page 371.
4. Connect the probes to the DUT:




   a) Connect the positive (+) signal socket of the differential voltage probe to the line of the input.
   b) Connect the negative (-) signal socket of the differential voltage probe to the neutral of the input.
   c) Connect the current probe to the line of the input.
   d) Connect a differential voltage probe and a current probe to each output to be measured.
5. In the signal bar, tap twice the signal icon of the efficiency measurement to open its settings dialog.
6. Select the correct channels for the "Voltage" and the "Current" probes of the input and the output.
7. Selects the type of the current flow for input and output: AC or DC: "Input type" and "Output type".

8. Enable statistics.
9. Select the "Display" tab.
10. Select the measurements that you want to see.
11. Select the "Power waveform" tab.
12. Adjust the scale and offset of the resulting waveforms if necessary.

    On the screen, you can see the selected measurement waveforms of the input power and the output power. If your instrument has more than 4 channels and the DUT several outputs, you can see all output power waveforms and the total power if selected. Also, the result table with numeric measurement results is shown. For details, see Section 12.3.6.7, "Efficiency results", on page 403.

### 12.3.6.2 Power efficiency setup

Access: "Menu" > "Apps" > "Power" > "+ Add" > "Efficiency" > select "Efficiency" > "Setup".




*Figure 12-14: Efficieny measurement setup of an 4-channel instrument*

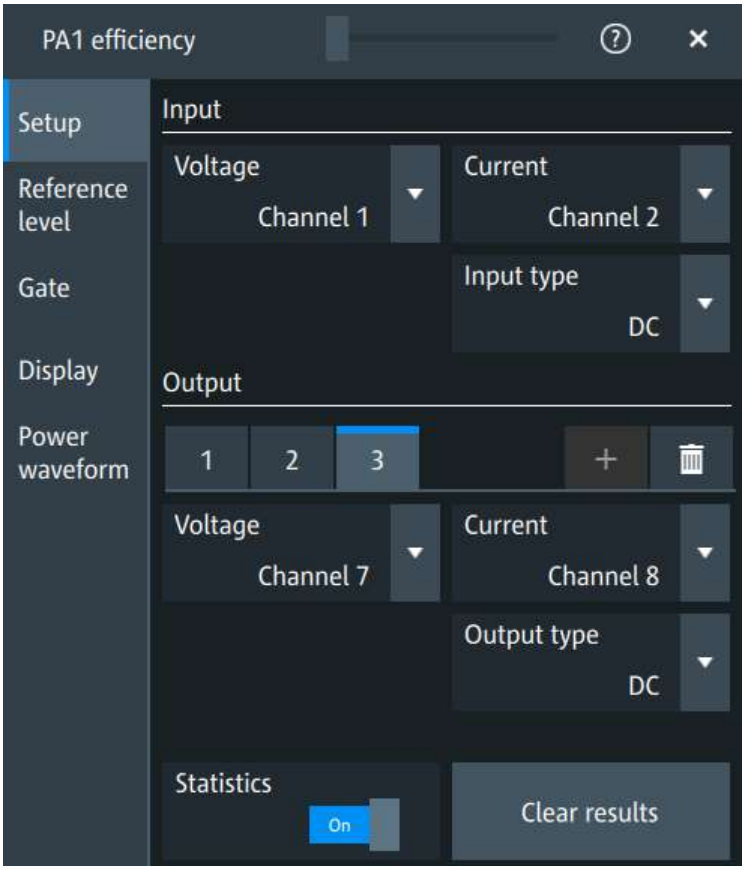



*Figure 12-15: Efficieny measurement setup of an 8-channel instrument*

In the "Setup" tab, you configure the efficiency measurement: select the sources for input and outputs and select whether they are AC or CD. Using several outputs requires more than 4 input channels. You also enable statistics here.

**Voltage (Input)**
Selects the voltage source waveform of the input line. Analog channels and math waveforms can be used.

Remote command:
`POWer<m>:EFFiciency:INPut<n>:VOLTage[:SOURce]` on page 1287

**Current (Input)**
Selects the current source waveform of the input line. Analog channels and math waveforms can be used.

Remote command:
`POWer<m>:EFFiciency:INPut<n>:CURRent[:SOURce]` on page 1286

**Input type**
Selects the type of the current flow: AC or DC.

Remote command:
`POWer<m>:EFFiciency:INPut<n>[:TYPE]` on page 1287

**Voltage (Output)**
Selects the voltage source waveform of the selected output line. Analog channels and math waveforms can be used.

Using several outputs requires more than 4 input channels.

Remote command:
`POWer<m>:EFFiciency:OUTPut<n>:VOLTage[:SOURce]` on page 1288

**Current (Output)**
Selects the current source waveform of the selected output line. Analog channels and math waveforms can be used.

Using several outputs requires more than 4 input channels.

Remote command:
`POWer<m>:EFFiciency:OUTPut<n>:CURRent[:SOURce]` on page 1288

**Output type**
Selects the type of the current flow: AC or DC.

Using several outputs requires more than 4 input channels.

Remote command:
`POWer<m>:EFFiciency:OUTPut<n>[:TYPE]` on page 1288

**Statistics, Clear results**
See "Statistics" on page 376 and "Clear results" on page 376.

#### 12.3.6.3 Reference level

See Section 12.3.8.1, "Reference level", on page 410.

Separate reference level settings are available for input and output. The reference levels take effect only for the AC input and output types.

#### 12.3.6.4 Gate

See Section 12.3.8.2, "Gate settings for power measurements", on page 414.

#### 12.3.6.5 Power efficiency display

In this dialog, you select the power efficiency measurements that you want to display. For more details on the different measurements, see Section 12.3.6.7, "Efficiency results", on page 403.

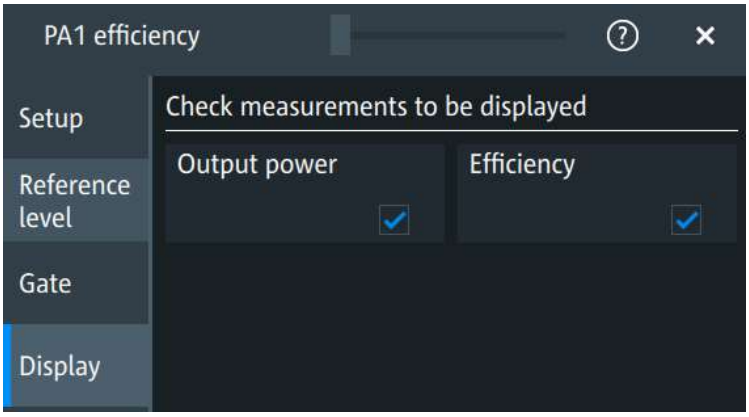



*Figure 12-16: Display settings of an 4-channel instrument*

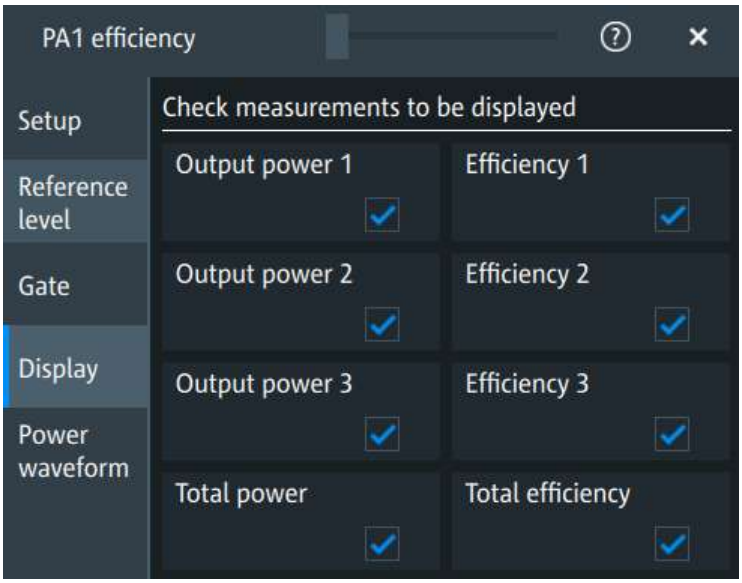



*Figure 12-17: Display settings of an 8-channel instrument*

Remote commands:

- `POWer<m>:EFFiciency:DISPlay:RESult:OUTPut<n>:EFFiciency[:ENABle]` on page 1289
- `POWer<m>:EFFiciency:DISPlay:RESult:OUTPut<n>:POWer[:ENABle]` on page 1289
- `POWer<m>:EFFiciency:DISPlay:RESult:TOTal<n>:EFFiciency[:ENABle]` on page 1289
- `POWer<m>:EFFiciency:DISPlay:RESult:TOTal<n>:OPOWer[:ENABle]` on page 1289

#### 12.3.6.6 Power waveform

You can enable and scale the input, output and total waveforms individually.




*Figure 12-18: Waveform settings of an 4-channel instrument*

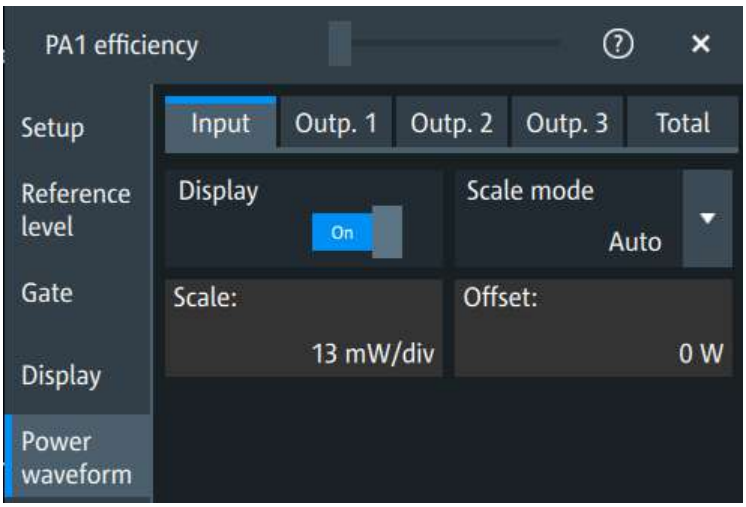



*Figure 12-19: Waveform settings of an 8-channel instrument*

For a description of the power waveform display and scaling, see Section 12.3.8.3, "Power waveform", on page 415.

### 12.3.6.7 Efficiency results

The results of efficiency measurements are:

- Input current and voltage waveforms
- Output current and voltage waveforms
- Input power waveform, which is the product of the input current and input voltage waveforms.
- Output power waveforms, which are the products of the corresponding output current and output voltage waveforms.
  Using several outputs requires more than 4 input channels.
- Total power waveform, which is the sum of the output power values.

Total power requires more than 4 input channels.

- Numeric measurement results of input power, output power, total power and efficiencies. The total efficiency is the ratio of input power to the sum of output power values.

If the input or output type is set to AC, the period is measured on the AC voltage source. If one of these measurements fails, the efficiency result is invalid.

If statistics are enabled, every period for an AC source is analyzed. Without statistics, only the first period is analyzed.

Remote commands are described in section "Efficiency results" on page 1290.

## 12.3.7 Safe operating area (SOA)

The safe operating area (SOA) analysis is defined by the voltage and current conditions over which a power semiconductor device is expected to operate without self-damage. The safe operating area analysis provides a diagram of the safe operating conditions of your DUT.

In parallel to the time domain diagram, an XY-plot is created, with voltage points on the x-axis, while current is shown on the y-axis. The XY-plot is tested against a mask, which represents the safe operating area.

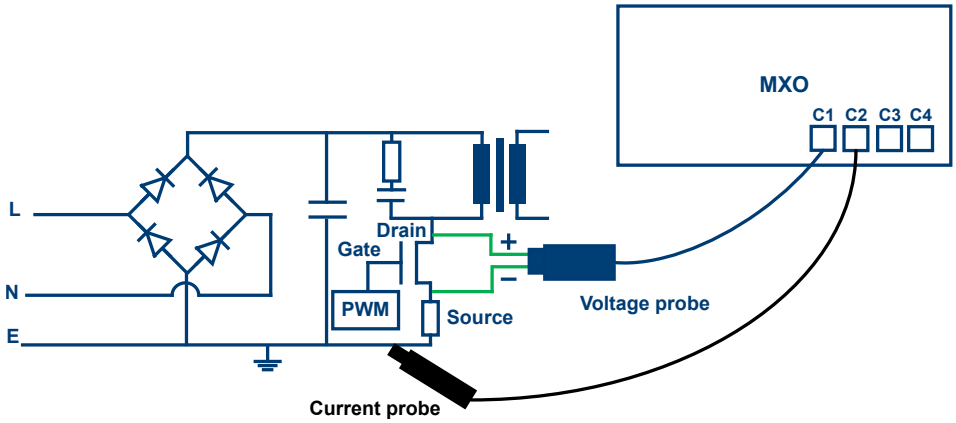
Required probes:

- Differential voltage probe
- Current probe

### 12.3.7.1 Configuring SOA measurements

For details of the configuration settings, see Section 12.3.7.2, "SOA setup", on page 405 and Section 12.3.7.3, "SOA masks", on page 407.

1. Add a SOA measurement as described in Section 12.3.1.1, "Select power measurements", on page 369.
2. Connect the differential voltage probes and the current probes to the oscilloscope.
3. Deskew the probes as described in Section 12.3.1.2, "Deskew", on page 371.
4. Connect the probes to the DUT:




a) Connect the positive (+) signal socket of the differential voltage probe to the drain of the transistor.
b) Connect the negative (-) signal socket of the differential voltage probe to the source of the transistor.
c) Connect the current probe to the source of the transistor.

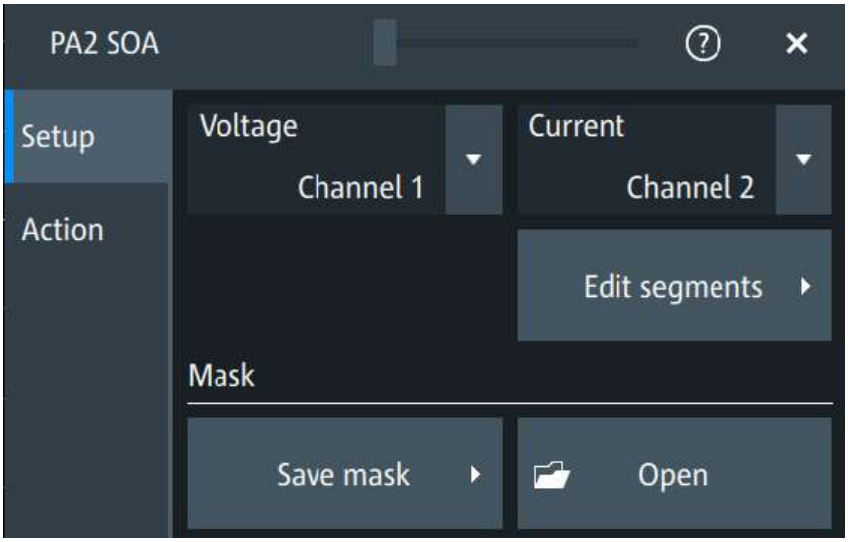
5. In the signal bar, tap twice the signal icon of the SOA measurement to open its settings dialog.
6. Select the correct channels for the "Voltage" and the "Current" probes.
7. If a SOA mask is already available, select "Open" and load the mask.
8. If no mask is available, select "Edit segments". Create the mask segments, either under "Predefined" or "Limit".

   Both segment types can be combined in one mask. For more information on predefined masks, see also Section 12.1.5, "Actions on mask test results", on page 338.
9. "Save mask" if you want to keep it for further tests.
10. Select the "Action" tab.
11. Select the actions to be initiated on mask violation or successful completion of the test. You can select several actions.
12. Run the acquisitions.

    On the screen, you can see the voltage and current waveforms and the XY-diagram with the mask. Also, the result table with numeric measurement results and the test status is shown. For details, see Section 12.3.7.5, "SOA results", on page 409.

### 12.3.7.2 SOA setup

Access: "Menu" > "Apps" > "Power" > "+ Add" > "SOA" > select "SOA" > "Setup".




In the "Setup" tab, you configure the SOA measurement: select the sources, load and save the mask, and start creating a mask.

**Voltage**
Selects the voltage source waveform. Analog channels and math waveforms can be used.

In the SOA diagram, voltage points are plotted on the x-axis.

Remote command:
`POWer<m>:SOA:SOURce:VOLTage` on page 1294

**Current**
Selects the current source waveform. Analog channels and math waveforms can be used.

In the SOA diagram, current is plotted on the y-axis.

Remote command:
`POWer<m>:SOA:SOURce:CURRent` on page 1294

**Edit segments**
Opens a dialog to create and configure the SOA mask. See Section 12.3.7.3, "SOA masks", on page 407.

**Save mask**
Define the storage location and filename, and save the mask.

| | |
|---|---|
| "<Folder>" | Opens a file explorer where you can select the directory where the file is saved. |
| "File name base" | Sets a name for the file, without extension. The name is extended with a time stamp when the file is saved. Thus, multiple "Save" actions are possible without changing the filename. |
| "File extension" | The format of the mask file is always `.xml`. |

"Save" Saves the file in the defined folder using the "File name base". If the specified file already exists, it is overwritten with the new data.

"Save As" Opens a file explorer where you select the folder, the file type, enter the filename and save the file.

Remote command:
POWer<m>:SOA:MTESt<n>:IMEXport:NAME on page 1293
POWer<m>:SOA:MTESt<n>:IMEXport:SAVE on page 1294

**Open**
Opens a file explorer to select a previously stored mask file. The selected mask is loaded.

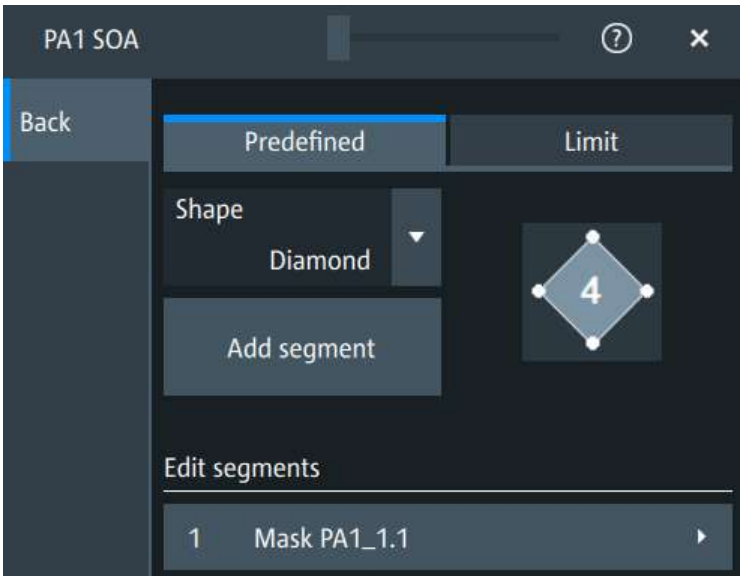
Remote command:
POWer<m>:SOA:MTESt<n>:IMEXport:NAME on page 1293
POWer<m>:SOA:MTESt<n>:IMEXport:OPEN on page 1293

### 12.3.7.3 SOA masks

Access: "Menu" > "Apps" > "Power" > "+ Add" > "SOA" > select "SOA" > "Setup" > "Edit segments".

You can define and edit 1 mask with up to 4 segments.

**Predefined**
To create a predefined mask, you select the shape of a mask segment, add it and then edit the positions of the corner points.

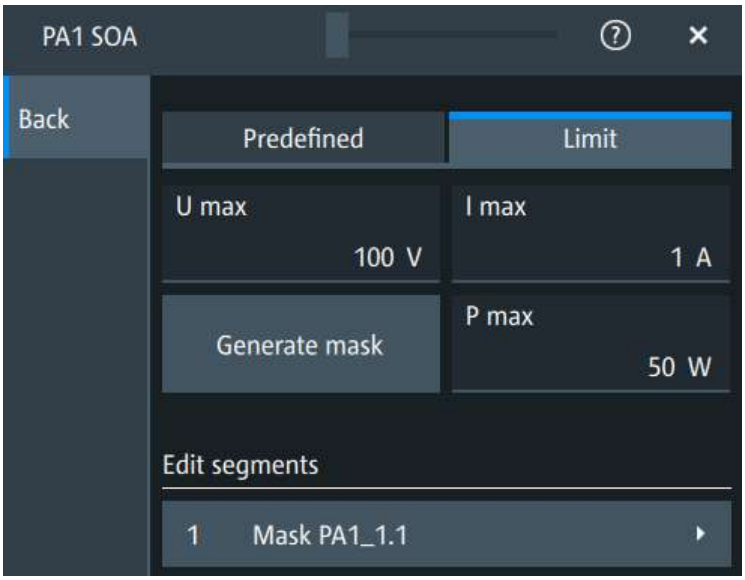



**Shape ← Predefined**
Selects the shape of the SOA mask: rectangle, diamond, hexagon, octagon.

**Add segment ← Predefined**
Adds a new segment with the selected shape to the mask. To adjust the size and position of the mask segment, select the segment and configure it. The procedure is the same as in the mask test application. See "Add segment" on page 335 and "Edit segments" on page 335.

Remote command:
POWer<m>:SOA:MTESt<n>:SEGMent<o>:ADD on page 1296
POWer<m>:SOA:MTESt<n>:SEGMent<o>:COUNt? on page 1296
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:ADD on page 1297
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:COUNt? on page 1297
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:REMove on page 1298
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:VALid? on page 1298
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:X on page 1298
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:Y on page 1299
POWer<m>:SOA:MTESt<n>:SEGMent<o>:REMove on page 1299

**Limit**
You can use the values $I_{max}$, $V_{max}$ and $P_{max}$ from the transistor specification to calculate a mask. $I_{max}$, $V_{max}$ and $P_{max}$ must be positive values. The result is a mask segment in the first quadrant of the XY-diagram. You can change the generated segment like other segments: move, add and remove points.




**U max ← Limit**
Sets the maximum voltage for the mask. The value must be positive.

Remote command:
POWer<m>:SOA:LIMit:VMAX on page 1296

**I max ← Limit**
Sets the maximum current for the mask. The value must be positive.

Remote command:
POWer<m>:SOA:LIMit:IMAX on page 1295

**P max ← Limit**
Sets the maximum power for the mask. The value must be positive.

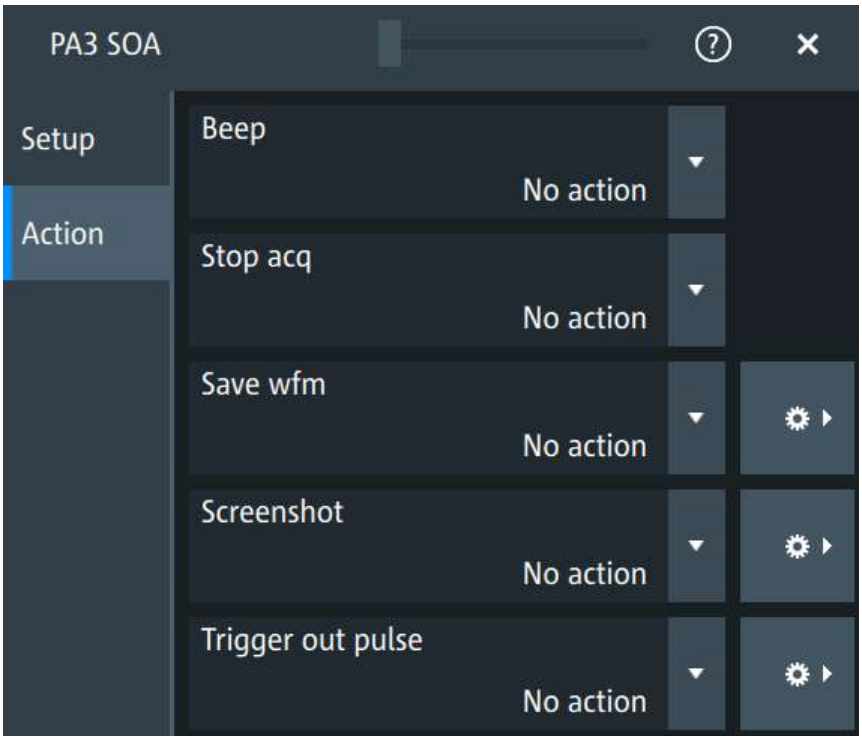
Remote command:
`POWer<m>:SOA:LIMit:PMAX` on page 1295

**Generate mask ← Limit**
Generates a mask using the given maximum values. The result is a mask segment in the first quadrant of the XY-diagram.

Remote command:
`POWer<m>:SOA:LIMit:APPLy` on page 1295

### 12.3.7.4 SOA actions

The safe operating analysis uses the same actions as the general mask test. For details, see Section 12.1.5, "Actions on mask test results", on page 338.




### 12.3.7.5 SOA results

The results of safe operating area measurements are provided in the following ways:

- Voltage source waveform and current source waveform in the diagram
- A logarithmic or linear XY-diagram of the voltage (x-axis) and current (y-axis) waveforms. This XY-diagram is a graphical representation of the power handling capability of the DUT under various conditions.
- If a power waveform in the XY diagram violates a mask, the violation is marked.
- The result table provides the numeric results of the SOA mask test.

*Table 12-2: Results of the SOA test*

| Result | Description |
|---|---|
| Total acq. | Number of tested acquisitions (waveforms) |
| Passed acq. | Number of acquisitions that have passed the mask test, i.e. they are within the defined safe operating area mask |
| Failed acq. | Number of acquisitions that have failed the mask test, i.e. they are out of the defined safe operating area mask |
| Fail rate | Ratio of failed acquisitions to the number of tested acquisitions in % |
| Result | Summary test status: Pass or Fail |

Remote commands are described in "SOA results" on page 1302.

## 12.3.8 Common functions for power measurements

### 12.3.8.1 Reference level

Some power analysis measurements require reference levels to obtain the measurement points. Power analysis uses its own reference level set, which is independent from other measurements and is used for all power measurements.

To ensure that the correct value for the period is used for the measurements, the period is determined on a voltage channel first. This period value is then applied wherever it is required to the current waveform and to the power (Math) waveform.

In power efficiency measurements, separate reference level settings are available for input and outputs. The reference levels take effect only for the AC input and output types.




*Figure 12-20: Relative reference level*

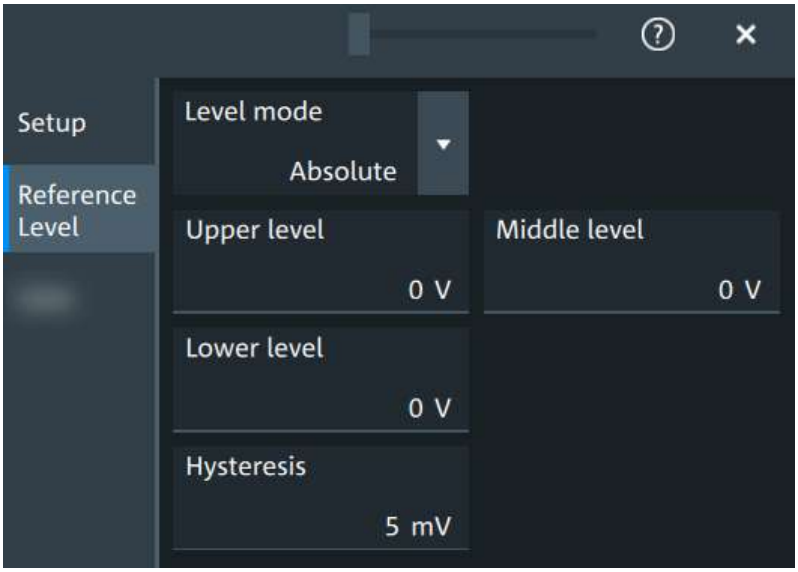



*Figure 12-21: Absolute reference level*

**Level mode**
Defines if the reference level is set in absolute or relative values.

Remote command:
`POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:LMODe` on page 1309
`POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:LMODe` on page 1309
`POWer<m>:HARMonics:REFLevel<rl>:LMODe` on page 1309
`POWer<m>:QUALity:REFLevel<rl>:LMODe` on page 1309
`P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:LMODe` on page 1351
`P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:LMODe` on page 1351
`P3PHase<m>:HARMonics:REFLevel<rl>:LMODe` on page 1351
`P3PHase<m>:QUALity:REFLevel<rl>:LMODe` on page 1351

**Absolute Level mode**
In the absolute mode, the reference values are set as voltages, in absolute values.

**Upper level, Middle level, Lower level ← Absolute Level mode**
Sets the low, middle and upper reference signal levels.

Remote command:
`POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:LLEVel` on page 1307
`POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:MLEVel` on page 1308
`POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:ULEVel` on page 1308
`POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:LLEVel` on page 1307
`POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:MLEVel` on page 1308

POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:ULEVel on page 1308
POWer<m>:HARMonics:REFLevel<rl>:ABSolute:LLEVel on page 1307
POWer<m>:HARMonics:REFLevel<rl>:ABSolute:MLEVel on page 1308
POWer<m>:HARMonics:REFLevel<rl>:ABSolute:ULEVel on page 1308
POWer<m>:QUALity:REFLevel<rl>:ABSolute:LLEVel on page 1307
POWer<m>:QUALity:REFLevel<rl>:ABSolute:MLEVel on page 1308
POWer<m>:QUALity:REFLevel<rl>:ABSolute:ULEVel on page 1308
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:LLEVel on page 1350
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:MLEVel on page 1351
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:ULEVel on page 1351
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:LLEVel on page 1350
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:MLEVel on page 1351
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:ULEVel on page 1351
P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:LLEVel on page 1350
P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:MLEVel on page 1351
P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:ULEVel on page 1351
P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:LLEVel on page 1350
P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:MLEVel on page 1351
P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:ULEVel on page 1351

**Relative Level mode**
In the relative mode, the reference values are set as percentages of the signal amplitude.

**Relative levels ← Relative Level mode**
Selects the lower, middle and upper reference levels, defined as percentages of the signal amplitude.

Available relative levels:

- 5/50/95
- 10/50/90
- 20/50/80
- User defined: Enter "Upper level", "Middle level", and "Lower level".

For example, for "5/50/95" the levels are set to the following values:

- Lower level = 5% of the signal amplitude
- Middle level = 50% of the signal amplitude
- Upper level = 95% of the signal amplitude

Remote command:
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MODE on page 1310
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MODE on page 1310
POWer<m>:HARMonics:REFLevel<rl>:RELative:MODE on page 1311
POWer<m>:QUALity:REFLevel<rl>:RELative:MODE on page 1311
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MODE on page 1353
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MODE on page 1353
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:MODE on page 1353
P3PHase<m>:QUALity:REFLevel<rl>:RELative:MODE on page 1353

**Upper level, Middle level,Lower level ← Relative Level mode**
Define the reference levels in percent, if "Level mode" is set to "User-defined".

Remote command:
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:LOWer on page 1309
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MIDDle on page 1310
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:UPPer on page 1311
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:LOWer on page 1309
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MIDDle on page 1310
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:UPPer on page 1311
POWer<m>:HARMonics:REFLevel<rl>:RELative:LOWer on page 1310
POWer<m>:HARMonics:REFLevel<rl>:RELative:MIDDle on page 1310
POWer<m>:HARMonics:REFLevel<rl>:RELative:UPPer on page 1311
POWer<m>:QUALity:REFLevel<rl>:RELative:LOWer on page 1310
POWer<m>:QUALity:REFLevel<rl>:RELative:MIDDle on page 1310
POWer<m>:QUALity:REFLevel<rl>:RELative:UPPer on page 1311
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:LOWer on page 1352
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MIDDle on page 1353
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:UPPer on page 1354
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:LOWer on page 1352
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MIDDle on page 1353
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:UPPer on page 1354
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:LOWer on page 1352

P3PHase<m>:HARMonics:REFLevel<rl>:RELative:MIDDle on page 1353
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:UPPer on page 1354
P3PHase<m>:QUALity:REFLevel<rl>:RELative:LOWer on page 1352
P3PHase<m>:QUALity:REFLevel<rl>:RELative:MIDDle on page 1353
P3PHase<m>:QUALity:REFLevel<rl>:RELative:UPPer on page 1354

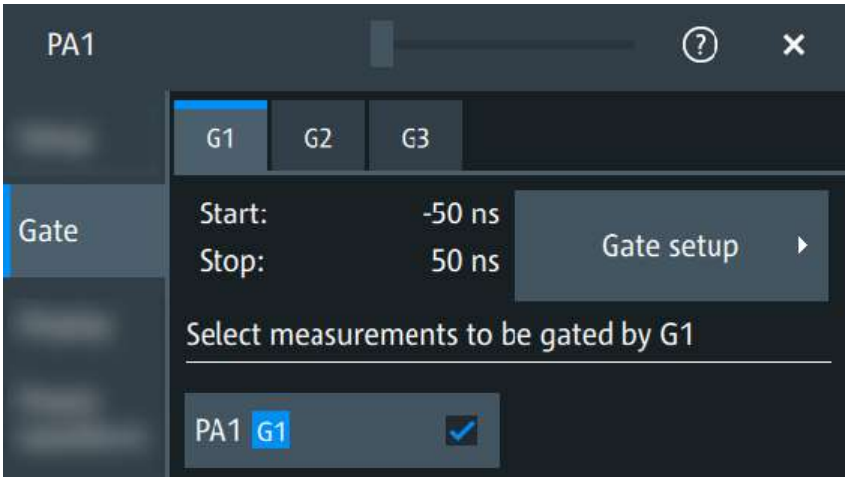
**Hysteresis**
Defines a hysteresis for the middle reference level. A rise or fall from the middle reference value that does not exceed the hysteresis is rejected as noise.

Remote command:
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:HYSTeresis on page 1307
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:HYSTeresis on page 1309
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:HYSTeresis on page 1307
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:HYSTeresis on page 1309
POWer<m>:HARMonics:REFLevel<rl>:ABSolute:HYSTeresis on page 1307
POWer<m>:HARMonics:REFLevel<rl>:RELative:HYSTeresis on page 1309
POWer<m>:QUALity:REFLevel<rl>:ABSolute:HYSTeresis on page 1307
POWer<m>:QUALity:REFLevel<rl>:RELative:HYSTeresis on page 1309
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:HYSTeresis on page 1350
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:HYSTeresis on page 1352
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:HYSTeresis on page 1350
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:HYSTeresis on page 1352
P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:HYSTeresis on page 1350
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:HYSTeresis on page 1352
P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:HYSTeresis on page 1350
P3PHase<m>:QUALity:REFLevel<rl>:RELative:HYSTeresis on page 1352

#### 12.3.8.2 Gate settings for power measurements

Access: Select power measurement > "Gate".

You can limit the some power measurement to a user-defined range of the waveform. In the "Gate" tab, you select the gate to be used for the selected power analysis. If no gate has been defined before, or the correct gate is missing, select "Gate setup" and configure the gate. See Section 8.2, "Gate setup", on page 214 for details.




Remote commands:

- POWer<m>:EFFiciency:GATE on page 1289
- POWer<m>:QUALity:GATE on page 1252
- P3PHase<m>:EFFiciency:GATE on page 1345
- P3PHase<m>:QUALity:GATE on page 1345

### 12.3.8.3 Power waveform

Some power analysis measurements create a power waveform as one of the measurement results.

In the "Power waveform" dialog, you can change the scaling of the power waveform.




**Display**
Displays or hides the power waveform.

Remote command:
POWer<m>:EFFiciency:DISPlay:WAVeform:INPut<n>:POWer[:ENABle]
on page 1304
POWer<m>:EFFiciency:DISPlay:WAVeform:OUTPut<n>:POWer[:ENABle]
on page 1304
POWer<m>:EFFiciency:DISPlay:WAVeform:TOTal<n>:OPOWer[:ENABle]
on page 1304

POWer<m>:HARMonics:DISPlay:POWer:WAVeform:ENABle on page 1304
POWer<m>:QUALity:DISPlay:POWer:WAVeform:ENABle on page 1304
POWer<m>:SWITching:DISPlay:POWer:WAVeform[:ENABle] on page 1304

**Scale mode**
Selects the scaling mode for display of the resulting waveform. By default, the instrument selects the most appropriate scale. In "Manual" mode, you can change the vertical scale and offset.

Remote command:
POWer<m>:EFFiciency:INPut<n>:POWer:VERTical:SCALe:MODE on page 1305
POWer<m>:EFFiciency:OUTPut<n>:POWer:VERTical:SCALe:MODE on page 1305
POWer<m>:EFFiciency:TOTal<n>:OPOWer:VERTical:SCALe:MODE on page 1305
POWer<m>:HARMonics:POWer:VERTical:SCALe:MODE on page 1305
POWer<m>:QUALity:POWer:VERTical:SCALe:MODE on page 1305
POWer<m>:SWITching:POWer:VERTical:SCALe:MODE on page 1305

**Scale**
Sets the vertical scale of the resulting power waveform if "Scale mode" is set to "Manual".

Remote command:
POWer<m>:EFFiciency:INPut<n>:POWer:VERTical:SCALe[:VALue] on page 1305
POWer<m>:EFFiciency:OUTPut<n>:POWer:VERTical:SCALe[:VALue] on page 1305
POWer<m>:EFFiciency:TOTal<n>:OPOWer:VERTical:SCALe[:VALue] on page 1305
POWer<m>:HARMonics:POWer:VERTical:SCALe[:VALue] on page 1305
POWer<m>:QUALity:POWer:VERTical:SCALe[:VALue] on page 1305
POWer<m>:SWITching:POWer:VERTical:SCALe[:VALue] on page 1305

**Offset**
Sets the offset of the resulting power waveform if "Scale mode" is set to "Manual".

Remote command:
POWer<m>:EFFiciency:INPut<n>:POWer:VERTical:OFFSet on page 1304
POWer<m>:EFFiciency:OUTPut<n>:POWer:VERTical:OFFSet on page 1304
POWer<m>:EFFiciency:TOTal<n>:OPOWer:VERTical:OFFSet on page 1304
POWer<m>:HARMonics:POWer:VERTical:OFFSet on page 1304
POWer<m>:QUALity:POWer:VERTical:OFFSet on page 1304
POWer<m>:SWITching:POWer:VERTical:OFFSet on page 1304

## 12.4 3-phase power analysis (option R&S MXO3-K333)

With the MXO 3 and option R&S MXO3-K333, you can perform 3-phase measurements.

For best measurement results, we recommended setting channel Bandwidth = 20 MHz.

If using the HD mode, we recommended setting Bandwidth = 100 MHz.

## 12.4.1 3-phase power analysis initial settings

In the main "3-phase" dialog, you select the required power measurements, align the channels and set the offset compensation to get correct measurements.

### 12.4.1.1 Select 3-phase measurements

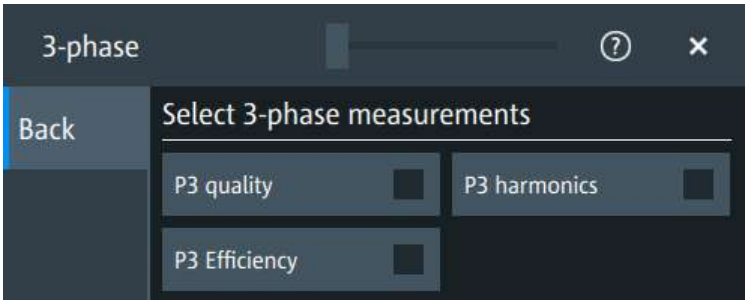
In the "Analysis" tab, you select the 3-phase measurements.

You can add up to 3 3-phase measurements, including several instances of the same measurement.

From here, you can access the settings for the enabled 3-phase measurements.

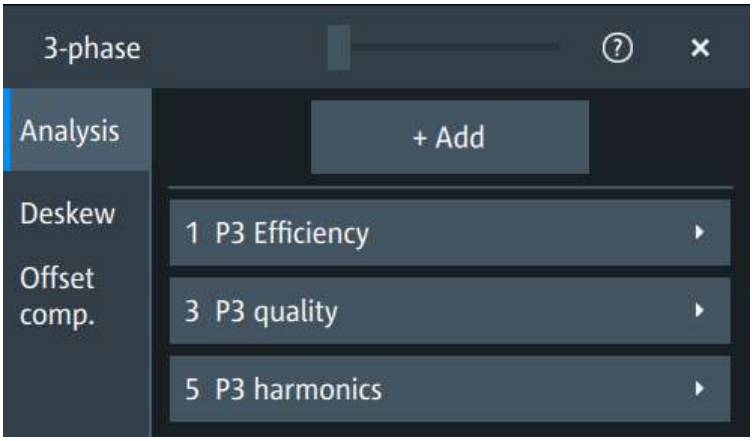
**To add a new 3-phase measurement**

1. Select the "Menu" > "Apps" > "General" tab > "3-phase".
2. In the "Analysis" tab, select "+ Add".




3. Select a 3-phase measurement on the list to enable it.
4. Select "+ Add" to add the selected measurements.

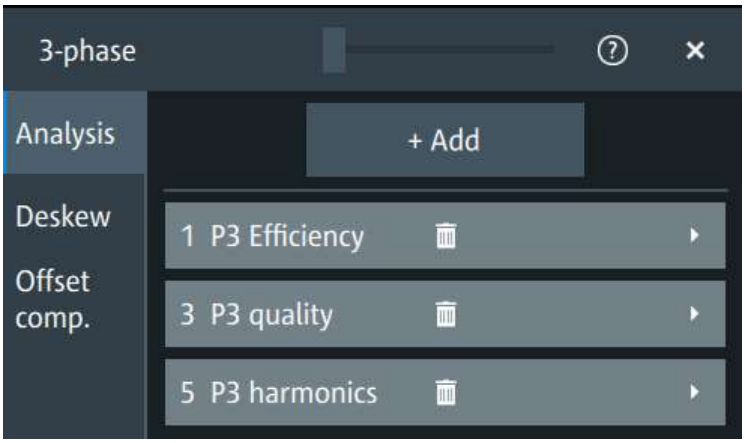
   All selected measurements are enabled.




**To delete a 3-phase measurement**

1. Select the "Menu" > "Apps" > "General" tab > "3-phase", to open the 3-phase measurement dialog.
2. In the "Analysis" tab, select .

   A delete icon appears next to each measurement.




3. Select the power measurement that you want to delete. Alternatively, select "Delete All" to delete all measurements.

**+ Add**
Opens a dialog to select the 3-phase measurements. See "To add a new 3-phase measurement" on page 417.

Remote command:
`P3PHase<m>[:ENABle]` on page 1314
`P3PHase<m>:TYPE` on page 1314

### 12.4.1.2 Deskew

See Section 12.3.1.2, "Deskew", on page 371.

#### 12.4.1.3 Offset compensation

See Section 12.3.1.3, "Offset compensation", on page 372.

### 12.4.2 Common functions for 3-phase power measurements

#### 12.4.2.1 3-phase power sources

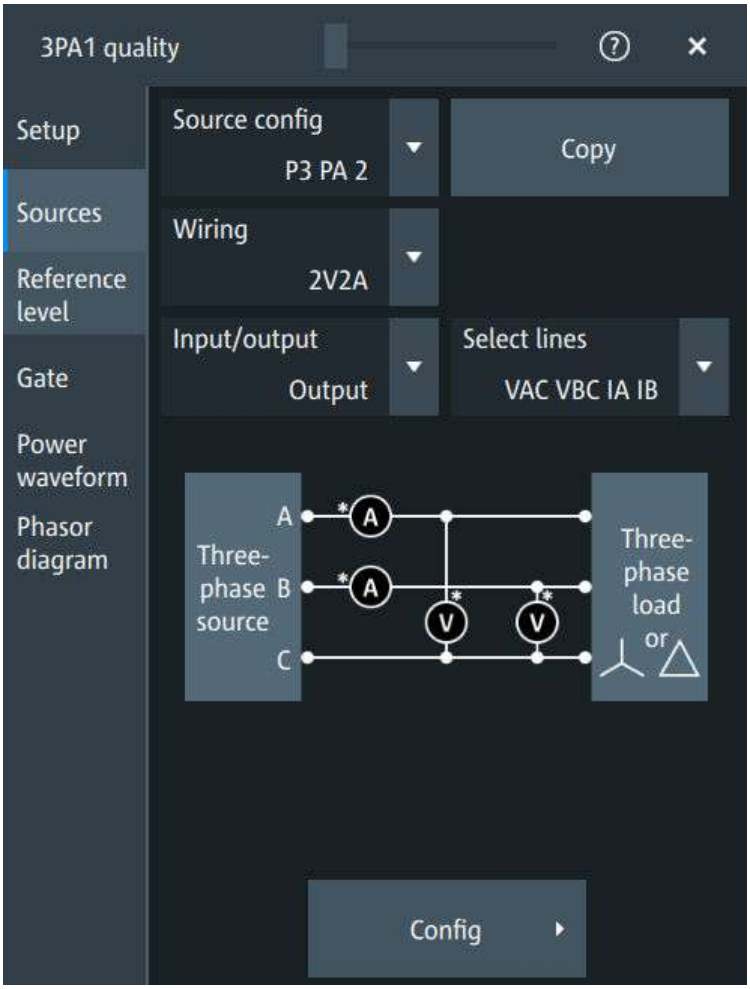



*Figure 12-22: 3-phase source configuration for the quality and harmonics measurements*

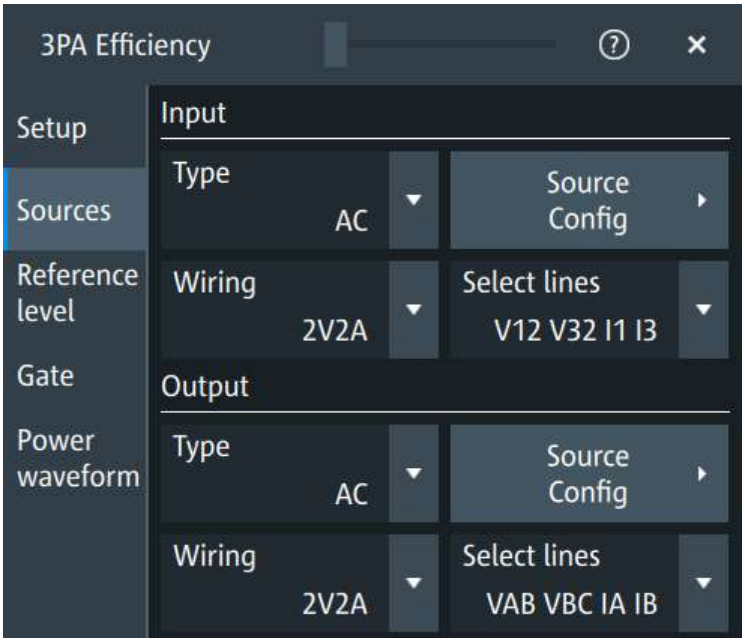



*Figure 12-23: 3-phase source configuration for the efficiency measurement*

**Source config**
Select a source configuration previously defined for another 3-phase power measurement. Only available for the quality and harmonics measurements.

**Copy**
Copies the selected source configuration to the current configuration. Only available for the quality and harmonics measurements.

**Type**
Selects the type of the current flow for the input and output channel: AC or DC. Only available for the efficiency measurement.

Remote command:
`P3PHase<m>:EFFiciency:INPut<n>:TYPE` on page 1342
`P3PHase<m>:EFFiciency:OUTPut<n>:TYPE` on page 1342

**Wiring**
Configures how you connect the probes to the three-phase device. For quality and harmonics measurements, this setting also shows the corresponding connection diagram for correct setup.

- "1V1A": Measures one line-to-line voltage and its corresponding phase current; the instrument assumes a single-phase system.
- "2V2A": Measures two line-to-line voltages and two phase currents; the third voltage and current are derived and the full three-phase power is computed.
- "3V3A": Measures the voltages and currents of all three phases directly
- "3VN3A": The voltages are measured against the neutral line and the currents in the individual phases.

Remote command:
P3PHase<m>:EFFiciency:INPut<n>:SOURce:WIRing on page 1342
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:WIRing on page 1342
P3PHase<m>:HARMonics:SOURce:WIRing on page 1342
P3PHase<m>:QUALity:SOURce:WIRing on page 1342

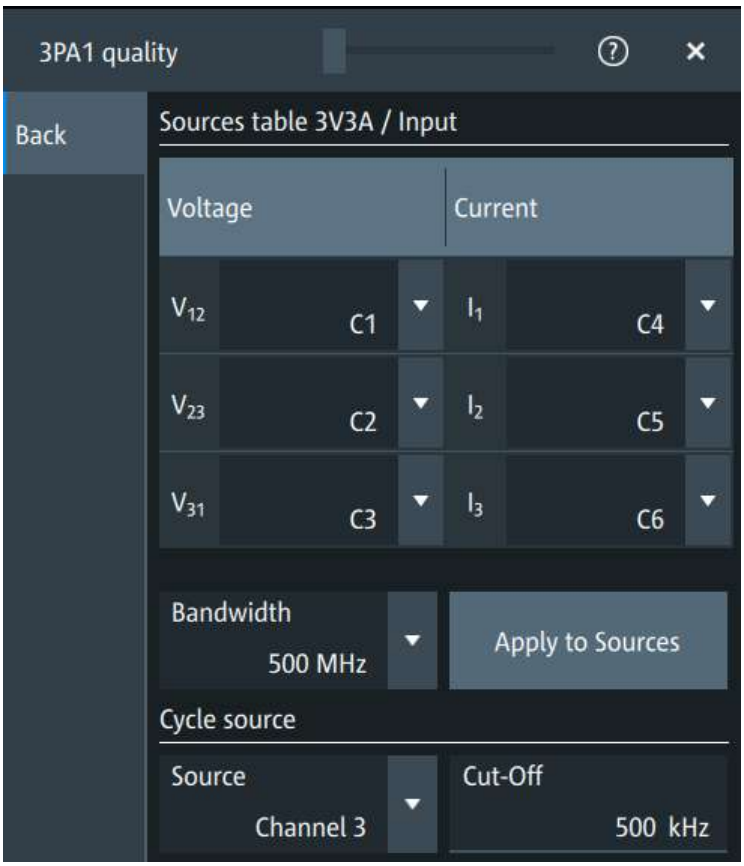
**Input/output**
Configures whether the oscilloscope is configured to receive signals (input) for analysis, or to generate signals output) as a source for testing. Only available for the quality and harmonics measurements.

Remote command:
P3PHase<m>:HARMonics:SOURce:INOut on page 1342
P3PHase<m>:QUALity:SOURce:INOut on page 1342

**Select lines**
Select the configuration for your probe setup. The visualization is adapted accordingly. This setting is only available for "Wiring"="2V2A".

Remote command:
P3PHase<m>:EFFiciency:INPut<n>:SOURce:LINes:W2V2a on page 1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:LINes:W2V2a on page 1343
P3PHase<m>:HARMonics:SOURce:LINes:W2V2a on page 1343
P3PHase<m>:QUALity:SOURce:LINes:W2V2a on page 1343

**Config**
Opens the "Source config" dialog.




**Voltage ← Config**
Selects the voltage source waveform. Analog channels and math waveforms can be used.

Remote command:
`P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W1V1a` on page 1344
`P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W2V2a` on page 1344
`P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W3V3a` on page 1344
`P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W3VN3a` on page 1344
`P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W1V1a` on page 1344
`P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W2V2a` on page 1344
`P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W3V3a` on page 1344
`P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W3VN3a` on page 1344
`P3PHase<m>:HARMonics:SOURce:VOLTage:W2V2a` on page 1344
`P3PHase<m>:HARMonics:SOURce:VOLTage:W3V3a` on page 1344
`P3PHase<m>:HARMonics:SOURce:VOLTage:W3VN3a` on page 1344
`P3PHase<m>:QUALity:SOURce:VOLTage:W2V2a` on page 1344
`P3PHase<m>:QUALity:SOURce:VOLTage:W3V3a` on page 1344
`P3PHase<m>:QUALity:SOURce:VOLTage:W3VN3a` on page 1344

**Current ← Config**
Selects the current source waveform. Analog channels and math waveforms can be used.

Remote command:
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W1V1a on page 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W2V2a on page 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W3V3a on page 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W3VN3a on page 1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W1V1a on page 1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W2V2a on page 1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W3V3a on page 1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W3VN3a on page 1343
P3PHase<m>:HARMonics:SOURce:CURRent:W2V2a on page 1343
P3PHase<m>:HARMonics:SOURce:CURRent:W3V3a on page 1343
P3PHase<m>:HARMonics:SOURce:CURRent:W3VN3a on page 1343
P3PHase<m>:QUALity:SOURce:CURRent:W2V2a on page 1343
P3PHase<m>:QUALity:SOURce:CURRent:W3V3a on page 1343
P3PHase<m>:QUALity:SOURce:CURRent:W3VN3a on page 1343

**Bandwidth ← Config**
If a warning message appears indicating differing bandwidths on the input channels, verify and adjust the bandwidth settings to match across all channels.

**Apply to Sources ← Config**
Applies the configured bandwidth setting to all sources.

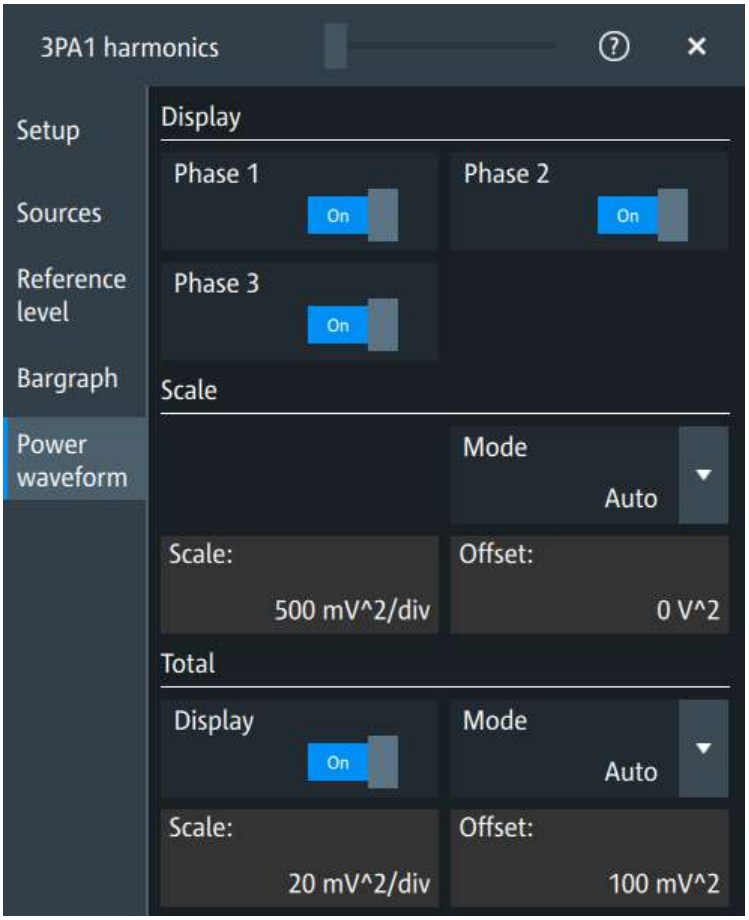
**Source ← Config**
Selects the channel (voltage source) to serve as the timing reference for data calculations across all sources when the periods are not aligned on the x-axis.

Remote command:
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CYCLe:SOURce on page 1344
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CYCLe:SOURce on page 1344
P3PHase<m>:HARMonics:SOURce:CYCLe[:SOURce] on page 1344
P3PHase<m>:QUALity:SOURce:CYCLe[:SOURce] on page 1344

**Cut-off ← Config**
Configures a bandwidth limitation. The adjustable filter frequency is linked to the sample rate. To access a wider range of frequencies not achievable with the current configuration, configure the sample rate in "Menu" > "Acquisition".

Remote command:
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CYCLe:FILTer on page 1345
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CYCLe:FILTer on page 1345
P3PHase<m>:HARMonics:SOURce:CYCLe:FILTer on page 1345
P3PHase<m>:QUALity:SOURce:CYCLe:FILTer on page 1345

#### 12.4.2.2 3-phase power waveform




In the "Power waveform" dialog, you can change the display and scaling of the power waveform of the single and total phases.

**Display individual phases**
Displays or hides the power waveform of the individual phases.

For harmonics analysis, a power waveform is available if "Standard" is set to EN 61000-3-2 Class C or EN 61000-3-2 Class D.

Remote command:
`P3PHase<m>:QUALity:DISPlay:POWer<n>:WAVeform[:ENABle]` on page 1346
`P3PHase<m>:HARMonics:DISPlay:POWer<n>:WAVeform[:ENABle]` on page 1346

**Scale individual phases**
Configures the scaling of the individual phases.

**Mode ← Scale individual phases**
Selects the scaling mode for display of the resulting waveform. By default, the instrument selects the most appropriate scale. In "Manual" mode, you can change the vertical scale and offset.

Remote command:
P3PHase<m>:QUALity:POWer<n>:VERTical:SCALe:MODE on page 1347
P3PHase<m>:HARMonics:POWer<n>:VERTical:SCALe:MODE on page 1347

**Scale ← Scale individual phases**
Sets the vertical scale of the resulting power waveform if "Mode" is set to "Manual".

Remote command:
P3PHase<m>:QUALity:POWer<n>:VERTical:SCALe[:VALue] on page 1348
P3PHase<m>:HARMonics:POWer<n>:VERTical:SCALe[:VALue] on page 1348

**Offset ← Scale individual phases**
Sets the offset of the resulting power waveform if "Mode" is set to "Manual".

Remote command:
P3PHase<m>:QUALity:POWer<n>:VERTical:OFFSet on page 1346
P3PHase<m>:HARMonics:TOTal:POWer:VERTical:OFFSet on page 1347

**Total phase**
Configures the display and the scaling of the total phase.

**Display ← Total phase**
Enables the display of the total phase.

Remote command:
P3PHase<m>:EFFiciency:DISPlay:WAVeform:INPut:TOTal:POWer[:ENABle] on page 1346
P3PHase<m>:EFFiciency:DISPlay:WAVeform:OUTPut:TOTal:POWer[:ENABle] on page 1346
P3PHase<m>:QUALity:DISPlay:TOTal:POWer:WAVeform[:ENABle] on page 1346
P3PHase<m>:HARMonics:DISPlay:TOTal:POWer:WAVeform[:ENABle] on page 1346

**Mode ← Total phase**
Selects the scaling mode for display of the resulting waveform. By default, the instrument selects the most appropriate scale. In "Manual" mode, you can change the vertical scale and offset.

Remote command:
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:VERTical:SCALe:MODE on page 1348
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:VERTical:SCALe:MODE on page 1348
P3PHase<m>:QUALity:TOTal:POWer:VERTical:SCALe:MODE on page 1348
P3PHase<m>:HARMonics:TOTal:POWer:VERTical:SCALe:MODE on page 1348

**Scale ← Total phase**
Sets the vertical scale of the resulting power waveform if "Mode" is set to "Manual".

Remote command:
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:VERTical:SCALe[:VALue] on page 1348
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:VERTical:SCALe[:VALue] on page 1348
P3PHase<m>:QUALity:TOTal:POWer:VERTical:SCALe[:VALue] on page 1348
P3PHase<m>:HARMonics:TOTal:POWer:VERTical:SCALe[:VALue] on page 1348

**Offset ← Total phase**
Sets the offset of the resulting power waveform if "Mode" is set to "Manual".

Remote command:
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:VERTical:OFFSet on page 1347
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:VERTical:OFFSet on page 1347
P3PHase<m>:QUALity:TOTal:POWer:VERTical:OFFSet on page 1347
P3PHase<m>:HARMonics:TOTal:POWer:VERTical:OFFSet on page 1347

## 12.4.3 3-phase power quality

Expanding on single-phase power quality principles, three-phase systems require dedicated analysis due to inherent phase shifts and potential imbalances. See also Section 12.3.2, "Power quality", on page 373. While active, reactive, and apparent powers remain key metrics, three-phase assessment focuses on inter-phase relationships.

The power quality analysis measurement provides per-phase data ("C1"-"C6") alongside total system values ("V", "A"), see Section 12.4.3.8, "3-phase power quality results", on page 430. Evaluating these metrics for each phase helps identify imbalances or anomalies. Optimal systems exhibit minimal inter-phase variations and low harmonic distortion.

Required probes:

- 3x differential voltage probes (one for each phase)
- 3x current probes (one for each phase)

### 12.4.3.1 Configuring 3-phase power quality

For details of the configuration settings, see:

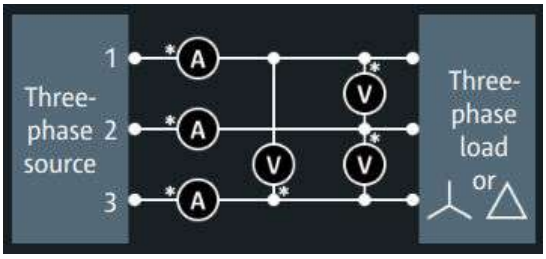
- Section 12.4.3.2, "3-phase power quality setup", on page 427.
- Section 12.4.3.3, "3-phase power sources", on page 428

1. Add a new power quality measurement as described in Section 12.4.1, "3-phase power analysis initial settings", on page 417.
2. Connect the differential voltage and the current probes to the oscilloscope.
3. Deskew the probes as described in Section 12.3.1.2, "Deskew", on page 371.

4. In the "Analysis" tab, select the "P3 quality" measurement.

   The "P3 quality" setup dialog opens.

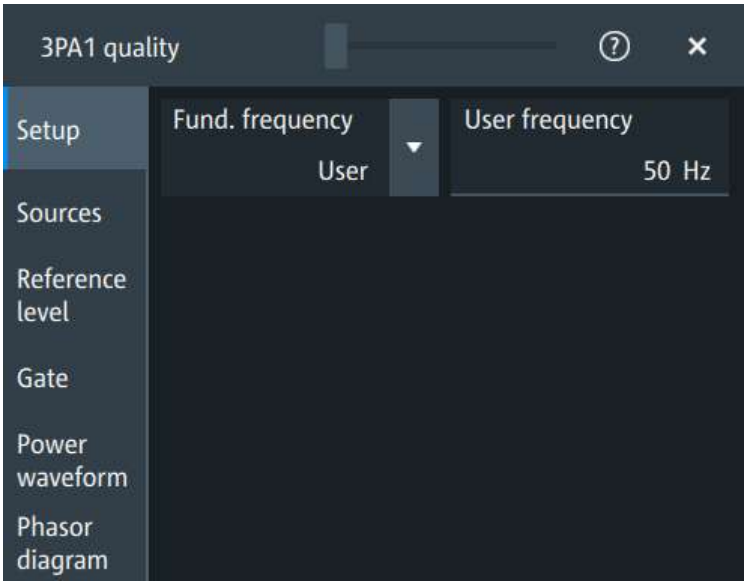
5. Select the "Fund. frequency" according to your signal.
6. Set a "User Frequency" to set the horizontal timebase.
7. Select the "Sources" tab.
8. Select the "Wiring" and if necessary "Select lines" to configure how your probes are connected to the three-phase device.
9. Select if you have an "Input/output" setup.
10. Select "Config".
11. Assign the correct channels for the "Voltage" and the "Current" probes.
12. Select the required "Bandwidth" for signal acquisition. Select "Apply to Sources" to apply the configured bandwidth setting to all sources.
13. Select "Back".
14. Connect the probes to the DUT according to the visualization of the setup. For example a "3V3A" wiring:




   On the screen, you can see the measurement waveforms of the current, the voltage and the power. Also, the result table with numeric measurement results is shown. For details, see Section 12.4.3.8, "3-phase power quality results", on page 430.

### 12.4.3.2 3-phase power quality setup

Access: "Menu" > "Apps" > "3-phase" > "+ Add" > "P3 quality" > select "P3 quality" > "Setup".




In the "Setup" tab, you configure the frequencies.

**Fund. frequency**
Selects the input frequency of the source signal.

- "50 Hz": Standard power grid frequency in Europe, Asia, and parts of Africa
- "60 Hz": Standard power grid frequency in North America and parts of South America
- "400 Hz": Used in aerospace and submarine applications and some industrial equipment
- "User": Custom fundamental frequency value
- "Automatic": Automatic detection of the fundamental frequency of the input signal

Remote command:
`P3PHase<m>:QUALity:FREQuency[:VALue]` on page 1315

**User Frequency**
Sets the horizontal timebase to control the time scale of the waveform display.

Remote command:
`P3PHase<m>:QUALity:FREQuency:USER` on page 1315

### 12.4.3.3 3-phase power sources

See Section 12.4.2.1, "3-phase power sources", on page 419.

### 12.4.3.4 3-phase power reference level

See Section 12.3.8.1, "Reference level", on page 410.

#### 12.4.3.5 3-phase power gate

See Section 12.3.8.2, "Gate settings for power measurements", on page 414.

#### 12.4.3.6 3-phase power waveform

For a description of the power waveform scaling, see Section 12.4.2.2, "3-phase power waveform", on page 424.

#### 12.4.3.7 3-phase phasor diagram

Access: "Menu" > "Apps" > "3-phase" > "+ Add" > "P3 quality" > select "P3 quality" > "Phasor diagram".




In the "Phasor diagram" tab, you configure the display of the phasor diagram.

**State**
Displays or hides the phasor diagram.

Remote command:
`P3PHase<m>:QUALity:PHASor[:ENABle]` on page 1316

**Voltage range, Current range**
Configures the voltage and current ranges of the phasor diagram.

**Offset ← Voltage range, Current range**
Sets the inner point of the phasor diagram.

Remote command:
P3PHase<m>:QUALity:PHASor:VOLTage:OFFSet on page 1316
P3PHase<m>:QUALity:PHASor:CURRent:OFFSet on page 1316

**Maximum ← Voltage range, Current range**
Sets the outer circumference of the phasor diagram.

Remote command:
P3PHase<m>:QUALity:PHASor:VOLTage:MAXimum on page 1316
P3PHase<m>:QUALity:PHASor:CURRent:MAXimum on page 1316

**Fit to Screen**
The instrument selects the most appropriate scale for the voltage and current phasors.

Remote command:
P3PHase<m>:QUALity:PHASor:FSCReen on page 1317

### 12.4.3.8 3-phase power quality results

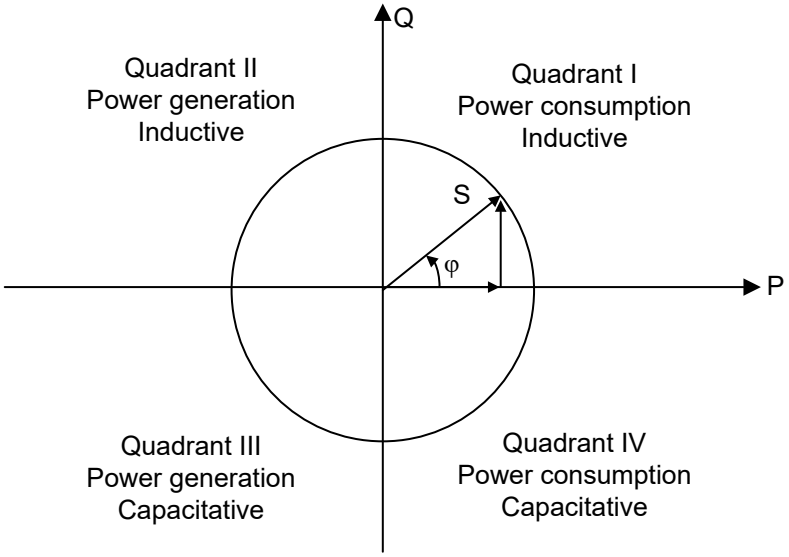



*Figure 12-24: 3-phase power quality measurement of a symmetrical inductive load with balanced phase voltages, currents, and power values.*




*Figure 12-25: 3-phase power quality measurement with one phase in the third quadrant, showing negative active power and negative reactive power.*

The operating condition of each phase can be interpreted with the four-quadrant analysis of active power and reactive power. Positive active power indicates power consumption, while negative active power indicates power generation. Positive reactive power indicates inductive behavior, while negative reactive power indicates capacitive behavior. The quadrant therefore identifies whether a phase operates as a consumer or source and whether the reactive component is inductive or capacitive.



The results of power quality measurements are provided as follows:

- The waveform diagram shows the graphical presentation of:
  - The voltage waveform of the individual and total phases
  - The current waveform of the individual and total phases
  - The power waveform of the individual and total phases, The power waveform is the product of the current and voltage waveforms.
    See also Section 12.4.2.2, "3-phase power waveform", on page 424.
- Results are visualized in a phasor diagram, which graphically displays the magnitude and phase angles of each phase voltage and current. In a balanced 3-phase system, the phase voltages are typically spaced by 120 degrees, and the phase currents are also typically spaced by 120 degrees.
  See also Section 12.4.3.7, "3-phase phasor diagram", on page 429.
- The result table displays the numeric measurement results. The total power values are the sum of the individual phase power values:

| Result table | Description |
|---|---|
| Cycle source | Channel (voltage) used as the timing reference for all measurements. |
| Cycle source frequency | Nominal (rated) system frequency, e.g., 50 Hz or 60 Hz. |
| Input, output | Selected lines in the source configuration |
| Current RMS, voltage RMS ("CRMS") | Square root of the mean of the square of the current or voltage. Represents the effective value of the AC waveform. |
| Current cycle crest factor, voltage cycle crest factor ("CCrest") | Ratio of the peak (maximum absolute value) to the RMS value of the waveform. Indicates the level of waveform distortion. |
| Current maximum, voltage maximum | Highest measured value of the current or voltage |
| Active power (P) | Real power that can be converted into work. The result can be positive or negative. Positive values indicate active power consumption. Negative values indicate active power generation. |
| Reactive power (Q) | Power flow that is temporally stored in a system because of the inductive and capacitive elements.<br>For the individual phase values, Q<0 indicates a capacitive load, Q>0 indicates an inductive load. The total reactive power is always positive because of $Q_{TOT} = \sqrt{(S_{TOT}^2 - P_{TOT}^2)}$. |
| Apparent power (S) | Magnitude of the vector sum of active and reactive power |
| Power factor | Measure of the system efficiency. The value varies between -1 and 1. Positive values indicate active power consumption. Negative values indicate active power generation. |
| Phase (φ) | Phase differences between the current and the voltage sine waves of the different phases.<br>For the individual phase values, φ<0 indicates a capacitive load, φ>0 indicates an inductive load.<br>The total phase angle has the same sign convention as $Q_{TOT}$. |

Remote commands: "3-phase power quality results" on page 1321.

### 12.4.4 3-phase power harmonics

Harmonic currents generated by nonlinear loads in 3-phase systems can impact power quality by affecting other equipment on the grid, see Section 12.3.3, "Power harmonics", on page 379. Unlike single-phase systems, harmonics in a 3-phase environment can exist with different phase angles and magnitudes on each line, requiring analysis of both individual harmonic components and their relationships to each other. To ensure power quality and prevent interference, industry and regulatory bodies have established compliance standards for end-products.

The power harmonics analysis tests the devices according to the precompliance standard EN 61000-3-2.

Required probes:

- 3x differential voltage probes (one for each phase)
- 3x current probes (one for each phase + optionally a neutral current probe)

#### 12.4.4.1 Configuring 3-phase power harmonics

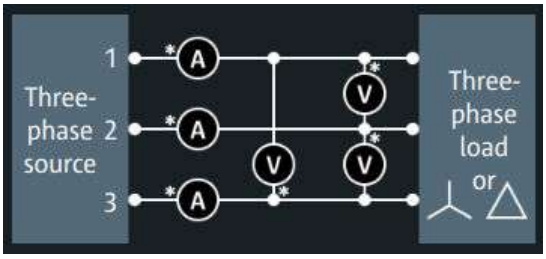
For details of the configuration settings, see:

- Section 12.4.4.2, "3-phase power harmonics setup", on page 434
- Section 12.4.4.3, "3-phase power sources", on page 436

1. Add a new power quality measurement as described in Section 12.4.1, "3-phase power analysis initial settings", on page 417.
2. Connect the differential voltage and the current probes to the oscilloscope.
3. Deskew the probes as described in Section 12.3.1.2, "Deskew", on page 371.
4. In the "Analysis" tab, select the "P3 harmonics" measurement.

   The "P3 harmonics" setup dialog opens.
5. Select the "Standard".
6. Select the "Fund. frequency" according to your signal.
7. Select the "Sources" tab.
8. Select the "Wiring" and if necessary "Select lines" to configure how your probes are connected to the three-phase device.
9. Select if you have an"Input/output" setup.
10. Select "Config".
11. Assign the correct channels for the "Voltage" and the "Current" probes.
12. Select the desired "Bandwidth" for signal acquisition. Select "Apply to Sources" to apply the configured bandwidth setting to all sources.
13. Select "Back".

14. Connect the probes to the DUT according to the visualization of the setup. For example a "3V3A" wiring:




On the screen, you can see the measurement waveforms of the current, the voltage and the power. Also, the result table with numeric measurement results is shown. For details, see Section 12.4.4.7, "3-phase power harmonic results", on page 438.

### 12.4.4.2 3-phase power harmonics setup

Access: "Menu" > "Apps" > "3-phase" > "+ Add" > "P3 harmonics" > select "P3 harmonics" > "Setup".

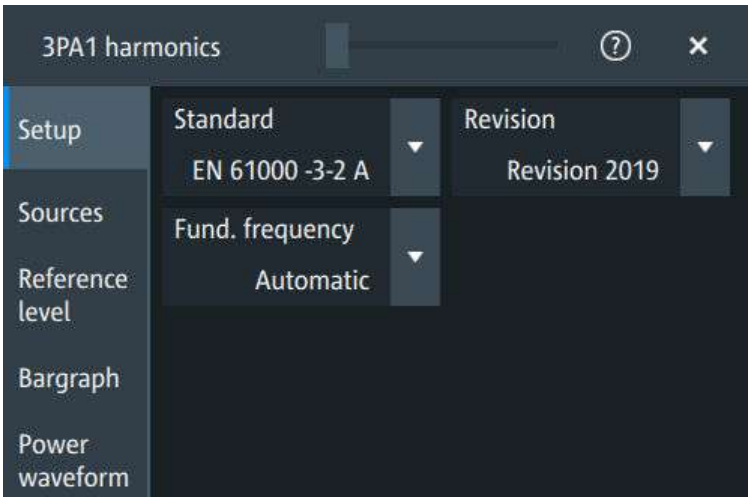



*Figure 12-26: 3-phase power harmonic setup for EN 61000-3-2 A / EN 61000-3-2 B*

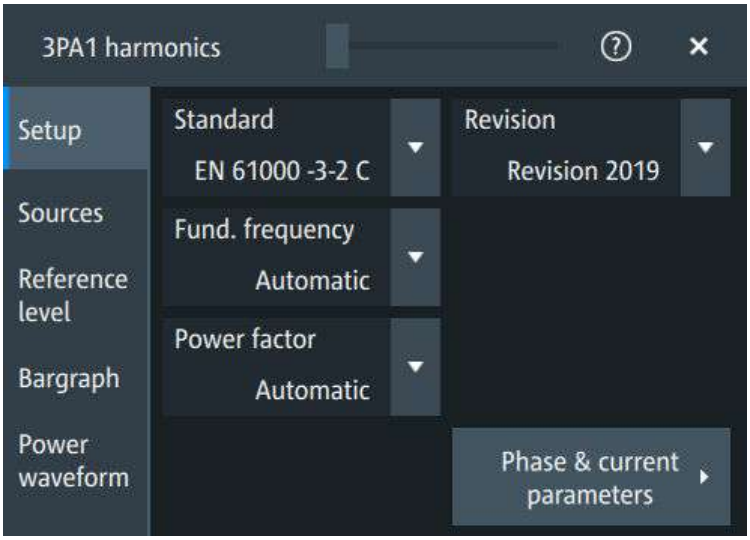



*Figure 12-27: 3-phase power harmonic setup for EN 61000-3-2 C*

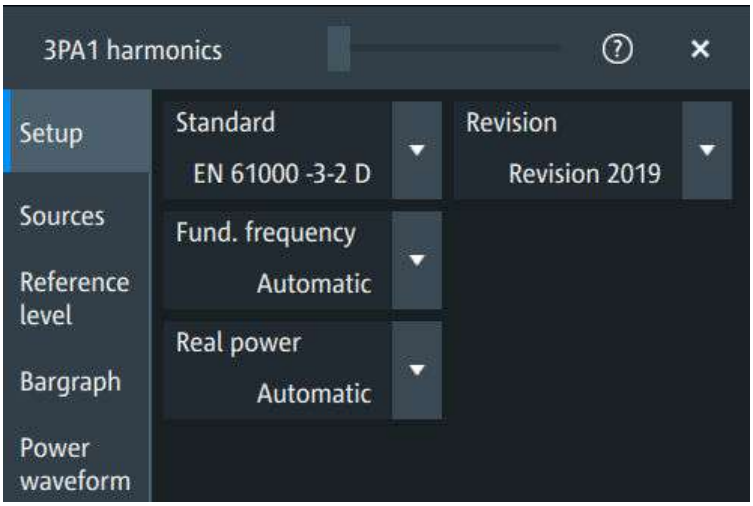



*Figure 12-28: 3-phase power harmonic setup for EN 61000-3-2 D*

### Standard, Revision

Select the standard in use, including which "Revision" of the standard is used, "Revision 2011" or "Revision 2019".

*Table 12-3: Current harmonic precompliance standards*

| Standard | Application |
|---|---|
| EN 61000-3-2 Class A | Balanced 3-phase equipment, household appliances (excluding equipment identified as class D), tools (excluding portable tools), dimmers for incandescent lamps, audio equipment |
| EN 61000-3-2 Class B | Portable tools, not professional arc welding equipment |
| EN 61000-3-2 Class C[1)] | Lighting equipment |
| EN 61000-3-2 Class D | PC, PC monitors, radio, or TV receivers with an input power less than or equal to 600W |

| Standard | Application |
|---|---|
| MIL-STD-1399 | Military shipboard user equipment |
| [1] The EN 61000-3-2 defines different limits for EN 61000-3-2 Class C equipment with an input power smaller or equal than 25W either. The limits that are implemented in the MXO 3 firmware comply with the limits of table 3 (column 2) of the EN 61000-3-2 standard. | |

Remote command:
`P3PHase<m>:HARMonics:STANdard` on page 1329
`P3PHase<m>:HARMonics:REVision` on page 1330

**Fund. frequency**
Selects the frequency of the input signal. The available frequencies depend on the selected Standard, Revision.

Remote command:
`P3PHase<m>:HARMonics:FREQuency:EN` on page 1330
`P3PHase<m>:HARMonics:FREQuency:MIL` on page 1330

**Power factor**
Available only for "Standard" = EN 61000-3-2 Class C.

Selects if the power factor is defined automatically, or a user-defined value ("Power factor") is used.

Remote command:
`P3PHase<m>:HARMonics:PFACtor:MODE` on page 1331
`P3PHase<m>:HARMonics:PFACtor:USER` on page 1331

**Phase & current parameters**
Shows a graphical explanation of the relative phase angle and current parameters.

**Real power, Power**
Available only for "Standard" = EN 61000-3-2 Class D.

Selects if the real power is defined automatically, or a user-defined value ("Power") is used.

Remote command:
`P3PHase<m>:HARMonics:RPOWer:MODE` on page 1331
`P3PHase<m>:HARMonics:RPOWer:USER` on page 1332

### 12.4.4.3 3-phase power sources

See Section 12.4.2.1, "3-phase power sources", on page 419.

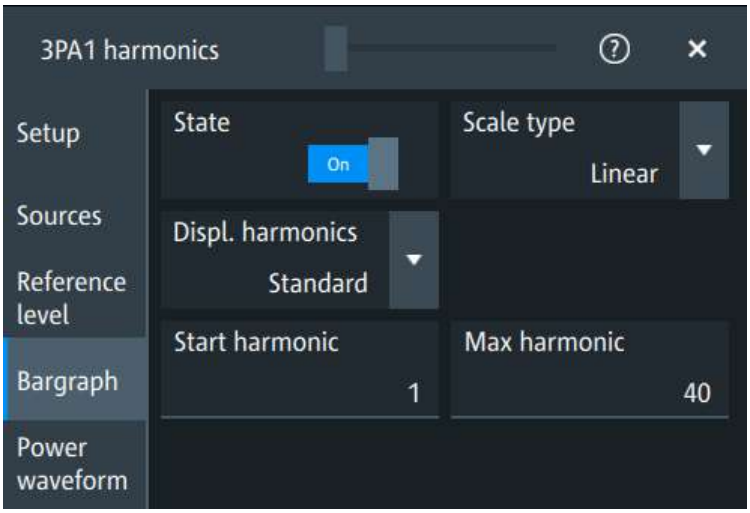
### 12.4.4.4 3-phase power reference level

See Section 12.3.8.1, "Reference level", on page 410.

#### 12.4.4.5 3-phase power bar graph

Access: "Menu" > "Apps" > "3-phase" > "+ Add" > "P3 harmonics" > select "P3 harmonics" > "Bargraph".

In this tab you can configure the scale for the bar graph display.




**State**
Displays or hides the bar graph.

Remote command:
`P3PHase<m>:HARMonics:DISPlay:BARGraph[:ENABle]` on page 1332

**Scale type**
Selects a logarithmic or linear scale for the display for the harmonics bar graph.

Remote command:
`P3PHase<m>:HARMonics:DISPlay:BARGraph:VERTical:SCALe:TYPE` on page 1333

**Displ. harmonics**
Selects which harmonics are displayed in the bar graph: all, odd even or depending on the standard definition.

Remote command:
`P3PHase<m>:HARMonics:DISPlay:HARMonics[:DISPlay]` on page 1332

**Start harmonic**
Sets the start frequency of a bar graph display. At least three bars are displayed.

Remote command:
`P3PHase<m>:HARMonics:DISPlay:HARMonics:STARt[:VALue]` on page 1333

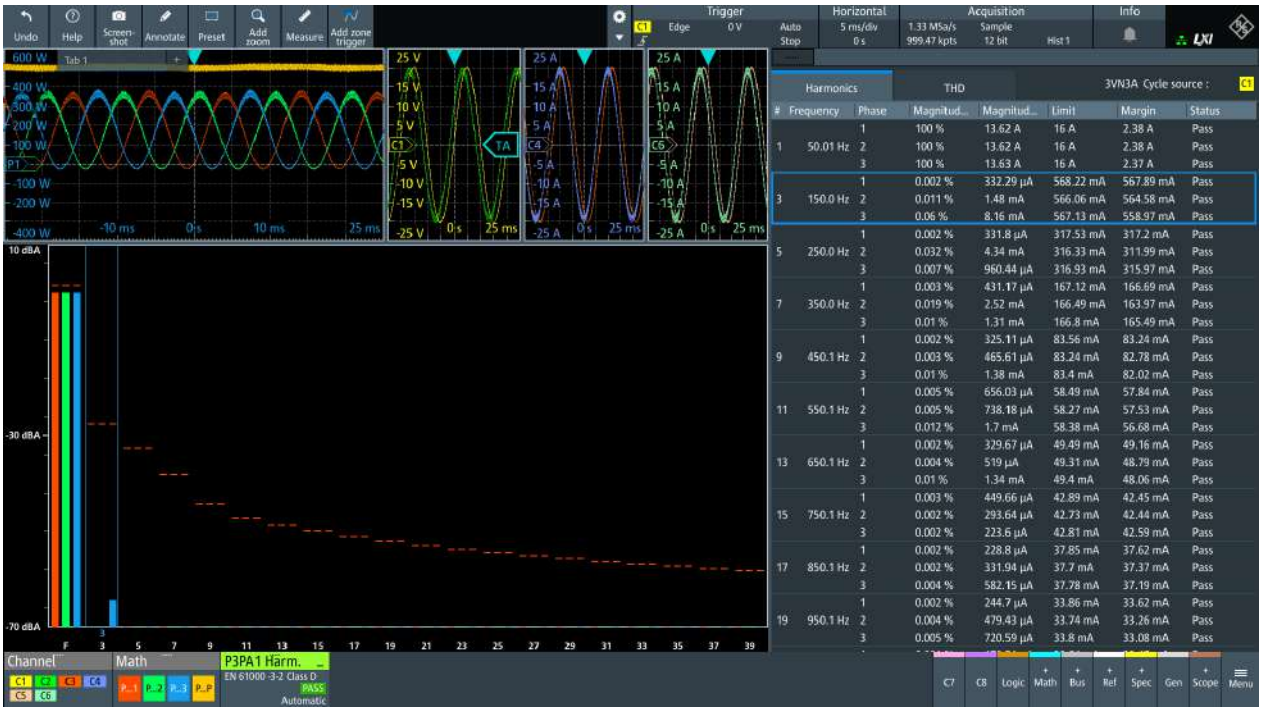
**Max harmonic**
Sets the stop frequency of the bar graph display. The maximum value is defined by standard and fundamental frequency.

Remote command:
P3PHase<m>:HARMonics:DISPlay:HARMonics:STOP[:VALue] on page 1333

### 12.4.4.6 3-phase power waveform

For a description of the power waveform scaling, see Section 12.4.2.2, "3-phase power waveform", on page 424.

### 12.4.4.7 3-phase power harmonic results




The results of the harmonic measurements are provided as follows:

- The waveform diagram shows the graphical presentation of:
  - The voltage waveform of the individual and total phases
  - The current waveform of the individual and total phases
  - The power waveform of the individual and total phases. The power waveform is the product of the current and voltage waveforms.
    See also Section 12.4.2.2, "3-phase power waveform", on page 424.
- The results are visualized in a bar graph. The number of displayed harmonics depends on the standard definition. When you select a harmonic in the bar chart, the corresponding row in the result table is highlighted.
  See also Section 12.4.4.5, "3-phase power bar graph", on page 437.
- The result table displays the numeric measurement results:

| Result table | Description |
|---|---|
| THD RMS | Total harmonic distortion (THD) relative to the RMS amplitude |
| THD fundamental | Total harmonic distortion relative to the fundamental |
| Harmonic index | Harmonic order. Represents the multiple of the fundamental frequency.<br>Value of the x-axis in the bar graph |
| Frequency | Nominal frequency of the power system. |
| Phase | Phase differences between the current and the voltage sine waves of the different phases. |
| Magnitude (%) | Present value of the current harmonic expressed as a percentage of the fundamental component.<br>Value of the y-axis in the bar graph |
| Magnitude (RMS) | Measured RMS value of the harmonic component. |
| Limit | Maximum allowed value according to the selected standard.<br>Visually represented by a dark orange line in the bar graph. |
| Margin | Sets the upper limit for the display of the Y scale. |
| Status | Displays whether the harmonic's magnitude is within the limit (pass - green bar) or exceeds the limit (fail - orange bar). |

Remote commands: "3-phase power harmonics results" on page 1334.

## 12.4.5 3-phase power efficiency

In a standard 3-phase drive setup, a single-phase AC or DC supply is converted into a 3-phase PWM output by a microcontroller-driven inverter. Real-time feedback such as speed, torque, or position is continuously read and used to modulate the inverter's switching pattern for precise motor control.

The 3-phase efficiency analysis determines how efficiently a 3-phase device converts total input power into total useful output power. The method is identical to the single-phase analysis, but the results from the three phases are summed to obtain the overall input, output and efficiency of the device. See also Section 12.3.6, "Power efficiency", on page 397.

Required probes for an example "1V-1A" input to a "2V-2A" output configuration:

- 3x differential voltage probes
- 3x current probes

### 12.4.5.1 Configuring 3-phase power efficiency

For details of the configuration settings, see:

- Section 12.4.5.2, "3-phase power efficiency setup", on page 441.
- Section 12.4.2.1, "3-phase power sources", on page 419

1. Add a power efficiency measurement as described in Section 12.4.1, "3-phase power analysis initial settings", on page 417.
2. Connect the differential voltage probes and the current probes to the oscilloscope.
3. Deskew the probes as described in Section 12.3.1.2, "Deskew", on page 371.
4. Connect the probes to the DUT. Figure 12-29 shows an exemplary configuration. Adjust your settings based on your connections to the DUT.

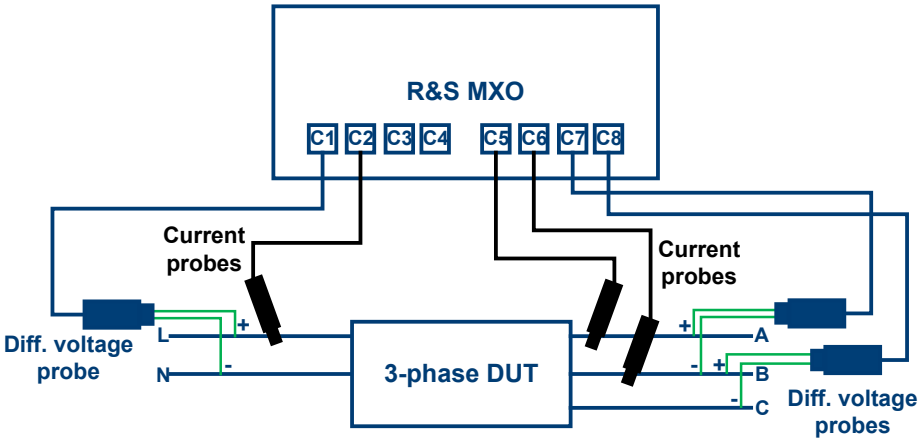



*Figure 12-29: Example for a 1V-1A input to a 2V-2A output wiring with selected lines VAB VBC IA IB*

5. In the "Analysis" tab, select the "P3 efficiency" measurement.

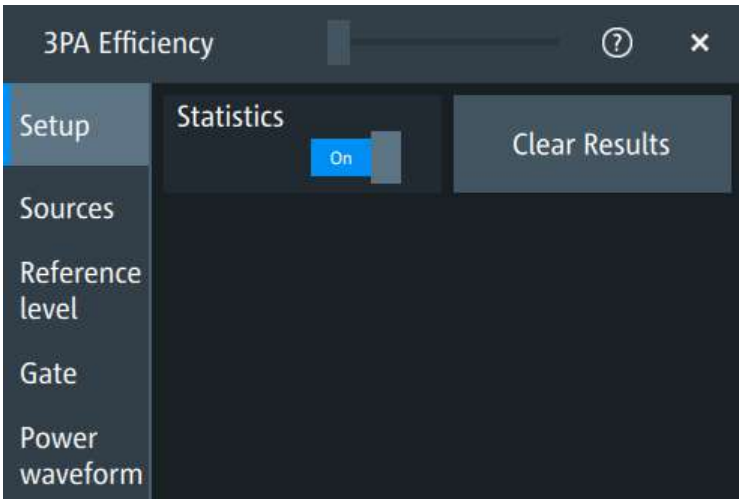
   The "P3 efficiency" setup dialog opens.
6. If necessary, enable "Statistics".
7. Select the "Sources" tab.
8. Select the "Type" of the current flow for input and output: AC or DC.
9. Select the "Wiring" and "Select lines" to configure how your probes are connected to the three-phase device.
10. Select "Config" for input and output.
11. Assign the correct channels for the "Voltage" and the "Current" probes.
12. Select the required "Bandwidth" for signal acquisition. Select "Apply to Sources" to apply the configured bandwidth setting to all sources.
13. Select "Back".
14. Select the "Power waveform" tab.

15. Adjust the scale and offset of the resulting waveforms if necessary.

    On the screen, you can see the measurement waveforms of the current, the voltage and the power. Also, the result table with numeric measurement results is shown. For details, see Section 12.4.5.7, "3-phase power efficiency results", on page 443.

### 12.4.5.2 3-phase power efficiency setup

Access: "Menu" > "Apps" > "3-phase" > "+ Add" > "P3 quality" > select "P3 efficiency" > "Setup".




**Statistics**
Activates or deactivates the statistical evaluation for the selected 3-phase measurement. If statistics are enabled, the following results are calculated:

| Label | Description |
|---|---|
| Current | Current value |
| Max | Maximum value |
| Min | Minimum value |
| Mean | Average |
| RMS | Root mean square |
| σ (S-dev) | Standard deviation |
| Event count | Number of measured events |
| Wave count | Number of waveforms (acquisitions) |

Remote command:
`P3PHase<m>:EFFiciency:STATistic:ENABle` on page 1337
`P3PHase<m>:EFFiciency:STATistic:WFMCount?` on page 1337

**Clear results**
Resets the statistical results for the respective 3-phase measurement, and starts a new statistical evaluation if the acquisition is running.

Remote command:
`P3PHase<m>:EFFiciency:STATistic:RESet` on page 1337

#### 12.4.5.3 3-phase power sources

See Section 12.4.2.1, "3-phase power sources", on page 419.

Separate source settings are available for input and output.

#### 12.4.5.4 3-phase power reference level

See Section 12.3.8.1, "Reference level", on page 410.

Separate reference level settings are available for input and output. The reference levels take effect only for the AC input and output types.

#### 12.4.5.5 3-phase power gate

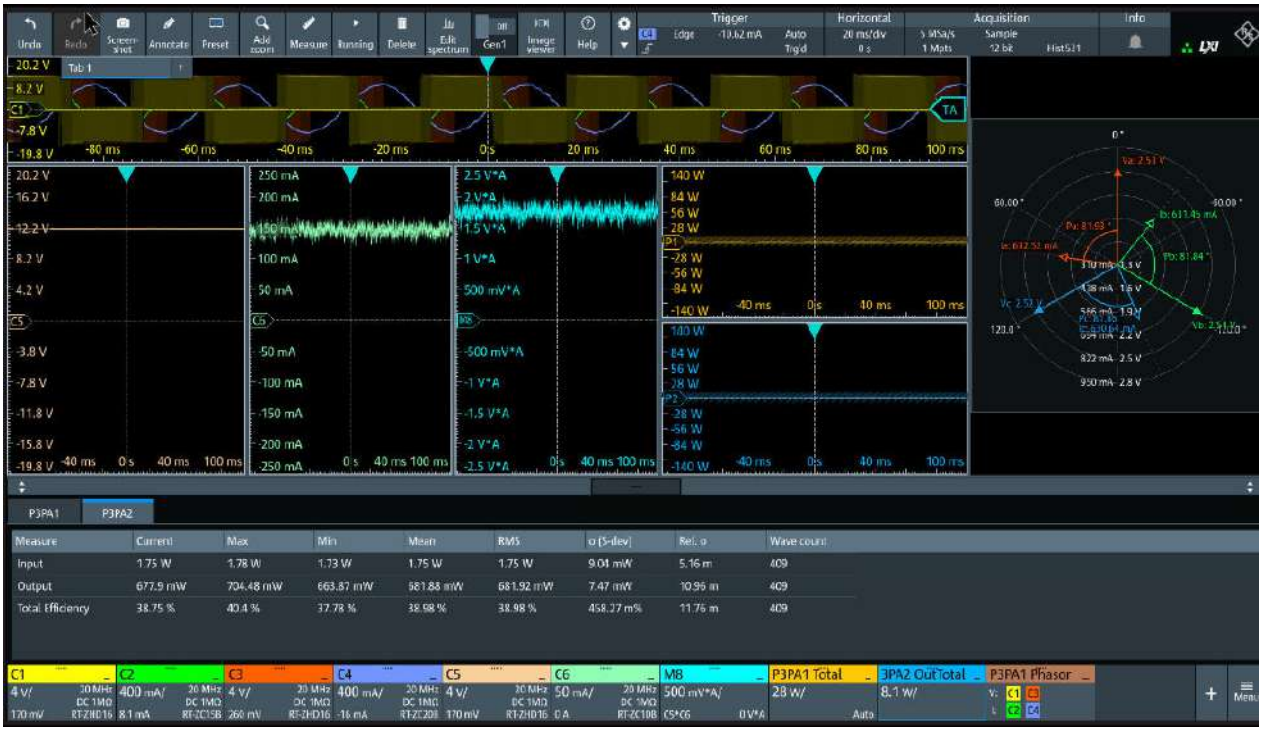
See Section 12.3.8.2, "Gate settings for power measurements", on page 414.

#### 12.4.5.6 3-phase power waveform

For a description of the power waveform scaling, see Section 12.4.2.2, "3-phase power waveform", on page 424.

Separate power waveform settings are available for input and output.

#### 12.4.5.7 3-phase power efficiency results




The results of power efficiency measurements are provided as follows:

- The waveform diagram shows the graphical presentation for the input and output of:
  - The voltage and current waveforms of the individual and total phases.
  - The power waveform of the total phase. The power waveform is the product of the current and voltage waveforms.
    See also Section 12.4.2.2, "3-phase power waveform", on page 424.
- The result table displays the numeric measurement results of input power, output power and total efficiency. The total efficiency is the ratio of input power to output power.

Remote commands: "3-phase power efficiency results" on page 1337.

## 12.5 Digital voltmeter

The software-based voltmeter features AC RMS, DC and DC RMS voltage measurements on a specific analog channel.

### 12.5.1 Using the digital voltmeter

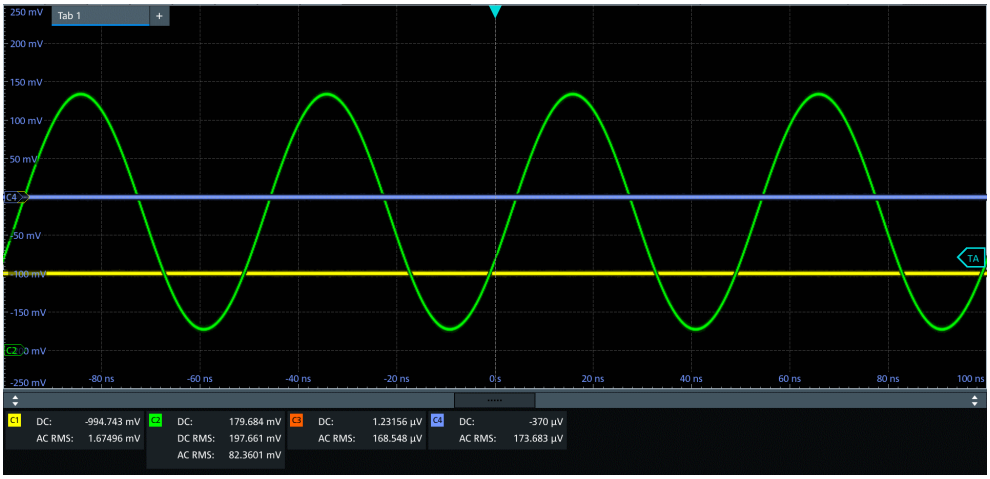
**To start a digital voltmeter measurement**

1. Tap the "Menu" > "Apps" > "General" tab > "DVM", to open the digital voltmeter measurement dialog.




2. Select the channel for your measurement.
3. Select a "Filter bandwidth" from the list.
4. Set a "Measurement time".
5. Tap on a measurement to enable it. Available are "DC", "DC RMS", "AC RMS".

   The measurement results of all enabled measurements are displayed in the result table.



### 12.5.2 Settings of the digital voltmeter

Access: "Menu" > "Apps" > "General" tab > "DVM".




**C<n>**
Selects the channel which is measured by the digital voltmeter.

**Filter bandwidth**
Selects the filter bandwidth.

Remote command:
`METer:BANDwidth` on page 1357

**Measurement time**
Sets the measurement time. The time begins in the moment that a measurement is enabled.

Remote command:
`METer:MTIMe` on page 1359

**Measurements**
Enables the voltmeter measurements. When a measurement is enabled, the voltmeter is turned on automatically.

"DC" Enables the DC voltage measurement.

"DC RMS" Enables the DC RMS voltage measurement.

"AC RMS" Enables the AC RMS voltage measurement.

Remote command:
`METer:DVMeter<m>:ACRMs:ENABle` on page 1358
`METer:DVMeter<m>:ACRMs:RESult?` on page 1358
`METer:DVMeter<m>:DC:ENABle` on page 1358
`METer:DVMeter<m>:DC:RESult?` on page 1358

METer:DVMeter<m>:DCRMs:ENABle on page 1359
METer:DVMeter<m>:DCRMs:RESult? on page 1359

**All on, All off**
Enables/disables all voltmeter measurements.

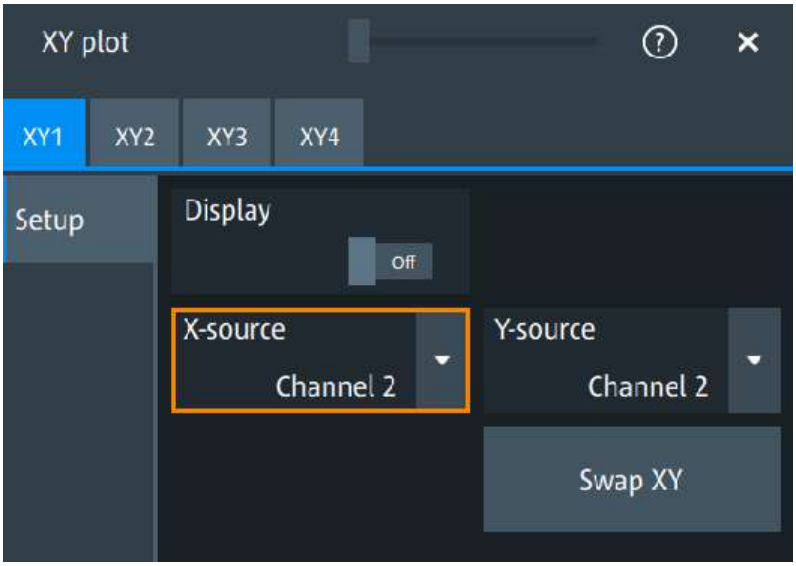
## 12.6 XY-plot

XY-plots combine the voltage levels of two waveforms in one diagram. They use the voltage level of a second waveform as the x-axis, rather than a time base. This allows you to perform phase shift measurements, for example. You can display up to four different XY-plots.

You can analyse XY-plots, for example, with cursor and automatic measurements, zone trigger and mask test. Available analysis methods depend on the selected sources of the XY-plot.

### 12.6.1 Settings for XY-plots

Access: "Menu" > "Apps" > "General" tab > "XY"




You can display up to four different XY-plots that use the voltage level of a waveform as the x-axis, rather than a time base.

Make sure to select the tab of the required XY-plot.

**Display**
If activated, the XY-waveform is active and shown in a diagram, or it is minimized in a signal icon.

Remote command:
XY<m>[:STATe] on page 1360

**X-source, Y-source**
Define the signal sources that supply the x-values and y-values of the XY-diagram.

Select from the active waveforms:

- Analog channel waveform
- Reference waveform
- Math waveform
- Track waveform (mask test and zone trigger on the resulting XY-plot is not possible)

Remote command:
XY<m>:XSOurce on page 1361
XY<m>:YSOurce on page 1360

**Swap XY**
Replaces the source of the x-axis with the source of the y-axis and vice versa.

Remote command:
XY<m>:SWAP on page 1360

## 12.6.2 Displaying an XY-diagram

**To set up an XY-plot**

1. Select "Menu" > "Apps".
2. In the > "General" tab, tap "XY".
3. In the "X-source" field, define the signal source that supplies the x-values of the XY-diagram.
4. In the "Y-source" field, define the signal source that supplies the y values of the XY-diagram.
5. Enable the "Display".
6. To switch the x- and y-values quickly, tap the "Swap XY" button.

# 13 Data and file management

This section describes how to manage general and measurement settings, waveform data, measurement results, and screenshots.

## 13.1 Accessing, retrieving and handling data

Depending on the origin and content, this description distinguishes between the following data types:

- Operational data
- Non-operational data

**Operational data (data for intended use)**

The MXO 3 uses and creates this data because of its intended use and according to the settings and configuration you have made. Thus, this data makes up most of the data that the MXO 3 creates, such as user files that contain user-defined configurations or data, like instrument settings or measurement data.

You can access, retrieve and delete user files, see Section 13.2, "Save/recall dialog", on page 449.

**Non-operational data (usage data)**

The MXO 3 creates this data during and through its use. Such data is collected e.g. for troubleshooting, to help our customer support center find solutions quickly.

The MXO 3 does not generate this data continuously and in real-time; this data is saved on the product. You can store the data on a remote server, for example via a network connection.

The MXO 3 generates the following data:

- User and authentication data
- Security and access control
- Network and communication data
- Software and firmware update data
- Diagnostic and troubleshooting data
- Environmental conditions
- Hardware component metadata.

Only when you create a service report, this data is stored in a ZIP file, see Section 19.4, "Information for technical support", on page 2060. The ZIP file contains files in TXT, XML, LOG, and CSV format, as well as binary database formats for the diagnostic and troubleshooting data. The volume of the non-operational data depends on the usage of the product and typically amounts to about 20 MB.

For additional information on the different memory types of your product and on how to protect sensitive data, see Section 19.5, "Data security", on page 2061.

## 13.2 Save/recall dialog

The "Save/recall" dialog provides functions for saving and restoring data.

Access to save settings and data: "Menu" > "Save/recall" > "Save" tab.

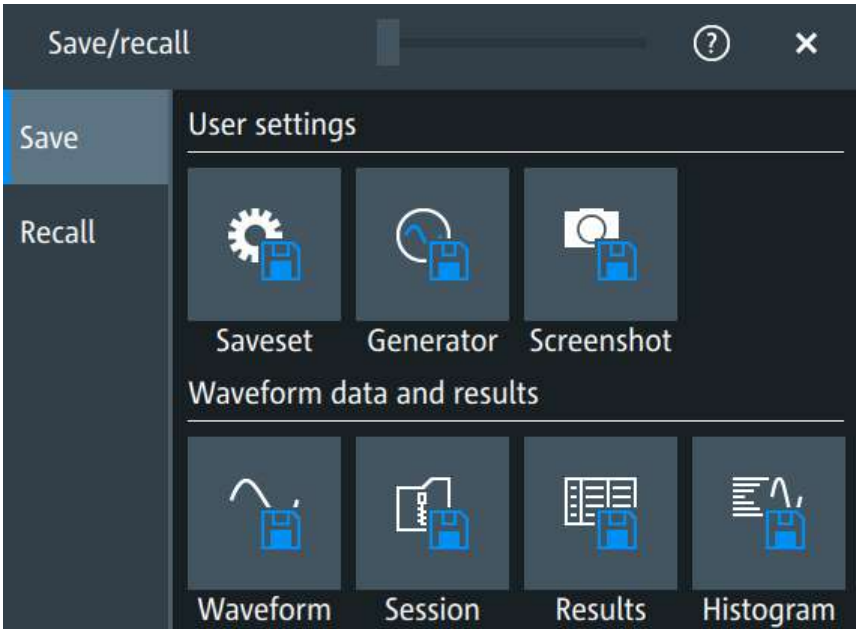



***Figure 13-1: Save tab***

Access to load settings and data: "Menu" > "Save/recall" > "Recall" tab.

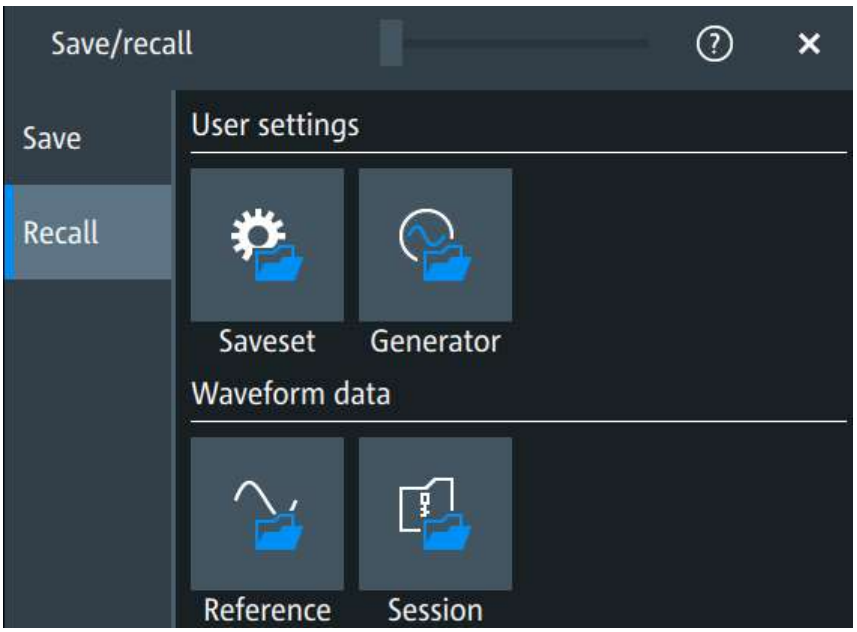



*Figure 13-2: Recall tab*

Using the "Save/recall" dialog, you can store or load various data to files for further usage, analysis and reporting:

- General settings: Section 13.3, "General and measurement settings: savesets", on page 451
- Waveforms: Section 13.4, "Waveform data", on page 455
- Reference waveforms: Section 13.4, "Waveform data", on page 455 and Section 9.4, "Reference waveforms", on page 245
- Histograms: Section 13.5, "Histogram data export", on page 471
- Measurement results: Section 13.6, "Results", on page 475
- Sessions: Section 13.7, "Sessions", on page 478
- Screenshots: Section 13.8, "Screenshots", on page 481
- Decode results of serial buses: Section 14.1.8, "Export", on page 509

The [Camera] key can be configured to save or set up screenshots.

### Storage locations

You can store data locally, or to a USB flash drive. For local storage, the path is always `/home/storage/userData`. The path has subdirectories to organize the data. For storage on USB flash drive, the default path is `/run/media/usb/<MyDriveName>/`. On both storage media, you can create directories to organize the data. You can also copy, paste and delete data.

On USB flash drives, the following file systems are supported: FAT32, FAT16, exFAT and NTFS.

The operating system differentiates between relative and absolute paths.
A relative path starts with your current directory and does not start with a `/`, e.g. `userData`.
An absolute path defines the location from the root directory, e.g. `/home/storage/userData`.

## 13.3 General and measurement settings: savesets

To repeat measurements at different times or perform similar measurements with different test data, you can save the used settings and load them again later. Furthermore, you can refer to the settings of a particular measurement when analyzing the results. Optionally, the current toolbar and dialog configuration can be included into the saveset.

Access: "Menu" > "Save/recall" > "Save" tab > "Saveset".

If you often save and load settings, try the following shortcuts:

- The "Save setup" toolbar icon saves the current settings to a file according to the settings in "Save/recall" > "Save".
- The "Recall" toolbar icon opens the dialog to select and load a setup.
- You can assign a saveset to the [Preset] key and the "Preset" function on the toolbar.
  See Section 5.6, "Preset setup", on page 109.

### 13.3.1 Using savesets

Savesets contain the complete general and measurement configuration. You can save an unlimited number of setting files.

**To save settings to a saveset file**

1. Open "Menu" > "Save/recall".
2. In the "Save" tab, tap the "Saveset" button.
   The save "Saveset" dialog opens.

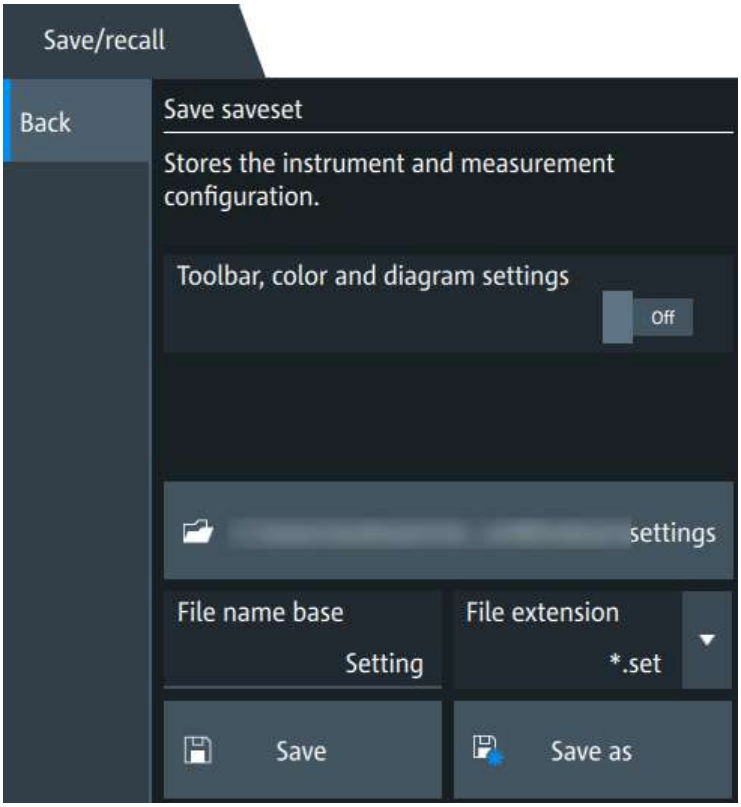



*Figure 13-3: Save saveset dialog*

3. If needed, enable "Toolbar, color and diagram settings".
4. Check the path on the "<Folder>" button. If the path does not fit, tap the button. Navigate to the folder, where you want to save the settings file.
5. Tap "Save to file".
   Alternatively, select "Save As" and select the destination folder and file name in the file selection dialog.

   The current settings are saved to the selected file.

**To load settings from a saveset file**

1. Open "Menu" > "Save/recall".
2. In the "Recall" tab, press the "Saveset" button.

   The recall "Saveset" dialog opens.

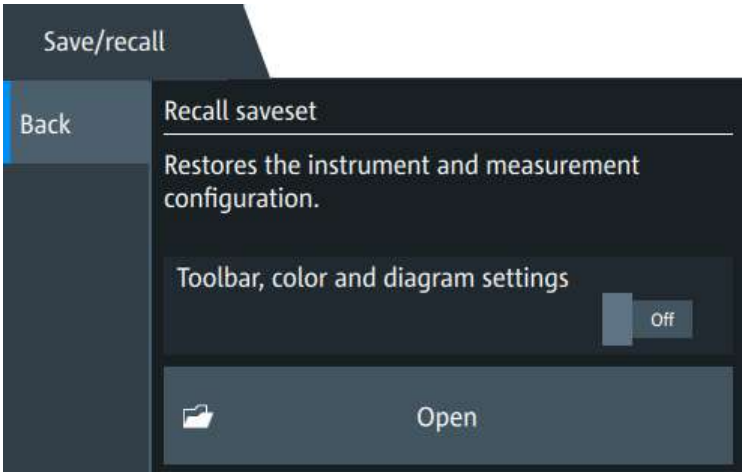



*Figure 13-4: Recall saveset dialog*

3. Tap "Open".
4. Navigate to the required saveset.

   The saved settings are loaded to the MXO 3.

Remote commands:

- `SAVeset:CONFig:UPReferences` on page 1126
- `MMEMory:SAV` on page 1124
- `MMEMory:RCL` on page 1125

### 13.3.2 Waveform generator saveset

Generator savesets store the instruments generator configuration.

**To save settings to a saveset file**

1. Open "Menu" > "Save/recall".
2. In the "Save" tab, press the "Generator" button.
   The save "Gen saveset" dialog opens.

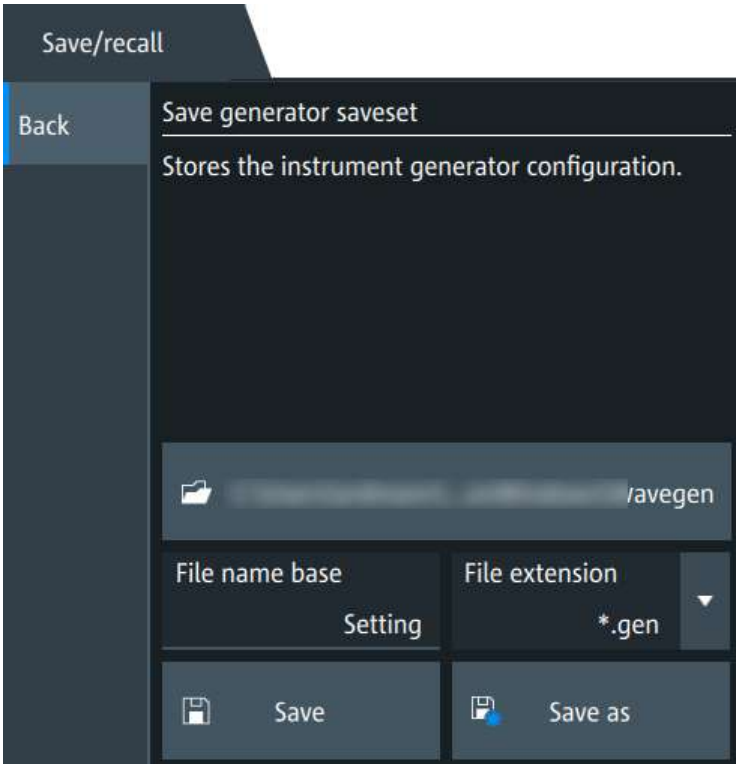



*Figure 13-5: Save generator saveset dialog*

3. Tap "Browse"

   The current settings are saved to the selected file.

**To load settings from a saveset file**

1. Open "Menu" > "Save/recall".
2. In the "Recall" tab, press the "Generator" button.

   The recall "Gen saveset" dialog opens.

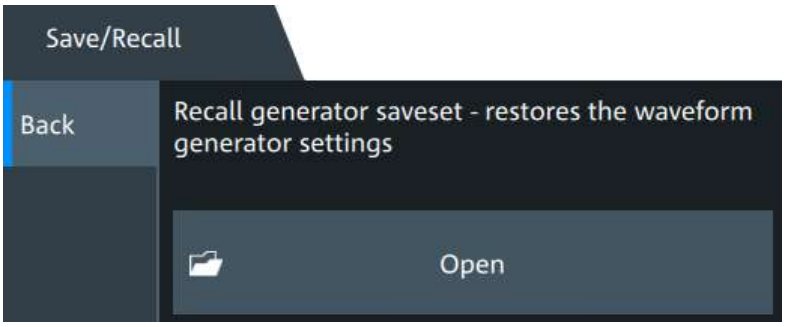



*Figure 13-6: Recall generator saveset dialog*

3. Tap "Open".

4. Navigate to the saveset.

   The saved settings are loaded to the MXO 3.

Remote commands:

- `GENerator:SAV` on page 1126
- `GENerator:RCL` on page 1126

## 13.4 Waveform data

You can save waveform data to files, and reload the data of REF files as reference waveforms.

### 13.4.1 Save waveforms: settings

Access: "Menu" > "Save/recall" > "Save" tab > "Waveform".

In this dialog, you define the storage settings for waveform data.

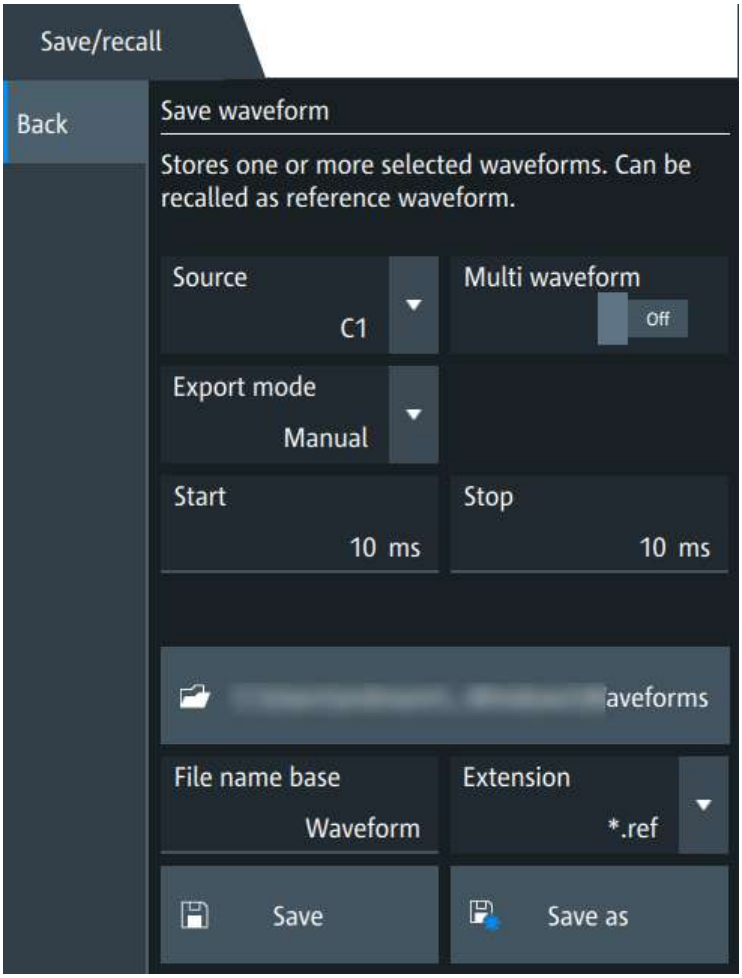



*Figure 13-7: Save a single waveform*

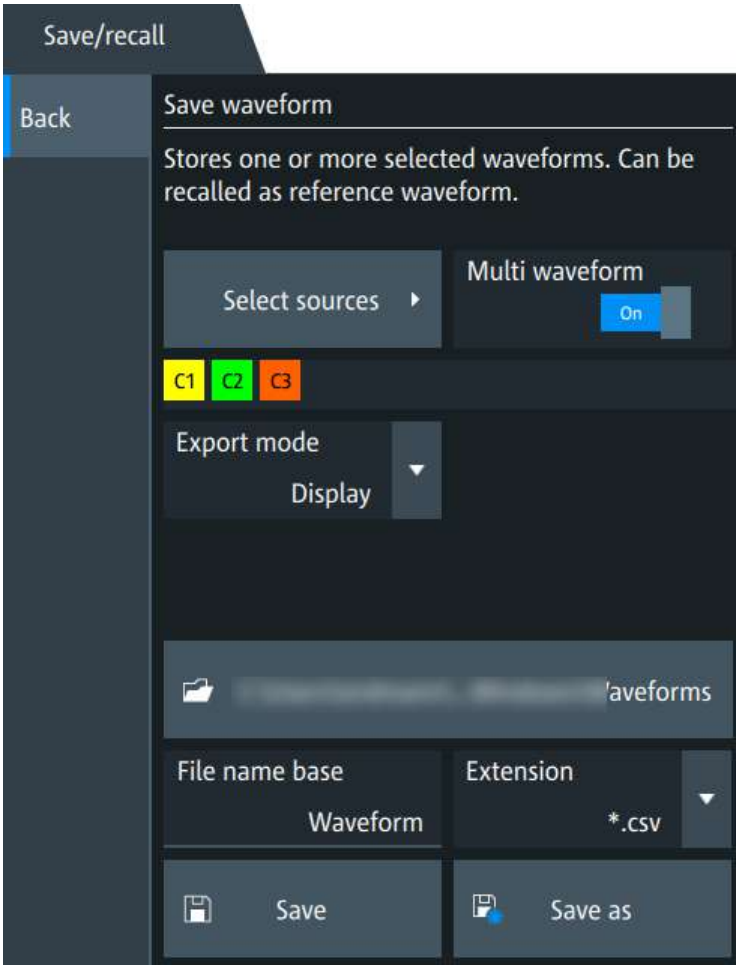



*Figure 13-8: Save multiple waveforms*

**Source**
Selects the waveform to be exported with single waveform export. The list shows all active waveforms that are available for export.

Analog and digital channels, math and reference waveforms, tracks and spectrum traces can be exported.

Remote command:
`EXPort:WAVeform:SOURce` on page 1130

**Multi waveform**
If set to "On", you can export several waveforms at once. Use Select sources to define the waveforms for data export. The following export formats are supported:

- REF files: One REF file per waveform is created. All REF files are combined in a ZIP file.
- HDF5 files: All data are written in one `.h5` file.
- CSV files: Multiple analog channels can be saved in one CSV file. Other waveform types are not supported.

For details, see Section 13.4.3, "Waveform export files", on page 462.

**Select sources**
The selection dialog provides all active waveforms. You can select or deselect all available waveforms at once with "All on" or "All off" on top of the dialog. You can also select or deselect all waveforms of a waveform group, using the button to the right of the group.

Analog and digital channels, math and reference waveforms, tracks and spectrum traces can be exported. Note that in CSV files, only multiple analog channels can be saved.




Remote command:
`EXPort:WAVeform:SOURce` on page 1130

**Export mode**
Defines the part of the waveform record that has to be stored.

| | |
|---|---|
| "Display" | Saves the waveform data that is displayed in the diagram. |
| "All data" | Saves the complete waveform record. |
| "Cursor" | Saves the data between the cursor lines if at least one cursor measurement is defined for the source waveform. If several cursor sets are defined, select the "Cursor set" to be used for export. |
| "Gate" | Saves the data included in the measurement gate if a gated measurement is defined for the source waveform. Select the "Gate" to be used for export. |
| "Manual" | Saves the data between user-defined "Start" and "Stop" values. |

Remote command:
`EXPort:WAVeform:SCOPe` on page 1129
`EXPort:WAVeform:CURSorset` on page 1128

EXPort:WAVeform:GATE on page 1128
EXPort:WAVeform:STARt on page 1131
EXPort:WAVeform:STOP on page 1132

**Save settings**
You define the details of the filename, extension and directory, and save the file.

A progress bar informs you about the process.

**<Folder> ← Save settings**
Opens a file explorer where you can select the directory where the file is saved.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation. It has **no** effect on EXPort:WAVeform:SAVE.

Remote command:
EXPort:WAVeform:AUTonaming:PATH on page 1127

**File name base ← Save settings**
Sets a name for the file, without extension. The name is extended with a time stamp when the file is saved. Thus, multiple "Save" actions are possible without changing the filename.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation. It has **no** effect on EXPort:WAVeform:SAVE.

Remote command:
EXPort:WAVeform:AUTonaming:NAME on page 1127

**File extension ← Save settings**
Selects the format of the exported file.

You can select between "*.csv" (Excel format), "*.ref" (Reference Waveform Format) and "*.h5" (HDF5 format). REF files can be reloaded as reference waveforms. See: Section 13.4.3, "Waveform export files", on page 462. If you have selected multiple waveforms other than analog channels, CSV is not available.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation. It has **no** effect on EXPort:WAVeform:SAVE.

Remote command:
EXPort:WAVeform:AUTonaming:TYPE on page 1128

**Save ← Save settings**
Saves the file in the defined folder using the "File name base". If the specified file already exists, it is overwritten with the new data.

**Save As ← Save settings**
Opens a file explorer where you select the folder, the file type, enter the filename and save the file.

Remote command:
EXPort:WAVeform:NAME on page 1128
EXPort:WAVeform:SAVE on page 1129

### 13.4.2 Saving waveforms

**To save a single waveform**

1. Open "Menu" > "Save/recall".
2. On the "Save" tab, tap "Waveform".

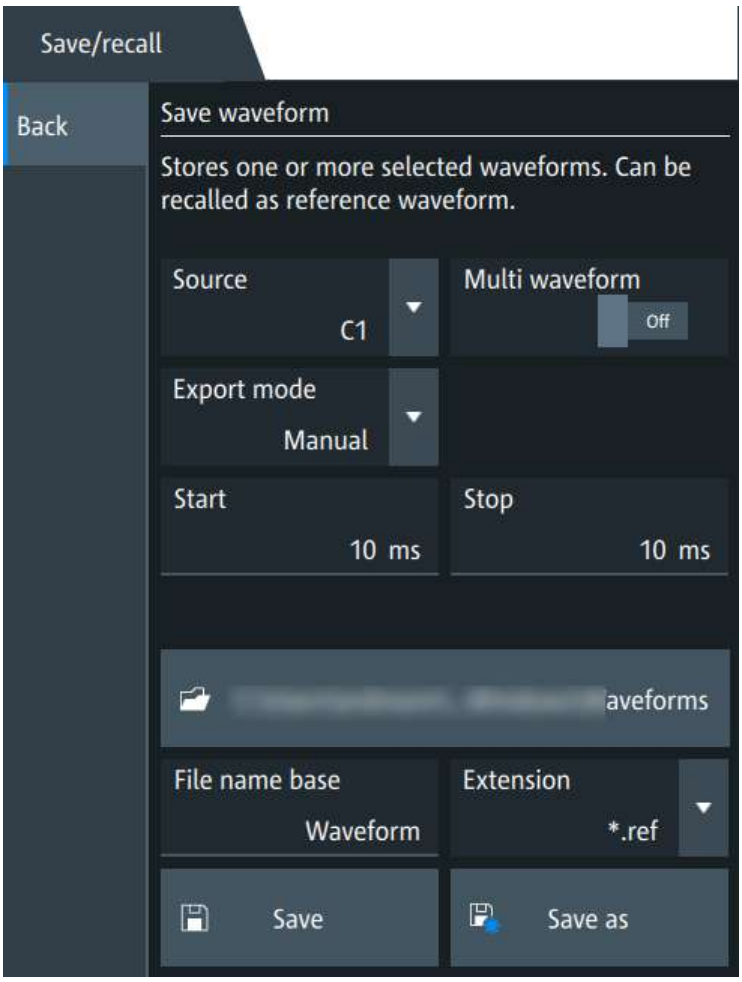



*Figure 13-9: Save waveform dialog*

3. Select the "Source".
4. Select the "Export mode" to define the exported part of the waveform.
5. If necessary, define the cursor, gate or start/stop values for the data.
6. Select the format of the export file: "File extension".
7. Set the "File name base".
8. Tap "Save" or "Save As".

   The selected waveform is saved to the specified file. A progress bar informs you about the saving process.

**To save multiple waveforms**

1. Open "Menu" > "Save/recall".
2. In the "Save" tab, tap "Waveform".
3. Enable "Multi waveform"

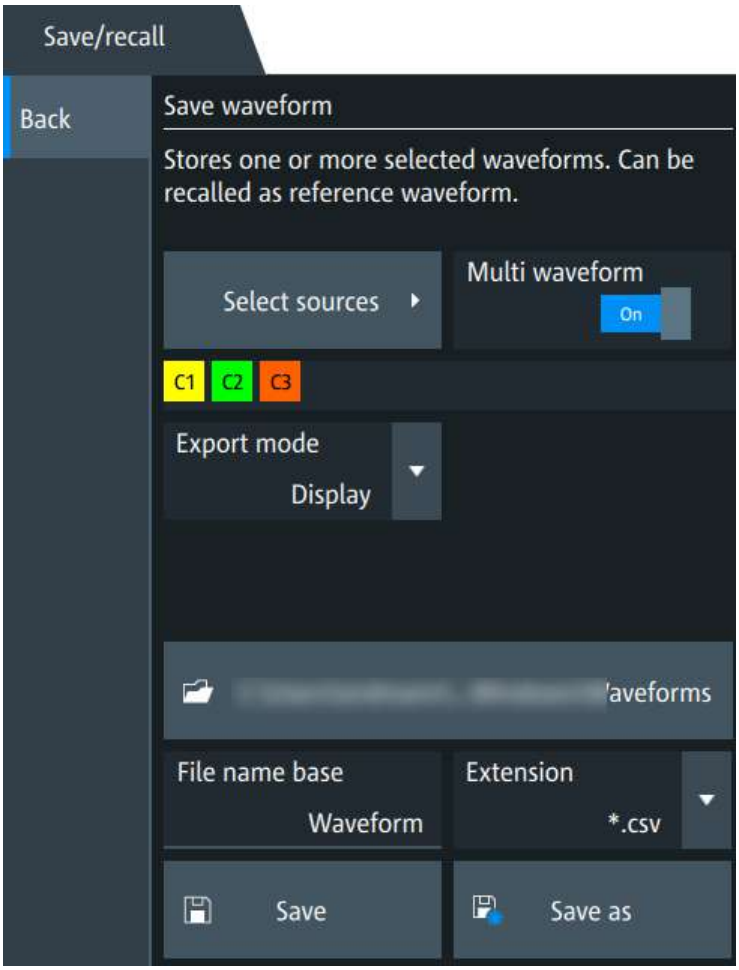
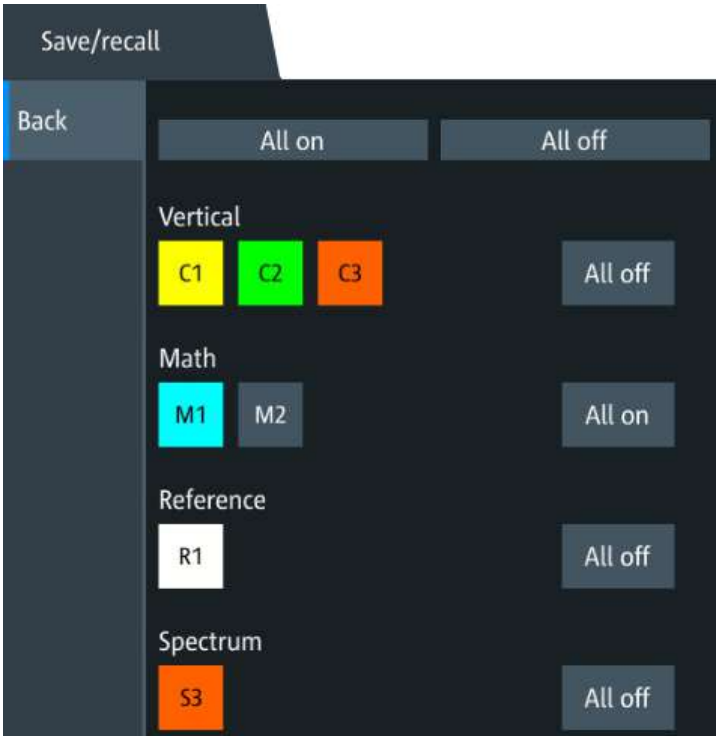



*Figure 13-10: Save multiple waveforms dialog*

4. Tap "Select sources".
5. Select the waveforms for export.




6. Tap "Back".
7. Select the "Export mode" to define the exported part of the waveform.
8. If necessary, define the cursor, gate or start/stop values for the data.
9. Select the format of the export file: "File extension".
10. Set the "File name base".
11. Tap "Save" or "Save As".

    The selected waveforms are saved to the specified file. A progress bar informs you about the saving process.

## 13.4.3 Waveform export files

You can store the waveform data in several formats:

- CSV (comma-separated values): ASCII-based format, easy access with Excel or text tools. See Section 13.4.3.1, "CSV files", on page 462.
- REF (reference waveform): binary data with XML header, can be reloaded as reference waveforms. See Section 13.4.3.2, "REF files", on page 466.
- HDF5 (`.h5`): binary container format for external processing of waveform data.

### 13.4.3.1 CSV files

A CSV file is a comma-separated values (CSV) text file, the waveform is stored in a table. The values are separated by commas.

You can convert the comma-separated text to columns. If you convert the data, take care of the correct number conversion, depending on the tool in use.

### Content of waveform files

Header and data are written into one file. The first lines of the file contain header data, for example, time scale, vertical scale, vertical and horizontal positions. Header data is required to interpret the waveform data, and to analyze the data values of the data file.

Below the header, the waveform data follows. For each sample, one line is written. The first value is the X-value (time or frequency), the next is the Y-value in the unit indicated in the header data. For envelope or peak detect waveforms, two Y-values (minimum and maximum) are written for each sample.

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Model,MXO54 | | | | | |
| 2 | SerialNumber,101416 | | | | | |
| 3 | , | | | | | |
| 4 | SourceType,CHANNEL_TRACE | | | | | |
| 5 | SignalSource,Ch1Wfm1 | | | | | |
| 6 | Resolution,2e-10 | | | | | |
| 7 | RecordLength,1000 | | | | | |
| 8 | HWRecordLength,1000 | | | | | |
| 9 | HorizontalPosition,0 | | | | | |
| 10 | TraceArithmetics,SINGLE | | | | | |
| 11 | BaseUnit,LEVEL_V | | | | | |
| 12 | BaseUnitRelative,OFF | | | | | |
| 13 | Impedance,50 | | | | | |
| 14 | UserTermination,1M_OHM | | | | | |
| 15 | XStart,-1.0000000000000001e-07 | | | | | |
| 16 | XStop,1.0000000000000001e-07 | | | | | |
| 17 | TimebaseScale,2e-08 | | | | | |

| | |
|---|---|
| 37 | TIME,C1 |
| 38 | -1e-07,0.0031761544099682382 |
| 39 | -9.98e-08,0.002565355484974346 |
| 40 | -9.9599999999999999e-08,0.000977278279990227 |
| 41 | -9.939999999999999e-08,0.0008551184949914487 |
| 42 | -9.919999999999999e-08,0.00036647935499633513 |
| 43 | -9.899999999999999e-08,-0.0010994380649890054 |
| 44 | -9.879999999999998e-08,-0.0018323967749816759 |
| 45 | -9.859999999999998e-08,-0.0020767163449792324 |
| 46 | -9.839999999999998e-08,-0.002321035914976789 |
| 47 | -9.819999999999998e-08,-0.001954556559980454 |
| 48 | -9.799999999999998e-08,-0.0008551184949914487 |

***Figure 13-11: CSV file of a time waveform, opened in Excel***

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Model,MXO54 | | | | |
| 2 | SerialNumber,101416 | | | | |
| 3 | , | | | | |
| 4 | SourceType,SPECTRUM | | | | |
| 5 | SignalSource,SPECTRUM1_NORMAL | | | | |
| 6 | BaseUnit,LEVEL_V | | | | |
| 7 | BaseUnitRelative,OFF | | | | |
| 8 | Impedance,50 | | | | |
| 9 | UserTermination,1M_OHM | | | | |
| 10 | VerticalPosition,0 | | | | |
| 11 | VerticalScale,10 | | | | |
| 12 | VerticalOffset,-45 | | | | |
| 13 | BaseYStart,0 | | | | |
| 14 | BaseYStop,0 | | | | |
| 15 | ViewUnit,LEVEL_DBM | | | | |
| 16 | ViewUnitRelative,OFF | | | | |
| 17 | ViewReferenceLevel,0 | | | | |
| 18 | NumericFormat,DECIMAL | | | | |
| 19 | CenterFreq,1000000000.0000001 | | | | |
| 20 | FreqSpan,2000000000 | | | | |

| | |
|---|---|
| 33 | FREQUENCY,SPECNORM1 |
| 34 | -213623.27970552444,-49.02157013180808 |
| 35 | 549316.1734194754,-49.769556105402145 |
| 36 | 1312255.6265444753,-53.96493039074978 |
| 37 | 2075195.079669475,-61.57760265043916 |
| 38 | 2838134.532794475,-70.19729413316635 |
| 39 | 3601073.985919475,-73.24404760765583 |
| 40 | 4364013.439044475,-75.68197528968501 |
| 41 | 5126952.892169475,-91.81643862397064 |

*Figure 13-12: CSV file of a spectrum waveform, opened in Excel*

If you export multiple analog channels at once, one column per channel is written. For envelope or peak detect waveforms, two values per channel are written.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | Model | MXO54C | | | | | | |
| 2 | SerialNumber | 100864 | | | | | | |
| 3 | | | | | | | | |
| 4 | SourceType | CHANNEL_TRACE | | CHANNEL_TRACE | | CHANNEL_TRACE | | CHANNEL_TRACE |
| 5 | SignalSource | Ch1Wfm1 | | Ch2Wfm1 | | Ch3Wfm1 | | Ch4Wfm1 |
| 6 | Resolution | 2.00E-10 | | 2.00E-10 | | 2.00E-10 | | 2.00E-10 |
| 7 | RecordLength | 1000 | | 1000 | | 1000 | | 1000 |
| 8 | HWRecordLength | 1000 | | 1000 | | 1000 | | 1000 |
| 9 | HorizontalPosition | 0 | | 0 | | 0 | | 0 |
| 10 | TraceArithmetics | ENVELOPE | | ENVELOPE | | ENVELOPE | | ENVELOPE |
| 11 | BaseUnit | LEVEL_V | | LEVEL_V | | LEVEL_V | | LEVEL_V |
| 12 | BaseUnitRelative | OFF | | OFF | | OFF | | OFF |
| 13 | Impedance | 50 | | 50 | | 50 | | 50 |
| 14 | UserTermination | 1M_OHM | | 1M_OHM | | 1M_OHM | | 1M_OHM |
| 15 | XStart | -1.00E-07 | | -1.00E-07 | | -1.00E-07 | | -1.00E-07 |
| 16 | XStop | 1.00E-07 | | 1.00E-07 | | 1.00E-07 | | 1.00E-07 |

| | TIME | C1 MAX | C1 MIN | C2 MAX | C2 MIN | C3 MAX | C3 MIN | C4 MAX | C4 MIN |
|---|---|---|---|---|---|---|---|---|---|
| 37 | | | | | | | | | |
| 38 | -1.00E-07 | 0.00586367 | -0.006352309 | 0.002565355 | -0.005130711 | 0.003786953 | -0.005008551 | 0.003542634 | -0.004275592 |
| 39 | -9.98E-08 | 0.008551185 | -0.003664794 | 0.002687515 | -0.005252871 | 0.003542634 | -0.004642072 | 0.003664794 | -0.003909113 |
| 40 | -9.96E-08 | 0.010750061 | -0.000732959 | 0.003053995 | -0.005008551 | 0.003176154 | -0.00561935 | 0.003420474 | -0.004275592 |
| 41 | -9.94E-08 | 0.013804056 | 0.001588077 | 0.002931835 | -0.005252871 | 0.003298314 | -0.005375031 | 0.003786953 | -0.004031273 |
| 42 | -9.92E-08 | 0.016247251 | 0.004519912 | 0.003053995 | -0.005130711 | 0.003053995 | -0.005008551 | 0.003786953 | -0.004764232 |
| 43 | -9.90E-08 | 0.019179086 | 0.007451747 | 0.002565355 | -0.005130711 | 0.003420474 | -0.004764232 | 0.003909113 | -0.004397752 |
| 44 | -9.88E-08 | 0.021133643 | 0.010017102 | 0.002687515 | -0.005130711 | 0.002931835 | -0.004764232 | 0.004519912 | -0.004397752 |
| 45 | -9.86E-08 | 0.023576839 | 0.012582458 | 0.003420474 | -0.004886391 | 0.003664794 | -0.004397752 | 0.004275592 | -0.004519912 |
| 46 | -9.84E-08 | 0.025897874 | 0.015392133 | 0.003053995 | -0.005130711 | 0.003664794 | -0.004397752 | 0.004153433 | -0.004031273 |

*Figure 13-13: CSV file of four analog channels, envelope acquisition mode, opened in Excel and converted to columns*

## Header data

The header lines contain the following properties, depending on the type of the exported waveform (time or frequency):

*Table 13-1: Header file properties in CSV files*

| Value | Description | Time | Frequency |
|---|---|---|---|
| Model | Model of the instrument | X | X |
| SerialNumber | Serial number of the instrument | X | X |
| SourceType | Type of the exported waveform | X | X |
| SignalSource | Source of the exported waveform | X | X |
| Resolution | Time between two samples<br>*Resolution = 1 / Sample Rate* | X | |
| RecordLength | Number of samples in a waveform record of one acquisition | X | |
| HWRecordLength | Equivalent to the RecordLength | X | |
| HorizontalPosition | Horizontal position of the waveform in divisions | X | |
| TraceArithmetics | Arithmetic of the waveform: off, envelope, or average | X | |
| BaseUnit | Base unit of a mathematic waveform, for example, linear unit | X | X |
| BaseUnitRelative | Base unit, if a relative unit (e.g. dB) is enabled | X | X |
| Impedance | Input impedance, used for power calculation | X | X |
| UserTermination | User-defined load impedance connected to the probe | X | X |
| XStart | Horizontal start value of the waveform (time or frequency), as defined in the export settings | X | |
| XStop | Horizontal stop value of the waveform (time or frequency), as defined in the export settings | X | |
| TimebaseScale | Horizontal scale in seconds per division | X | |
| ReferencePoint | Position of the zero point in % of the screen | X | |
| VerticalPosition | Vertical position of the waveform in divisions | X | X |
| VerticalScale | Vertical scale of the waveform | X | X |
| VerticalOffset | Vertical offset of the waveform in Volts, or other unit | X | X |
| BaseYStart | Vertical start value of the waveform | X | X |
| BaseYStop | Vertical stop value of the waveform | X | X |
| ViewUnit | User-selected unit of a waveform, for example, logarithmic unit for a spectrum. | X | X |
| ViewUnitRelative | Indication of a relative unit. It is true if the ViewUnit implies a reference level, for example dB. | X | X |
| ViewReferenceLevel | Reference level for a relative unit. The value is only valid if the unit is relative (ViewUnitRelative = ON). | X | X |
| NumericFormat | Number format of bus values and digital channel data (bit pattern format) | X | X |
| EnhancementMode | Method to increase the sample rate if the required sample rate is higher than the ADC sample rate. | X | |

| Value | Description | Time | Frequency |
|---|---|---|---|
| InterpolationMode | Interpolation method. The value is relevant when the enhancement mode is interpolated time. | X | |
| CenterFreq | Center frequency of the spectrum | | X |
| FreqSpan | Frequency span of the spectrum | | X |
| ResolutionBW | Resolution bandwidth of the spectrum | | X |
| FrequencyStart | Start frequency of the spectrum | | X |
| FrequencyStop | Stop frequency of the spectrum | | X |
| WindowType | Window used for the spectrum computation | | X |
| GateRBWCoupling | Indication whether the record length or the resolution bandwidth is a constant for the spectrum computation | | X |
| XAxisMode | Indicates a linear or logarithmic x-axis | | X |
| HorizontalDivisionCount | Number of horizontal divisions | X | X |
| VerticalDivisionCount | Number of vertical divisions | X | X |
| FirmwareVersion | Firmware version that is installed on the MXO 3 | X | X |
| DecimationMode | Method to reduce the number of data samples to achieve the required sample rate: Sample, peak detect, HiRes, RMS | X | |

#### 13.4.3.2 REF files

The REF format is a specific format, which allows you to reload the waveform data as a reference waveform. The data is saved in a zipped file that contains two files per waveform. One file is a binary file (BIN file) and contains the waveform data values. The second file is an XML file and contains the header data in several data groups. Header data is required to reload the waveform from data, or to analyze the data values of the data file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Group Name="SignalExportBase">
  <Group Name="SignalAttributes">
    <Prop Name="SourceType" Value="CHANNEL_TRACE" UserValue="NONE" />
    <Prop Name="SignalSource" Value="Ch1Wfm1" />
    <Prop Name="Resolution" Value="2.0000000000000001e-10" />
    <Prop Name="RecordLength" Value="1000" />
    <Prop Name="HWRecordLength" Value="1000" />
    <Prop Name="HorizontalPosition" Value="0" />
    <Prop Name="TraceArithmetics" Value="SINGLE" UserValue="SINGLE" />
    <Prop Name="BaseUnit" Value="LEVEL_V" UserValue="LEVEL_V" />
    <Prop Name="BaseUnitRelative" Value="OFF" UserValue="OFF" />
                    ⋮
    <Group Name="PostProcessingBufferSampleCounts">
      <Prop Name="DeltaPreSamples" Value="25" />
      <Prop Name="PreSamples" Value="500" />
      <Prop Name="PostSamples" Value="500" />
      <Prop Name="DeltaPostSamples" Value="29" />
    </Group>
  </Group>
</Group>
```

*Figure 13-14: Header file in XML format, part of the REF file container*

*Table 13-2: Header file properties in REF files*

| Value | Description |
|---|---|
| **SignalAttributes**<br>Same values as in CSV files, except for Model and SerialNumber. See Table 13-1. | |
| **SignalAttributesPostProcessing** | |
| SignalFormat | Format of the data values in the BIN file: INT8BIT, INT16BIT, INT32BIT, FLOAT, DOUBLE |
| Origin | WFM_EXPORT: export of one acquisition or export of multiple acquisitions |
| ByteOrder | Byte order of the values in the BIN file<br>● LSB first: little endian, least significant byte first<br>● MSB first: big endian, most significant byte first |
| NumberOfWaveforms | Number of waveforms. always = 1 |
| TimestampState | OFF |
| Timestamp | Absolute time of the waveform recording in ISO 8601 format |
| DecimationMode | Method to reduce the number of data samples to achieve the required sample rate |
| IsMinMax | If ON, then min and max values are saved for each sample. For envelope and peak detect waveforms. |
| IsComplex | If ON, then two values are saved for each sample (I and Q value). |
| TriggerOffsetToPostSampleInSeconds | Offset between the exact trigger position and the first sample after the trigger |
| SignalRecordLength | Number of samples in the BIN file. If the signal is a spectrum, the value indicates the number of FFT bins.<br>The value can differ from RecordLength and HWRecordLength because the data file has additional samples before and after the record to consider interpolation and other waveform parameters and their dependencies. |

| Value | Description |
|---|---|
| SignalXStart | Time of the first sample in the acquired data. Different from XStart because the data file has additional samples before and after the record, and if only part of the data is exported. |
| SignalXStop | Time of the last sample in the data acquired. Different from XStop because the data file has additional samples before and after the record, and if only part of the data is exported. |
| AdjustedResolutionBW | Actual resolution bandwidth of a spectrum waveform. Only valid if the exported waveform is a spectrum. |
| NoiseBandwidth | Noise bandwidth of a spectrum waveform, required for power calculation. Only valid if the exported waveform is a spectrum. |
| ValuesPerSample | Number of y-values saved at each sampling time. The value is usually 1. The value is 2, if min and max values are saved for each sample, for example, for envelope waveforms. |
| NofQuantisationLevels | Number of quantization levels of y values |
| TOADone | If ON, then TriggerOffsetToPostSampleInSeconds is compensated. |
| SignalResolution | Time between two samples in this waveform. The value can differ from Resolution if the source is, for example, a spectrum. The value is determined automatically considering the waveform parameters and their dependencies. If the signal is a spectrum, the value indicates the frequency range of FFT bins. |
| HorizontalDomain | Domain of the x-axis (time or frequency) |
| **PostProcessingVerticalAxisAttributes** | |
| Range | Range of the signal in y-direction |
| Offset | Center of the y-range |
| Resolution | Resolution of the y-axis |
| IsLogarithmic | Indicates if the y-axis is logarithmic (ON or OFF) |
| **PostProcessingSampleFormatAttributes** | |
| ValueStorage | Interpretation of the data: is integer or float (floating point data) |
| ValueInterpretation | Indicates if the data is saved as binary ADC values (BINARY), or physical values (PHYSICAL) |
| SampleBitSize | Value size in bit, as used in processing |
| ValueBitSize | Value size of ADC data in bit |
| LsbExtensionBits | Number of bits the ADC value is shifted to the left in the CPU sample size |
| AdcValueBitSize | Number of bits of the ADC that was used to sample the signal. It does not include any additionally gained bits due to high definition mode. |
| IsSigned | Signed values (ON) or unsigned values (OFF) |

| Value | Description |
|---|---|
| **PostProcessingBufferSampleCounts** | |
| [DeltaPreSamples \| PreSamples \|\| PostSamples \| DeltaPostSamples] | Indicates the distribution of the samples. \|\| marks the trigger time. The sum of PreSamples and PostSamples matches the requested record length. DeltaSamples are additional samples that are needed for computation but they are not measured or displayed. |

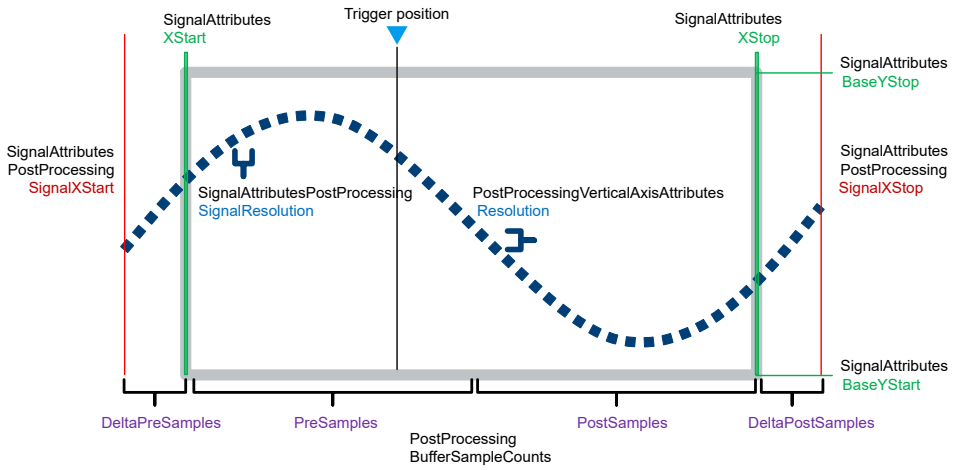



*Figure 13-15: Main header properties at the waveform*

Note: FractionalBits was removed in FW version 1.3.x. Reference waveforms created with older FW versions are compatible and can be reloaded.

### 13.4.3.3 HDF5 files

An HDF5 (`.h5`) file is a container format that contains attributes and waveform data. The format can be read by different programming languages with existing libraries. HDF5 stores binary data, thus the file size is smaller than CSV files.

Use the HDF5 format to export data from the MXO 3 and analyze the waveforms with external software.

You can save one or more waveforms in one HDF5 file.

The HDF5 file consists of several groups in a hierarchical structure. The data samples are float32 values of the physical values. No conversion is needed. For envelope and peak detect waveforms, two Y-values are written for each sample ("min" and "max").

*Table 13-3: Contents and structure of the HDF5 file*

| Group | Subgroup | Attribute, example values, data | Description |
|---|---|---|---|
| /FileType | | | Type of the HDF5 file. |
| | | Rohde&Schwarz: "MXO" | Instrument family. |
| | | Version: "1.0" | Version of the HDF5 format. |

| Group | Subgroup | Attribute, example values, data | Description |
|---|---|---|---|
| /Frame | | | Export information. |
| | | DateTime: "<ISO time stamp>" | Date and time of the file export. |
| | | FirmwareVersion: "2.5.0.60" | Version of the instrument firmware. |
| | | Model: "MXO44" | Instrument model. |
| | | SerialNumber: "100000" | Serial number of the exporting instrument. |
| /Waveforms | | | Attributes and data. |
| | /Waveforms/C1 | | One subgroup for each waveform. |
| | | Count: 1 | Number of acquisitions, always = 1. |
| | | NumPoints: 1000 | Number of exported samples (record length of the exported waveform). |
| | | XInc: 2e-10 | Resolution, the time between two samples. |
| | | XOrg: -1.0e-7 | Horizontal start value of the waveform, time of the first sample. |
| | | XUnit: "Time" | Unit of the horizontal axis (Time, Frequency). |
| | | YUnit: "V" | Unit of the vertical axis. |
| | | /Waveforms/C1/C1 Additional Info | Subgroup with further information.<br>See "Additional info (/Waveforms/C1/C1 Additional Info)" on page 470. |
| | | DataSet(/Waveforms/C1/C1 Data) | Data values. |
| | | DateTime: "<ISO time stamp>" | Date and time of the trigger of this acquisition. |
| | | TriggerOffsetToPostSampleInSeconds: 1.1e-12 | Offset between the trigger time and the first sample after the trigger, in seconds. |
| | | DataSet(/Waveforms/C1/C1 PreData) | Additional samples before the samples in DataSet, are used to settle the filters. |
| | | DataSet(/Waveforms/C1/C1 PostData) | Additional samples after the samples in DataSet, are used to settle the filters. |
| | /Waveforms/C2 | Analogous to /Waveforms/C1 | |
| | /Waveforms/C3 | Analogous to /Waveforms/C1 | |

### Additional info (/Waveforms/C1/C1 Additional Info)

The `/Waveforms/C1/C1 Additional Info` group provides additional information on the waveform, which is not essential to interpret the waveform data. Most of this information is also available in the header of CSV or REF files.

| Field | Description | Identical to CSV | Identical to REF |
|---|---|---|---|
| DecimationMode | Method to reduce the number of data samples to achieve the required sample rate. | X | |
| EnhancementMode | Method to increase the sample rate if the required sample rate is higher than the ADC sample rate. | X | |
| HorizontalPosition | Horizontal position of the waveform in divisions. | X | |
| Impedance | Input impedance, used for power calculation. | X | |
| InterpolationMode | Interpolation method. The value is relevant when the enhancement mode is interpolated time. | X | |
| Label | Channel label, currently empty. | | |
| NofQuantisationLevels | Number of quantization levels of y values. | | X |
| ReferencePoint | Position of the zero point in % of the screen. | X | |
| SignalSource | Source of the exported waveform. | X | |
| SourceType | Type of the exported waveform. | | X |
| TOADone | If ON, then TriggerOffsetToPostSampleInSeconds is compensated. | X | |
| TimebaseScale | Horizontal scale in seconds per division. | X | |
| TraceArithmetics | Arithmetic of the waveform: off, envelope, or average. | X | |
| UserTermination | User-defined load impedance connected to the probe. | X | |
| VerticalOffset | Vertical offset of the waveform in Volts, or other unit. | X | |
| VerticalPosition | Vertical position of the waveform in divisions. | X | |
| VerticalScale | Vertical scale of the waveform. | X | |
| XDispOrigin | X value on the left border of the screen. | | |
| XDispRange | X value range that is shown on the screen. | | |
| YDispOrigin | Y value of the bottom border of the screen. | | |
| YDispRange | Y value range that is shown on the screen. | | |
| YInc | Value of the quantization level of the Y values. | | |

## 13.5 Histogram data export

You can export the data of waveform histograms and histograms on measurement results to file.

### 13.5.1 Files with waveform histograms

The data of one histogram is saved to one CSV file. The file content starts with header data followed by normalized or absolute histogram values.

The header has the following information:

- Name of the histogram
- Source waveform of the histogram
- Histogram mode: vertical or horizontal
- Normalization on or off
  - On: Data values are float values that represent the ratio of samples in a given bin to the sum of all samples. The sum of all normalized values is 1.
  - Off: Data values are integer values that represent the number of samples in a given bin.
- Histogram range: start and end values.
  - Vertical histograms: the values are given in the unit of the vertical axis
  - Horizontal histograms: the unit is s.

#### Example of file content

*Table 13-4: Content of histogram CSV file*

| Normalization on | Normalization off | Number of the data row for value calculation |
|---|---|---|
| HistogramName,Hist1 | HistogramName,Hist1 | |
| HistogramSource,Ch1Wfm1 | HistogramSource,Ch1Wfm1 | |
| HistogramType,VERTICAL | HistogramType,VERTICAL | |
| Normalization,OFF | Normalization,ON | |
| HistogramStart,-0.25024431956999754 | HistogramStart,-0.25024431956999754 | |
| HistogramEnd,0.25012215978499874 | HistogramEnd,0.25012215978499874 | |
| , | , | |
| , | , | |
| , | , | |
| DATA | DATA | |
| 0 | 0 | 1 |
| 0 | 0 | 2 |
| 100 | 0.1 | 3 |
| 800 | 0.8 | 4 |
| 100 | 0.1 | 5 |
| 0 | 0 | 6 |
| 0 | 0 | 7 |
| 0 | 0 | 8 |

### 13.5.2 Files with histograms on measurement results

The data of one histogram is saved to one CSV file. The file content starts with header data followed by normalized or absolute histogram values.

The header has the following information:

- Index of the measurement
- Measurement type
- Measurement source
- Normalization on or off
  - On: Data values are float values that represent the ratio of samples in a given bin to the sum of all samples. The sum of all normalized values is 1.
  - Off: Data values are integer values that represent the number of samples in a given bin.
- Histogram range: start and end values in the measurement unit

#### Example of file content

*Table 13-5: Content of histogram CSV file*

| Normalization on | Normalization off | Number of the data row for value calculation |
|---|---|---|
| MeasurementIndex,1 | MeasurementIndex,1 | |
| MeasurementType,AMPLITUDE | MeasurementType,AMPLITUDE | |
| MeasurementSource,C1 | MeasurementSource,C1 | |
| Normalization,OFF | Normalization,ON | |
| HistogramStart,-0.25024431956999754 | HistogramStart,-0.25024431956999754 | |
| HistogramEnd,0.25012215978499874 | HistogramEnd,0.25012215978499874 | |
| , | , | |
| , | , | |
| , | , | |
| DATA | DATA | |
| 0 | 0 | 1 |
| 3 | 0.003 | 2 |
| 100 | 0.1 | 3 |
| 700 | 0.7 | 4 |
| ... | ... | 5... |

The bins that correspond to the clipping values are excluded in the histogram export. The limits of the histogram correspond to XRangeStart and XRangeEnd.

### 13.5.3 Data conversion

Using the header data, you can calculate the waveform value to which a histogram value belongs. The calculation is valid for vertical and horizontal histograms.

A histogram bin covers a domain of this width:

*BinWidth = (HistogramStart - HistogramEnd) / NumberOfHistogramBins*

BinIndex is the index of the histogram bin of interest, starting with 1. It is the number of the DATA row in the CSV file.

The center value corresponding to a histogram bin is:

*Value = (HistogramStart - HistogramEnd) * (BinIndex - 0.5) / NumberOfHistogramBins + HistogramStart*

The domain covered by the histogram bin is: *Value +/- 0.5 * BinWidth*.

For the example in Table 13-4, the voltage value that corresponds to the histogram value 800 or 0.8 (histogram bin number 4) is:

*(0.25 - (-0.25)) * (4 - 0.5) / 8 - 0.25 = -0.03125 V*

The bin width in this example is *(0.25 - (-0.25)) / 8 = 0.0625 V*.

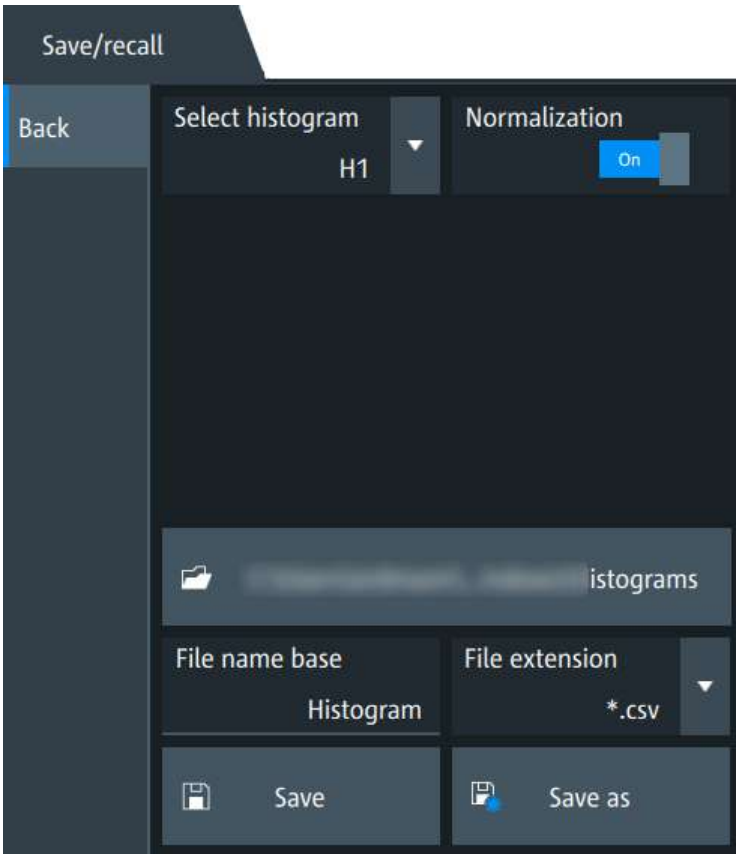
The 4th bin with value 800 covers a voltage domain of

*-0.03125 + 0.5 * 0.0625 = 0 V* to *-0.03125 - 0.5 * 0.0625 = -0.0625 V*

### 13.5.4 Settings for histogram export

Access: "Menu" > "Save/recall" > "Save" tab > "Histogram".

The settings in the dialog are identical for all histogram types.



**Select histogram**
Select the histogram that you want to save. All active waveform histograms and histograms on measurement results are listed.

**Normalization**
If normalization is off, the number of samples in a given bin is exported as integer values. With normalization, the ratio of samples in a given bin to the sum of all samples is exported as float value. The sum of all normalized values is 1.

Remote command:
`EXPort:HISTogram<m>:NORMalize` on page 1132
`EXPort:HMEasurement<m>:NORMalize` on page 1132

**Save settings**
Define the filename and storage location, and start the saving process.

| | |
|---|---|
| "<Folder>" | Opens a file explorer where you can select the directory where the file is saved. |
| "File name base" | Sets a name for the file, without extension. The name is extended with a time stamp when the file is saved. Thus, multiple "Save" actions are possible without changing the filename. |
| "File extension" | The format of the histogram file is always `.csv`. |
| "Save" | Saves the file in the defined folder using the "File name base". If the specified file already exists, it is overwritten with the new data. |
| "Save As" | Opens a file explorer where you select the folder, the file type, enter the filename and save the file. |

A progress bar informs you about the process.

Remote command:
`EXPort:HISTogram<m>:NAME` on page 1133
`EXPort:HISTogram<m>:SAVE` on page 1133
`EXPort:HMEasurement<m>:NAME` on page 1133
`EXPort:HMEasurement<m>:SAVE` on page 1133

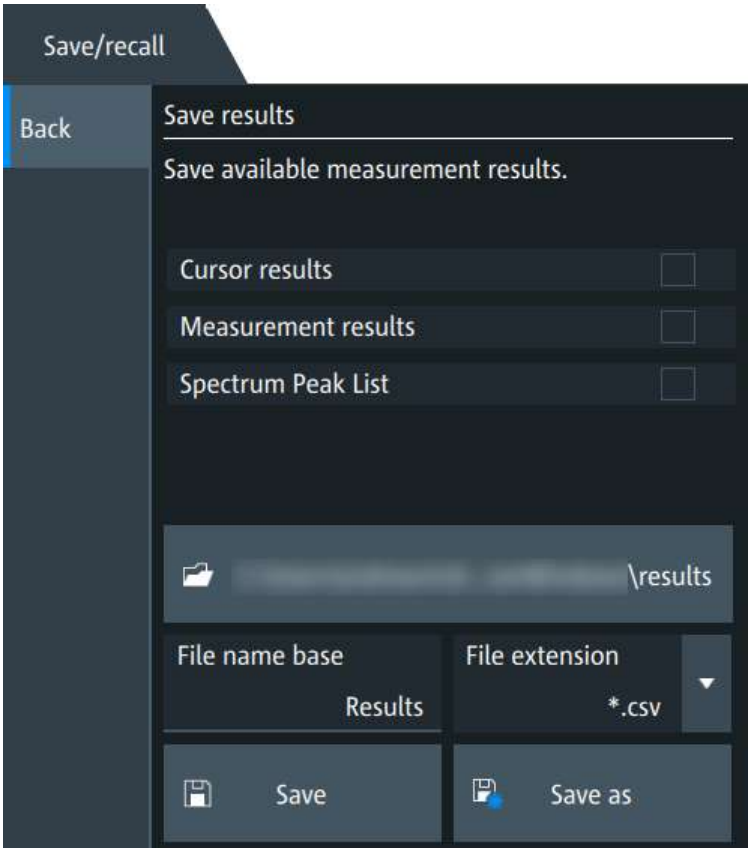
## 13.6 Results

You can save measurement results manually to a file, and automatically on trigger event or mask violation.

### 13.6.1 Save results: settings

Access: "Menu" > "Save/recall" > "Save" tab > "Results".

In this dialog, you define the storage settings for measurement results.




**Save results name**
Select the result tables to be saved: "Cursor results", "Measurement results", "Spectrum peak list". Only active result tables can be selected.

Remote command:
`EXPort:RESult:SELect:CURSor` on page 1136
`EXPort:RESult:SELect:MEASurement` on page 1137
`EXPort:RESult:SELect:PLISt` on page 1137

**Save settings**
You define the details of the filename, extension and directory, and save the file.

A progress bar informs you about the process.

**<Folder> ← Save settings**
Opens a file explorer where you can select the directory where the file is saved.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation. It has **no** effect on `EXPort:RESult:SAVE`.

Remote command:
`EXPort:RESult:AUTonaming:PATH` on page 1136

**File name base ← Save settings**
Sets a name for the file, without extension. The name is extended with a time stamp when the file is saved. Thus, multiple "Save" actions are possible without changing the filename.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation. It has **no** effect on `EXPort:RESult:SAVE`.

Remote command:
`EXPort:RESult:AUTonaming:NAME` on page 1135

**File extension ← Save settings**
Selects the format of the exported file. Result files have always the CSV format.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation. It has **no** effect on `EXPort:RESult:SAVE`.

Remote command:
`EXPort:RESult:AUTonaming:TYPE` on page 1136

**Save ← Save settings**
Saves the file in the defined folder using the "File name base". If the specified file already exists, it is overwritten with the new data.

**Save As ← Save settings**
Opens a file explorer where you select the folder, the file type, enter the filename and save the file.

Remote command:
`EXPort:RESult:NAME` on page 1136
`EXPort:RESult:SAVE` on page 1136

### 13.6.2 Saving results

1. Open "Menu" > "Save/recall".
2. In the "Save" tab, press the "Results" button.
   The "Results" dialog opens.
3. Select the results to be saved. The "Save results" results list shows all result tables that are currently open.
4. Select the target folder. Enter a "File name base" and select "Save".
   Alternatively, select "Save As" and define the target directory and filename in the file selection dialog.

   All selected results are written into one file.

## 13.7 Sessions

Sessions store the general and measurement settings and the data of active channel waveforms and reference waveforms together in a ZIP file. The user-specific display settings for the toolbar, waveform colors and diagrams can also be included.

### 13.7.1 Save a session

Access: "Menu" > "Save/recall" > "Save" tab > "Session"

The session file includes active channel and reference waveforms and the current measurement setup. Before saving the session, consider where you want to use the session data.

- In R&S ScopeStudio, included waveforms and settings can be recalled. If both analog and digital channels are active, only digital channels are saved in the session file and can be loaded. Sessions with only digital waveforms can be saved and loaded.
- When recalling on an MXO oscilloscope, channel waveforms are not imported. Therefore, if you want to recall the session on an instrument, save channel waveforms as references and change the sources of analysis functions to these references before saving the session.

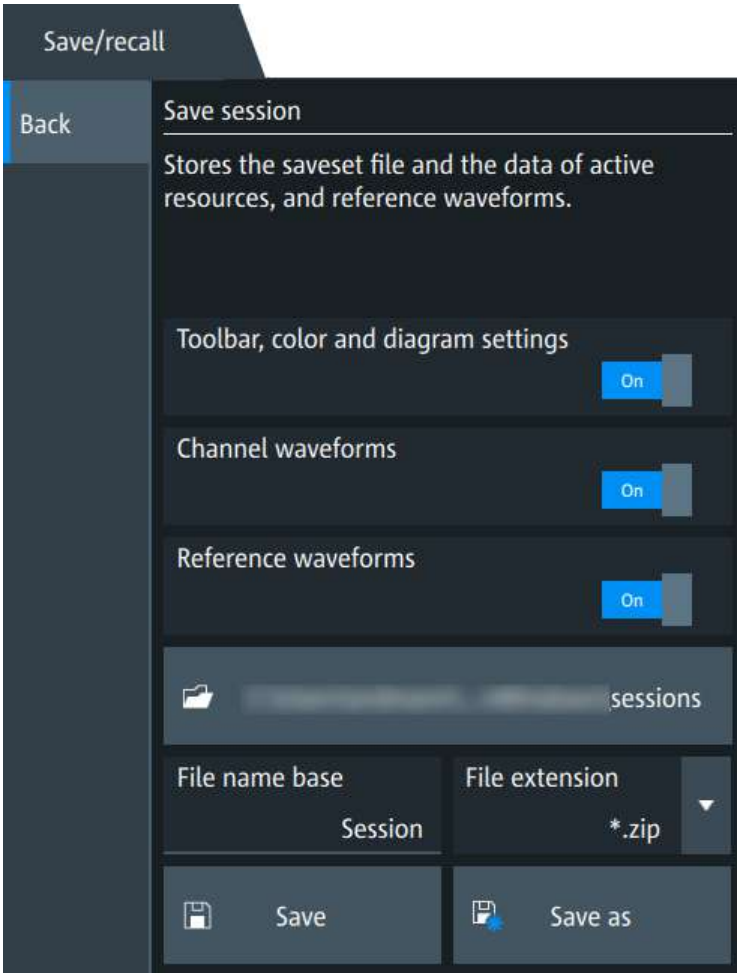



*Figure 13-16: Save session dialog*

**Toolbar, color and diagram settings**
If selected, the user-specific display settings for the toolbar, waveform colors and diagram settings are included in the session file.

Remote command:
`SESSion:USERpref` on page 1135

**Channel waveforms**
Select if you want to include channel waveform data in the session file.

Remote command:
`SESSion:SAVE:CHANnel` on page 1134

**Reference waveforms**
Select if you want to include reference waveform data in the session file.

Remote command:
`SESSion:SAVE:REFerence` on page 1134

**Save settings**
Defines the filename and storage location.

| | |
|---|---|
| "<Folder>" | Opens a file explorer where you can select the directory where the file is saved. |
| "File name base" | Sets a name for the file, without extension. The name is extended with a time stamp when the file is saved. Thus, multiple "Save" actions are possible without changing the filename. |
| "File extension" | The format of the session file is always `.zip`. |
| "Save" | Saves the file in the defined folder using the "File name base". If the specified file already exists, it is overwritten with the new data. |
| "Save As" | Opens a file explorer where you select the folder, the file type, enter the filename and save the file. |

A progress bar informs you about the process.

Remote command:
`SESSion:SAVE[:EXECute]` on page 1135
`SESSion:SAVE:ABORt` on page 1134

## 13.7.2 Load a session

Access: "Menu" > "Save/recall" > "Recall" tab > "Session"

When you recall a session, the settings and the reference waveforms (if included in the session file) are restored. You can also restore some user-specific display settings: toolbar, waveform colors, and diagram settings. Channel waveforms are not imported even if they are included in the session file. Therefore, save channel waveforms as references and change the sources of analysis functions to these references before saving the session.

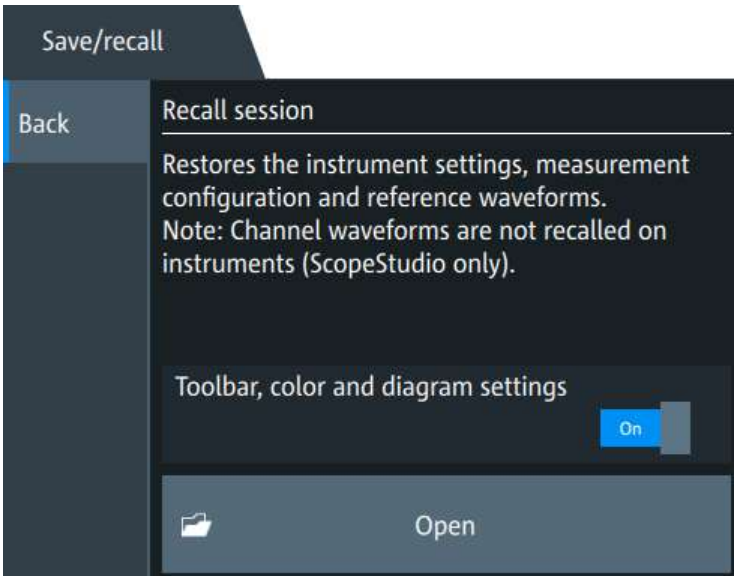



*Figure 13-17: Recall session dialog*

**Toolbar, color and diagram settings**
If selected, the user-specific display settings for the toolbar, waveform colors and diagram settings are included in the session file.

Remote command:
`SESSion:USERpref` on page 1135

**Open**
Opens the file browser dialog. Select the required session file and tap "Open".

Remote command:
`SESSion:LOAD[:EXECute]` on page 1134

## 13.8 Screenshots

To store the graphical results of the measurement, you can save a screenshot of the graphic area. To document current settings, the open dialog box can be included in the screenshot.

If you often save screenshots, try the following shortcuts:

- The "Screenshot" toolbar icon saves the current display to a file according to the settings in "Menu" > "Save/recall" > "Save" tab > "Screenshot".
- You can configure the [Camera] key to save screenshots by a single keypress. See also "Camera hardkey action" on page 107.

If a USB flash drive is connected, the default path of the user data directory is set to the USB flash drive. Thus, you save data to USB flash drive automatically, and you can change the directory in the file explorer at any time.

### Screenshots on a computer using the Web interface

If the MXO 3 is connected to a LAN, you can create and save screenshots of the instrument's display on a computer. See Section 17.2.2.3, "Web control", on page 921.

### Meta information in screenshots

The metadata of the screenshot also contains instrument information. In PNG and JPEG files, meta information is saved as EXIF information and can be read, for example, using the ExifTool.

**Example:**

Reading meta information using the ExifTool.

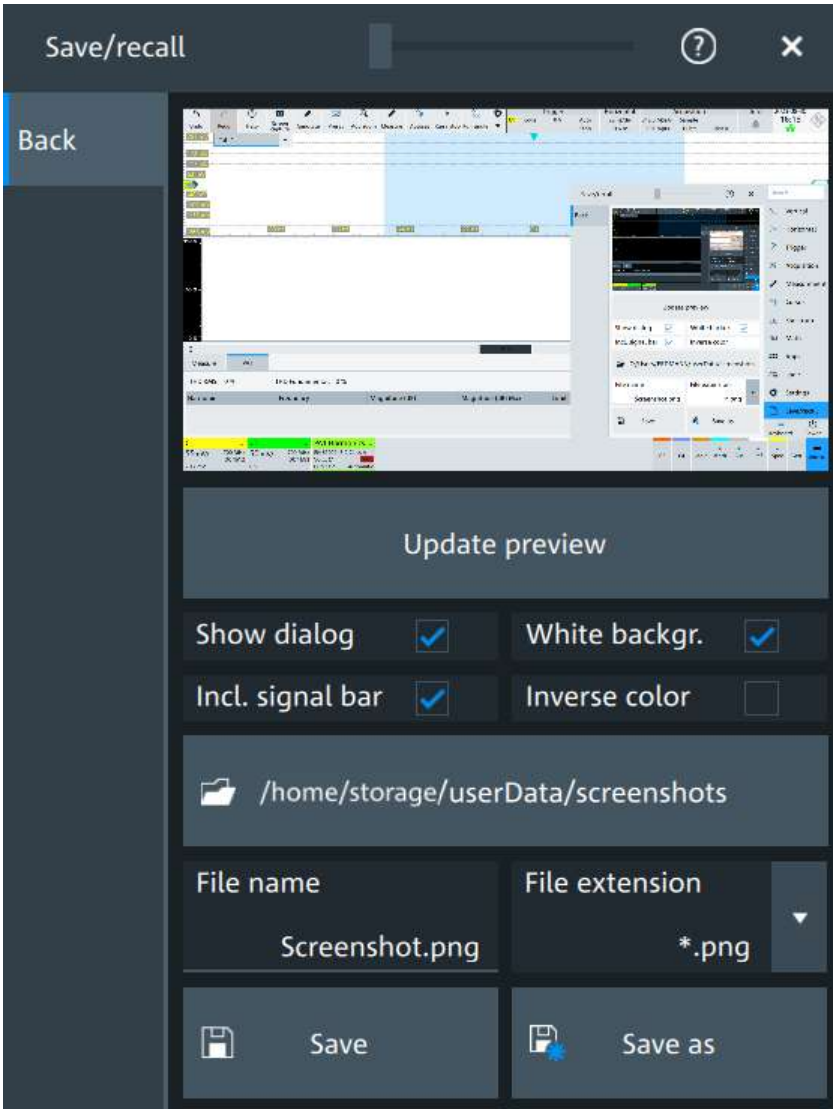
Command: `# exif C:\Screenshot_2020-07-14_0_110551.png`

Result:

```
ExifTool Version Number         : 10.20
File Name                       : Screenshot_2016-07-14_0_110551.png
Directory                       : C:/
File Size                       : 37 kB
File Modification Date/Time     : 2020:07:14 11:05:51+02:00
File Access Date/Time           : 2020:07:14 11:05:51+02:00
File Creation Date/Time         : 2020:07:14 11:05:51+02:00
...
Instrument Firmware Version     : 3.30.0.46
Instrument Material Number      : 1329.7002k44
Instrument Serial Number        : 123456
Image Size                      : 1280x800
Megapixels                      : 1.0
```

### 13.8.1 Screenshot settings

Access: "Menu" > "Save/recall" > "Save" tab > "Screenshot"




In the "Screenshot" dialog box, you configure the image to be saved or included in a report. You select the storage location for screenshot files. The image is created when you open the dialog box, and can be updated at any time.

You can also edit the colors of the image before saving it, and include an open dialog box or the sidebar in the image.

Preview ..... 484
Update ..... 484
Show setup dialog ..... 484
White background ..... 484
Include signal bar ..... 484
Inverse color ..... 484
Save settings ..... 484
└ <Folder> ..... 485
└ File name base ..... 485

└ File extension....485
└ Save....485
└ Save As....485

**Preview**
Shows a preview of the screenshot. The image is created when the dialog box opens.

**Update**
Updates the preview of the screenshot with the current display view, e.g. after changes to the settings have been made, or an additional channel has been activated.

**Show setup dialog**
If enabled, the currently open dialog box is included in the screenshot.

Remote command:
`HCOPy:SSD` on page 1140

**White background**
Inverts the background color, so you can picture waveforms with normal waveform colors on white background.

If both "White background" and "Inverse color" are enabled, the background is inverted twice, and it appears black.

| "White background" | "Inverse color" | Background | Waveform and results |
|---|---|---|---|
| On | Off | White | Screen colors |
| Off | On | White | Inverted colors |
| On | On | Black | Inverted colors |
| Off | Off | Black | Screen colors |

Remote command:
`HCOPy:WBKG` on page 1140

**Include signal bar**
If enabled, the screenshot shows the signal bar below the diagram area.

Remote command:
`HCOPy:ISBA` on page 1140

**Inverse color**
Inverts the colors of the output, i.e. a dark waveform is shown on a white background.

Remote command:
`HCOPy:DEVice<m>:INVerse` on page 1138

**Save settings**
You define the details of the filename, extension and directory, and save the screenshot.

A progress bar informs you about the process.

**<Folder> ← Save settings**
Opens a file explorer where you can select the directory where the file is saved.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation.

Remote command:
`HCOPy:DESTination<m>` on page 1138

**File name base ← Save settings**
Sets a name for the file, without extension. The name is extended with a time stamp when the file is saved. Thus, multiple "Save" actions are possible without changing the filename.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation.

Remote command:
`MMEMory:NAME` on page 1138

**File extension ← Save settings**
Selects the format of the exported file.

You can save the screenshot as `*.jpg` or `*.png` file.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation.

Remote command:
`HCOPy:DEVice<m>:LANGuage` on page 1138

**Save ← Save settings**
Saves the file in the defined folder using the "File name base". If the specified file already exists, it is overwritten with the new data.

Remote command:
`HCOPy:IMMediate<m>:NEXT` on page 1139
`HCOPy:IMMediate<m>` on page 1139

**Save As ← Save settings**
Opens a file explorer where you select the folder, the file type, enter the filename and save the file.

### 13.8.2 Configuring and saving screenshots

You can select which elements are shown in the screenshot, invert the colors and the background color. A preview of the current image is shown for reference.

1. Open the "Menu" > "Save/recall" > "Save" tab > "Screenshot".
2. To enhance the images for later print on white paper, enable "White background" or "Inverse color". If you print this image later on a monochrome printer, you get a grayscaled picture. The contrast of the gray lines depends on waveform colors and the printer used.

3. To change the directory, tap "<Folder>", and configure the path.

   The symbols of often used target folders are listed on the left of the file explorer. By default, screenshots are saved in the `/home/storage/userData/ScreenShots` directory.

4. Select the "File extension".

5. Tap "Save".

   The file is saved with an automatic name in the specified folder.

6. To save further screenshots, use one of the following ways:

   - Configure the [Camera] key. Press the key to save a screenshot.
     See also Camera hardkey action.
   - Add the "Screenshot" icon to the toolbar. Tap the icon to save an image.
     See Section 4.4.2, "Configuring the toolbar", on page 63.
   - Tap "Save" in the "Screenshot" dialog. The filename contains a number, which is automatically counted up.
   - To save the image with a dedicated filename or to another directory, tap "Save As" in the "Screenshot" dialog.
     Select the path, enter a filename, and tap "Save".

### 13.8.3 Image viewer

You can look at the saved screenshots directly, without transferring them to a computer or using a dedicated software.

1. Add the "Image viewer" icon to the toolbar.

2. To view the saved screenshots, select the icon.

   You can scroll the images, browse in other folders, and use the "Auto play" function for viewing.

## 13.9 File browser dialog

The file browser dialog provides a file explorer from which you can select a file to load or to save data to. You can also manage your files in this dialog.

Access: "Menu" > "Apps" > "File browser".




**Path**
Navigate the path elements to change the current folder. The default folder is defined in Section 5.8.1, "Autonaming", on page 117.

You can save the data in a local folder, to an external storage device (usually a USB flash drive), or to a folder on a connected network drive. The path list provides all available drives and folders.

On the left, shortcut icons provide access to often used folders.

**Toolbar**
The toolbar on the top provides various functions for file management.

| | |
|---|---|
| "Filter" | Displays only the files and folders that match the current search term. |
| "Delete" | Deletes the selected file or folder. |
| "New folder" | Creates new folder in the current directory. |
| "Rename" | Renames the selected file. |
| "Find" | Searches for files within the current folder. |
| "Copy" | Copies the selected file. |
| "Paste" | Pastes the file. |

**File name**
The file name to be loaded or stored to. Tap the file name, or tap the keyboard icon to enter the file name using the online keyboard.

The default file name for new files is defined in the "Autonaming" tab, see Section 5.8.1, "Autonaming", on page 117.

**File type**
The file extension of the file to be loaded or stored to.

**Save, Select**
Selects the specified file for the open or save operation and closes the dialog box.

**Cancel**
Closes the dialog box without selecting a file.

# 14 Protocol analysis

Using the serial protocol options for the MXO 3, you can analyze various serial protocols.





- Basics of protocol analysis....489
- SPI bus (IC electronics, option R&S MXO3-K510)....514
- QUAD-SPI bus (IC electronics, option R&S MXO3-K510)....533
- OCTO-SPI bus (IC electronics, option R&S MXO3-K510)....549
- I²C (IC electronics, option R&S MXO3-K510)....565
- I3C (IC electronics, option R&S MXO3-K550)....590
- UART (IC electronics, option R&S MXO3-K510)....610
- NRZ clocked & unclocked (option R&S MXO3-K510)....631
- Manchester (option R&S MXO3-K510)....653

- CAN (automotive electronics, option R&S MXO3-K520)..... 674
- LIN (automotive electronics, option R&S MXO3-K520)..... 704
- SENT (automotive electronics, option R&S MXO3-K520)..... 724
- ARINC 429 (aerospace electronics, option R&S MXO3-K530).....747
- MIL-1553 (aerospace electronics, option R&S MXO3-K530).....764
- SpaceWire (aerospace communications, option R&S MXO3-K530)..... 777
- SPMI (mobile electronics, option R&S MXO3-K550)..... 794
- RFFE (mobile electronics, option R&S MXO3-K550).....812
- 10BASE-T1S (automotive Ethernet, option R&S MXO3-K560)..... 832
- USB 1.1 (option R&S MXO3-K570)..... 851
- USB 2.0 (option R&S MXO3-K570)..... 874

## 14.1 Basics of protocol analysis

With the MXO 3 oscilloscope, you can decode selected protocols.

With dual-path protocol analysis, you can set the instrument sample rate for the waveform path and the oscilloscope automatically uses another internal decoupled sample rate for the decoding path. Even with very slow sample rates, the protocol data is correctly decoded.

With the MXO 3, you can use deep memory to capture more packets. The oscilloscope can capture long time periods where the cause and result are distanced in time. Over the entire capture, signal detail is time-correlated with packet content for fast debug.

Before you can analyze a serial signal, the bus has to be configured according to the protocol and specifics of the signal. The configuration contains:

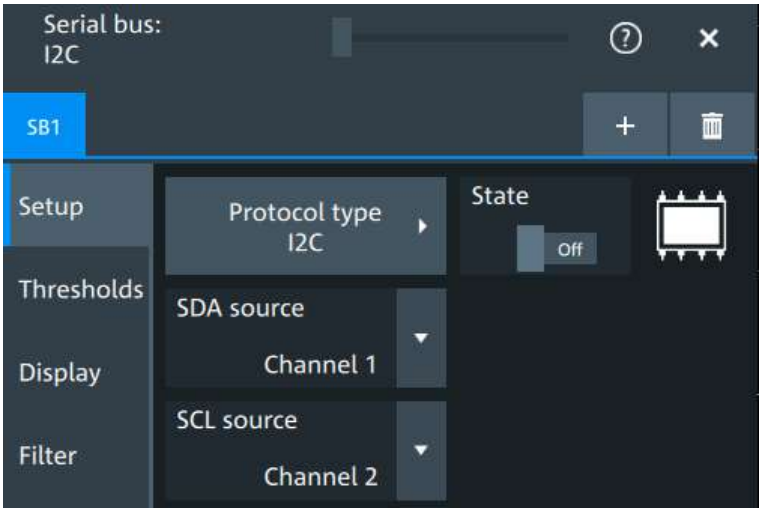
- Assignment of the data and clock lines to the input channels
- Logical thresholds
- Protocol-specific settings

Serial data can be analyzed in several ways:

- Triggering: You can trigger on various events that are typical for the selected protocol type, for example, on start and stop of messages, or on specified data patterns in the message.
  Triggering on a trigger event sequence is not supported, and holdoff settings are not available.
- Protocol decoding: The digitized signal data is displayed on the screen together with the decoded content of the messages in readable form, and the decode results are listed in a table.

### 14.1.1 Setup - general settings

For all protocols, configuration starts with the selection of the serial bus and the protocol.




Configuration settings are protocol-specific. They are described in the related sections.

Make sure that the tab of the correct serial bus is selected.

**SB1 /SB2 /SB3 /SB4**
Select the correct bus tab before you enter the settings.

**Add serial bus, Delete serial bus**
Use the icons on the upper right to add a new bus, or to delete the selected serial bus.

Remote command:
`SBUS<sb>:RMSBus` on page 1364 (delete)

**Protocol type**
Selects the protocol type to be decoded.

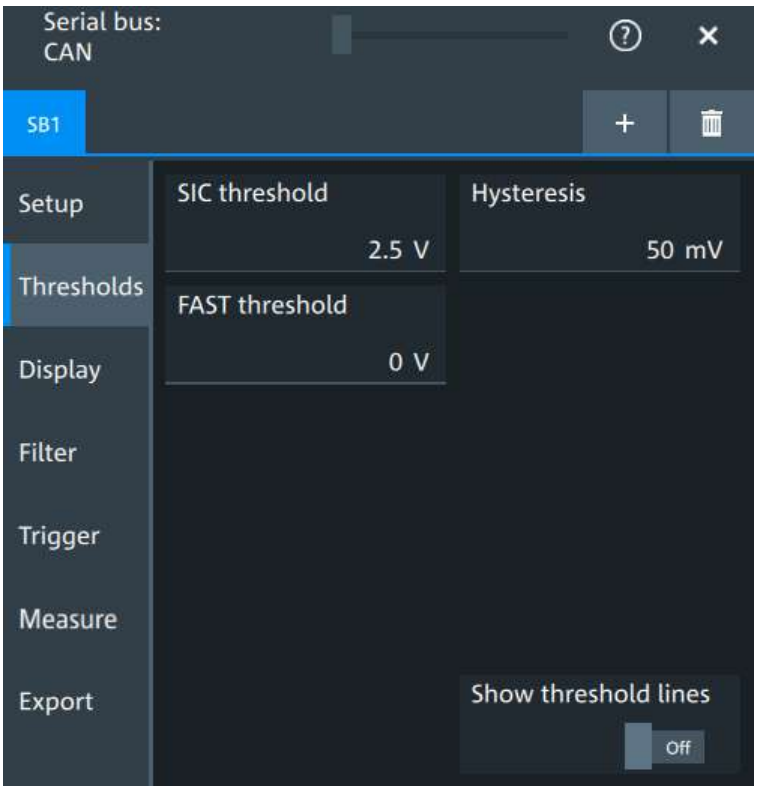
Remote command:
`SBUS<sb>:TYPE` on page 1362

**State**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
`SBUS<sb>[:STATe]` on page 1362

## 14.1.2 Thresholds

In the "Thresholds" tab, you can set values for the threshold and hysteresis of the signal.




**Show threshold lines**
Enables the display of the threshold lines in the diagram.

The label of the threshold line is set according to the signal that it is referring to. It is displayed in the color of the signal channel.

**Example:**

When you enable "Show threshold lines", the threshold lines are shown on the display, and also the hysteresis ranges are shown for a few seconds. Afterwards only the threshold lines are visible. When you click the diagram near or on a threshold line, its hysteresis is shown again briefly.

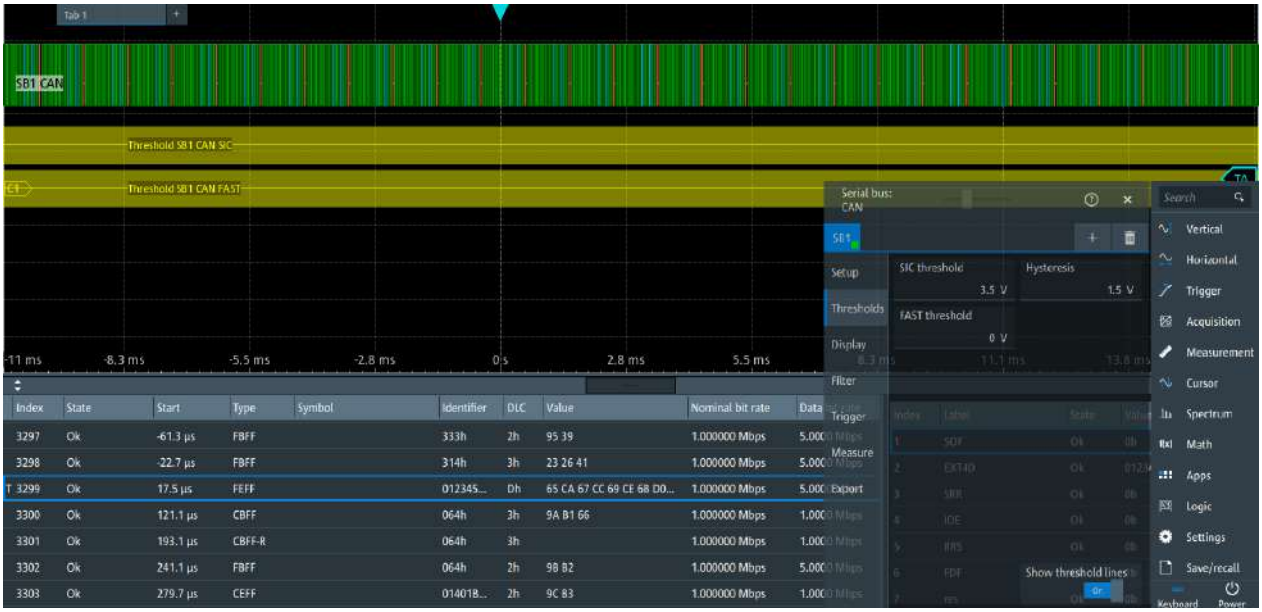



*Figure 14-1: Threshold lines, shortly after enabling Show threshold lines*

Remote command:
`SBUS<sb>:THReshold` on page 1364

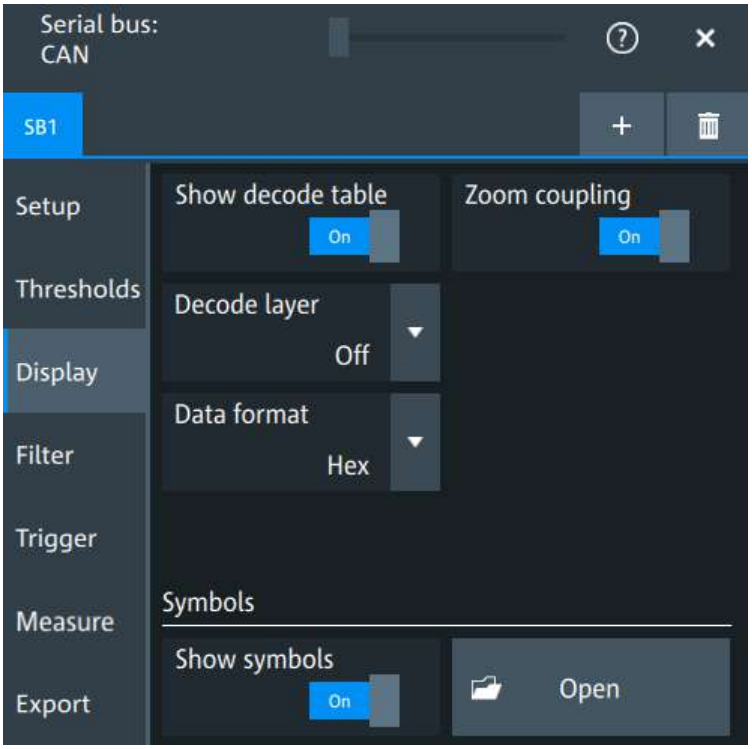
## 14.1.3 Display

The following subsections describe the display settings and its zoom-coupling features.

- Display settings ..... 492
- Display synchronization ..... 494

### 14.1.3.1 Display settings

For all protocols, you can select how to display the decoded signal and its results table along with the waveform diagram.




**Show decode table**
Opens a table with decoded data of the serial signal. The function requires the option for the analyzed protocol.

Decode results are protocol-specific.

Remote command:
`SBUS<sb>:RESult` on page 1364

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

The available values are protocol-specific. For example:

| | |
|---|---|
| "Off" | Disables the display of the decode layer. |
| "Edges" | Enables the display of all edges. |
| "Bits" | Enables the display of all bits. |
| "Words" | Enables the display of all words. |

**Data format**
Sets the data format for the values displayed in the decode table and in the combs of the decoded signal.

Available formats are: hex, octal, binary, ASCii, signed, unsigned.

For more information on the available formats, see Section 14.1.10, "Bit pattern editor", on page 512.

Remote command:
SBUS<sb>:FORMat on page 1363

**Zoom coupling**
Available only, if Show decode table is enabled. If you enable "Zoom coupling", the software synchronizes the decode zoom window with the result tables, as described in Section 14.1.3.2, "Display synchronization", on page 494.

Remote command:
SBUS<sb>:ZCOupling on page 1365

**Symbols**
Symbol lists are protocol-specific and not available for all protocols. Symbols are described in the respective configuration section of the protocol. For finding symbols in the decode table, enable ""Show symbols"". See also Section 14.1.3.2, "Display synchronization", on page 494.

The RC commands for this function are also protocol-specific, for example in the I²C protocol: SBUS<sb>:I2C:SYMBols enables symbols, and SBUS<sb>:I2C:NEWList loads the symbols translation file.

### 14.1.3.2 Display synchronization

This section describes the synchronization (or coupling) of the following components in the display:

- The waveform diagram and its zoom window
- Decode results table with decoded frames
- Frame details table with fields of selected frame

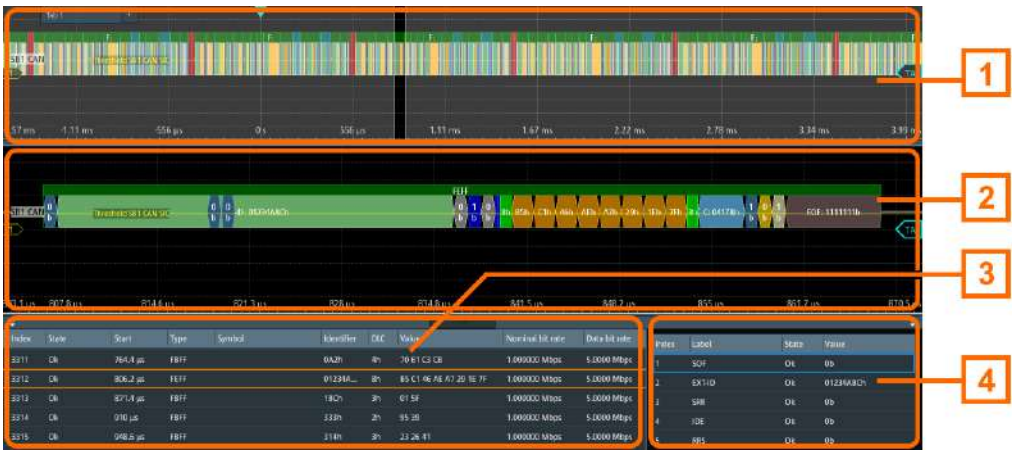



*Figure 14-2: Main components in the decoded results display, here for the CAN protocol*

1 = Waveform diagram, it shows an overview of some part of the acquisition, or even all of it
2 = Zoom window, it shows the zoom area selected in the waveform diagram
3 = Decode results table, it lists all frames of the acquisition
4 = Frame details table, it lists all fields of the frame selected in the results table

Prerequisites for synchronizing displayed results:

- State is enabled in the "Setup" tab.
- Show decode table is enabled in the "Display" tab.
- Zoom coupling is enabled in the "Display" tab.
- Decoded results are available.
  The dialog shows the results for frames in the results table and for fields in the details table.

When you select a row in one of the tables, hence a decoded frame or field, the software automatically synchronizes the zoom window to display this frame or field on the screen with a margin at each side. For details of the synchronization and margin, see Figure 14-3 and Figure 14-4. The software also synchronizes the waveform diagram with the zoom window.

When you move the zoom window to the left or right, the software updates the results table to highlight the window's center frame, without resizing.

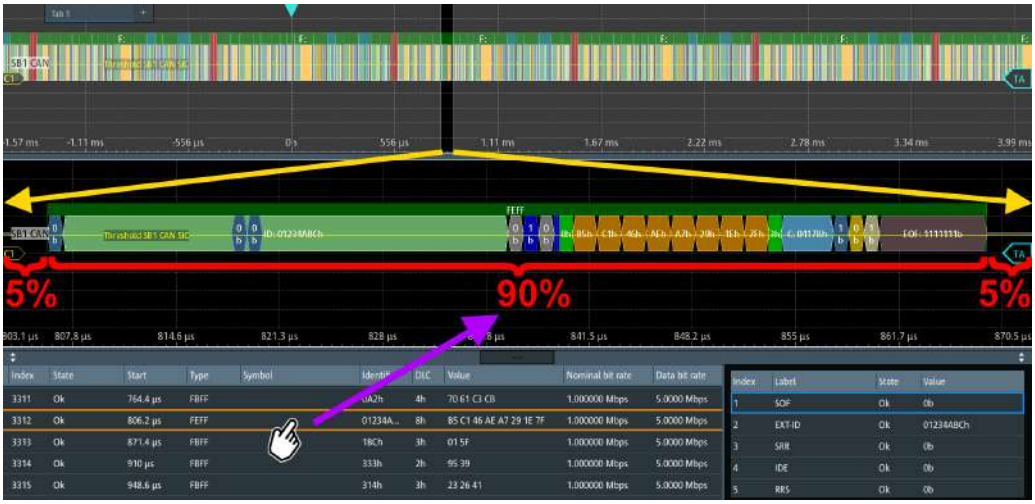



***Figure 14-3: Zoom coupling of selected frame (left table)***

Pink arrow = Selected frame covers 90% of the zoom window's width (red scale), centered on the screen
Yellow arrows = Zoom segment, automatically selected in the diagram, covers the full zoom window's width

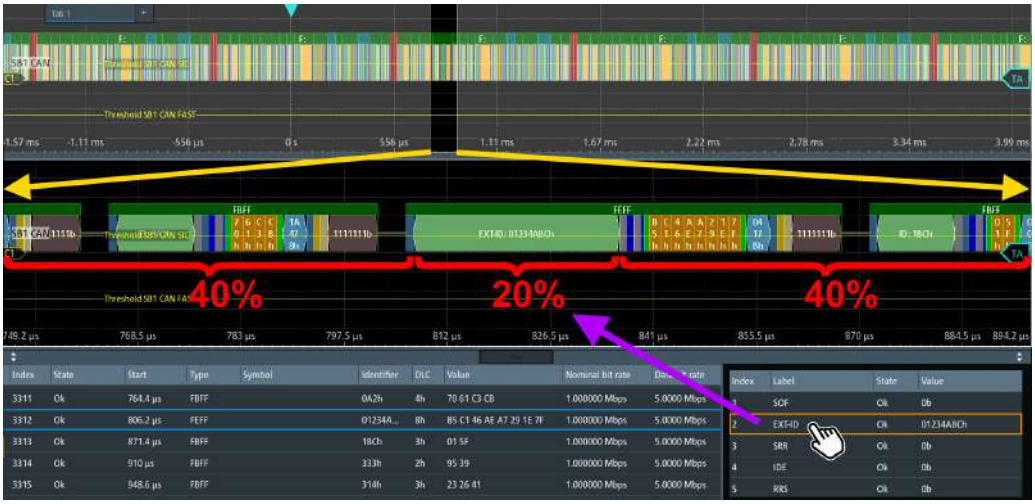
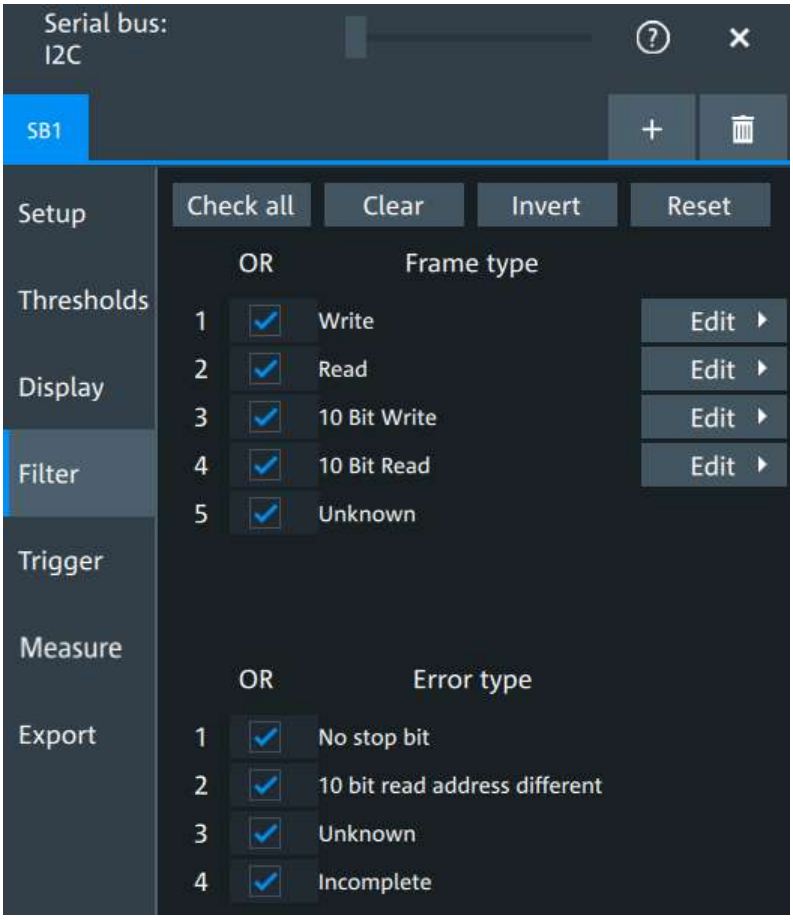



*Figure 14-4: Zoom coupling of selected field (right table)*

Pink arrow = Selected field covers 20% of the zoom window's width (red scale), centered on the screen
Yellow arrows = Zoom segment, automatically selected in the diagram, covers the full zoom window's width

### 14.1.4 Filter

In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.




The settings are protocol-specific and are described in the corresponding protocol section.

### 14.1.5 Format

The format tab allows you to configure certain aspects of how the decoding results are translated and visualized. The defined formats are applied by the filter, the software trigger, and the measurements.

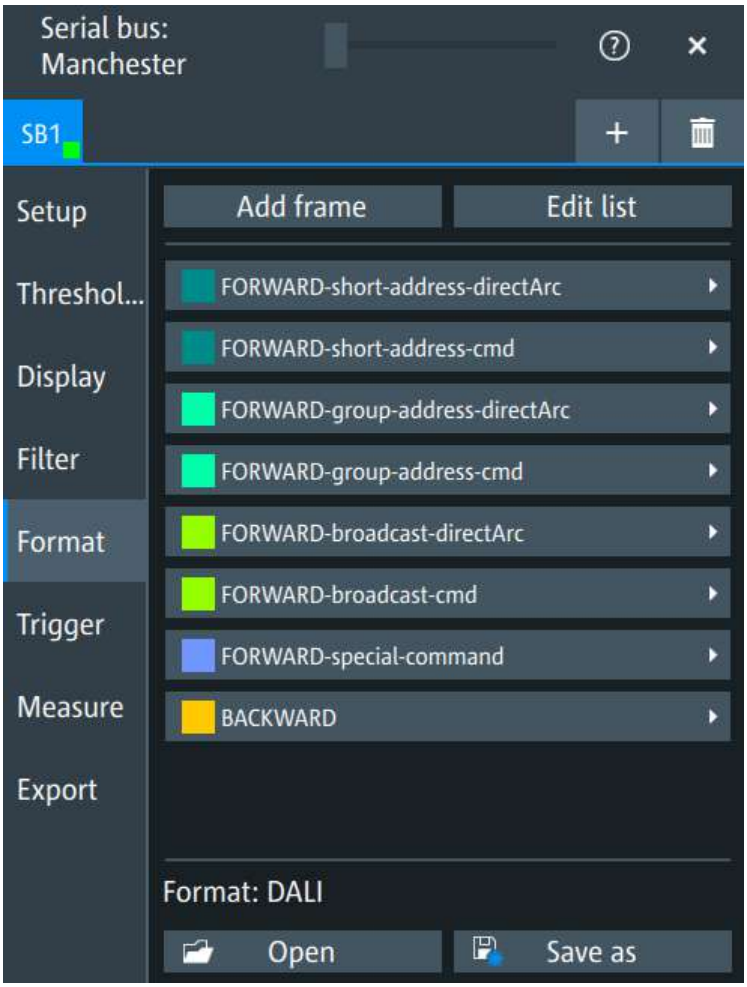
The format configuration differs depending on the protocol.

- Format for custom protocols ..... 497
- Format for QUAD-SPI and OCTO-SPI ..... 500
- Format for all other protocols ..... 502

#### 14.1.5.1 Format for custom protocols

For the custom protocols NRZ-(un)clocked, Manchester, 8b/10b, the format tab fully specifies how frames and their fields are identified and created during the decoding. The decoder reads the incoming raw bitstream, applies the custom frame list defined in the tab, and instantiates frames and their fields according to the field definitions. You

can also use functions to calculate new values based on the data fields, for example see also "Numeric format" on page 667.




Each frame format occupies one line in the frame list. The frame identification is executed top down in the frame list.

Describing a frame format requires assigning it a name and specifying field descriptions, which must be sequential and complete. No gaps are allowed, because the field descriptions are used to calculate the start position of the next frame.

A frame is identified when all user-defined conditions of the field descriptions are met. These conditions can be regarded as related by the Boolean `AND` operator.

A condition can also locate a synchronization pattern, specified by the equal operator in the field. For example, if you define a "Preamble" field with the condition *=FFFFFFFF*, the decoder scans the data for this pattern, and then synchronizes to it.

If no frame format is suitable to identify an incoming frame, by default such a "missed" frame is reproduced as "Undescribed Bits" in the honeycomb display. These bits are not shown in the results table.

If you have not defined an "equal" operator for any field of a frame format, you can use this description to positively identify each kind of frame. Hence, it "catches" every

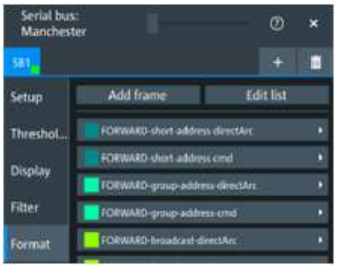
frame, even if there are other frame formats to follow in the frame list. Therefore, if you use a "catch all" frame format, we recommend to place it at the end of the frame list, otherwise it overwrites any subsequent frame format. However, instead of using a "catch all" frame format, consider using the default "Undescribed Bits" display mentioned above.

It is in your responsibility to define unambiguous formats for each frame type.

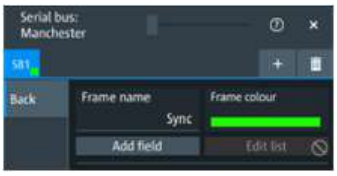
If you do not want to start from scratch, you can load a predefined format table via "Menu" > "Apps" > "Protocol" tab > select "DALI" or "MDIO". You can always configure additional formats with the following procedure.

**To configure formats for custom protocols**

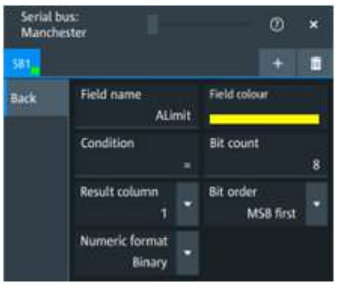
1. Navigate to "Menu" > "Apps" > "Protocol" tab > select a custom protocol > "Format".
2. Select "Add frame".




3. Specify the frame details: Name and color.




4. Select "Add field".
5. Specify the field details: Name, color and display features.




6. Add a "Condition".

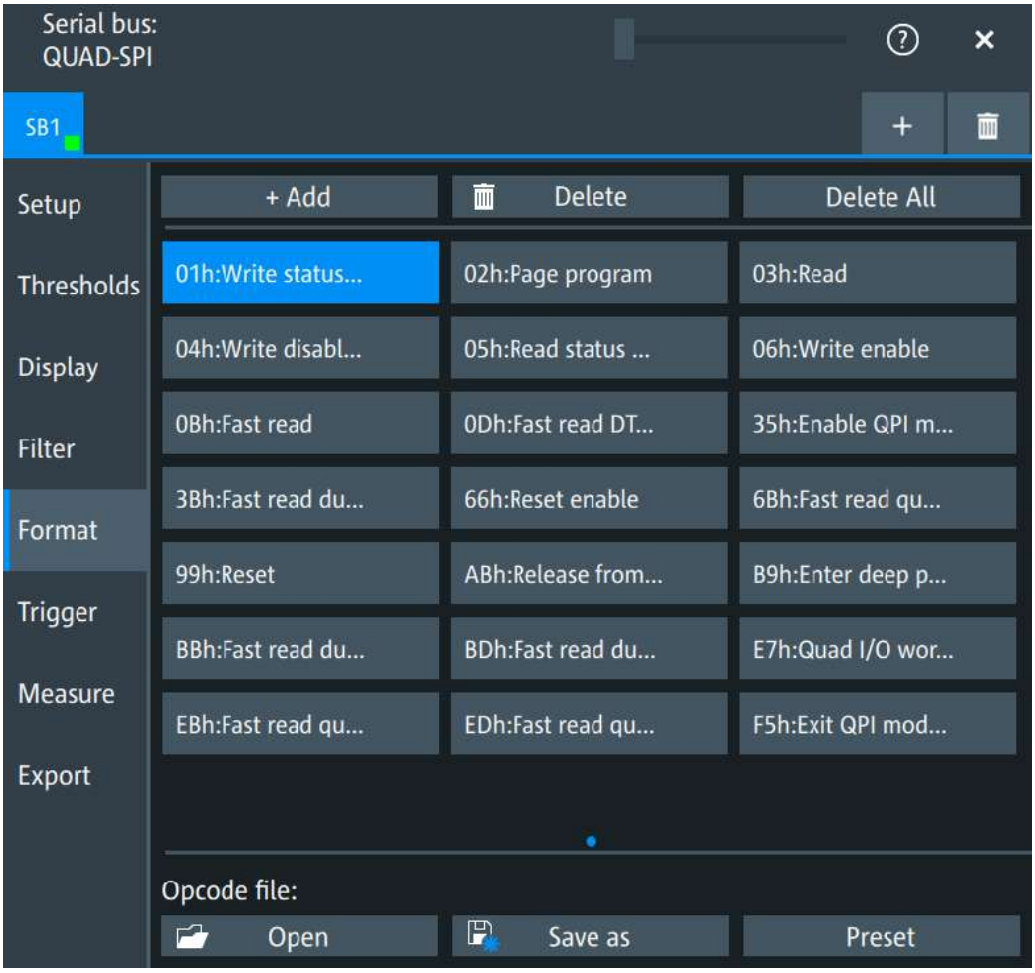
   **Note:** The fields must be sequential and complete. No gaps are allowed, because the field definitions are used to calculate the start position of the next frame.

7. Select "Back".
8. Repeat the previous steps to add and define more fields.
9. Select "Back".

   The "Decode results" table, filter and trigger tabs display the new fields instead of the original fields.

#### 14.1.5.2 Format for QUAD-SPI and OCTO-SPI

For QUAD-SPI and OCTO SPI, the format tab provides a customizable list of opcode and the formatting of the opcode data.




Opcode is a specific command byte sent from the microcontroller to the QUAD-SPI or OCTO-SPI device to initiate a particular action, such as reading, writing or erasing data. The opcodes vary by device manufacturer and memory model.

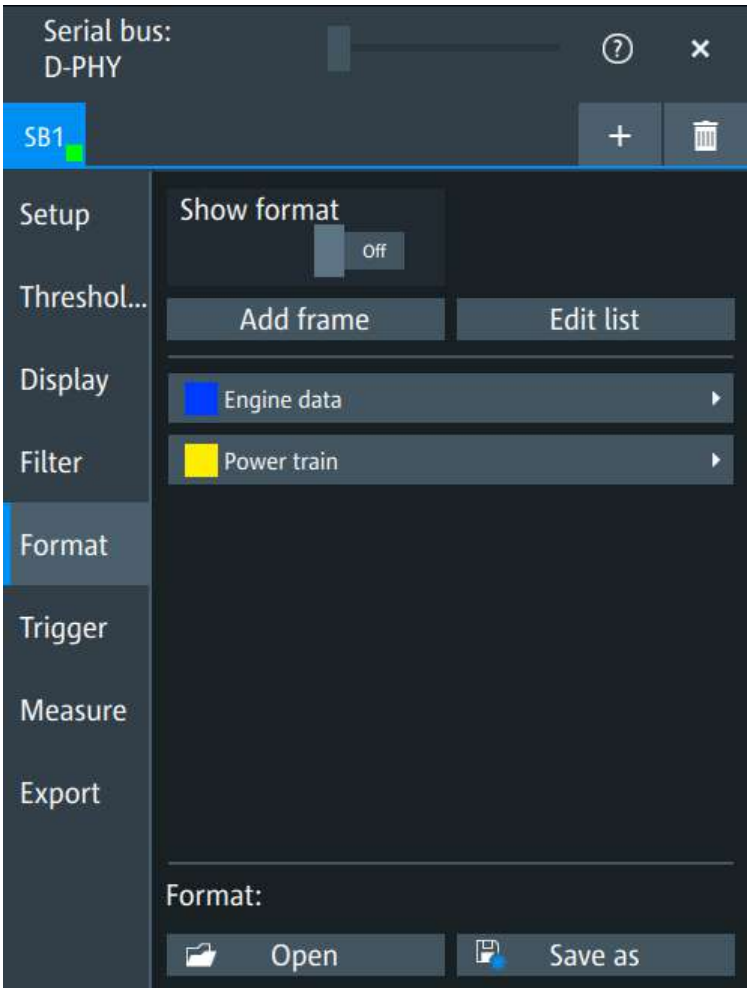
Opcode files are in JSON format, and contain frame formats for each hexadecimal opcode value.

The following is an example from a JSON file:

```
{
   "Firmware Version": "2.2.0.0",
   "Description": "Rohde Schwarz Inc. oscilloscope Quadspi Opcode data",
   "Opcode Version": 1,
   "Opcodes": {
    "0x05,RDSR1": [
        {
            "name": "data",
            "bytes": 0,
            "ddr": false,
            "lanes": 1
        }
    ],
    "0x07,RDSR2": [
        {
            "name": "data",
            "bytes": 0,
            "ddr": false,
            "lanes": 1
        }
    ],
    "0x35,RDCR1": [
        {
            "name": "data",
            "bytes": 0,
            "ddr": false,
            "lanes": 1
        }
    ],
    "0x0D,DDRFAST READ": [
        {
            "name": "address",
            "bytes": 3,
            "ddr": true,
            "lanes": 1
        },
        {
            "name": "data",
            "bytes": 0,
            "ddr": true,
            "lanes": 1
        }
    ]
    }
}
```

#### 14.1.5.3 Format for all other protocols

For protocols that are not custom protocols, the protocol decoder identifies frames and their fields in the bit stream. With the format tab, you can identify specific data fields in the decoded frames and further characterize it by redefining them to new fields of user-defined properties, such as name, color, length, and format. You can also use functions to calculate new values based on the data fields, for example see also "Numeric format" on page 525.




Each frame format occupies one line in the frame list. The frame identification is executed top down in the frame list.

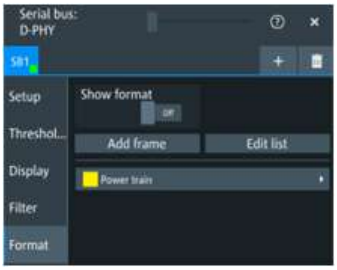
The following applies for the field descriptions:

- Define individual bit start positions and lengths; any gaps in the bit order are automatically filled with “undescribed” fields.
- Bits do not need to be consecutive, but must not be overlapping.
- The upper boarder of a field (bit start position + length) must not exceed the total payload length.

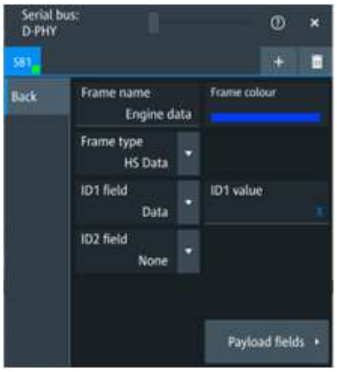
**To configure formats for protocols like CAN, SENT**

Every change in the format configuration triggers a re-decode. We recommend pre-defining the frame formats before loading the files and starting the acquisition.

1. Navigate to "Menu" > "Apps" > "Protocol" tab > select a higher layer protocol > "Format".
2. Select "Add frame".




3. Specify the frame details: Name, color, frame type.




4. Select an "ID<n> field".
5. Add a condition "ID<n> value" to the respective "ID<n> field".
6. Add fields to interpret the payload (optional):
   a) Select "Payload fields".
   b) Select "Add field".



c) Specify the field details: Name, color and display features.




**Note:** Gaps in the bit order are filled as "Undescribed" fields and not shown in the honey-comb view.

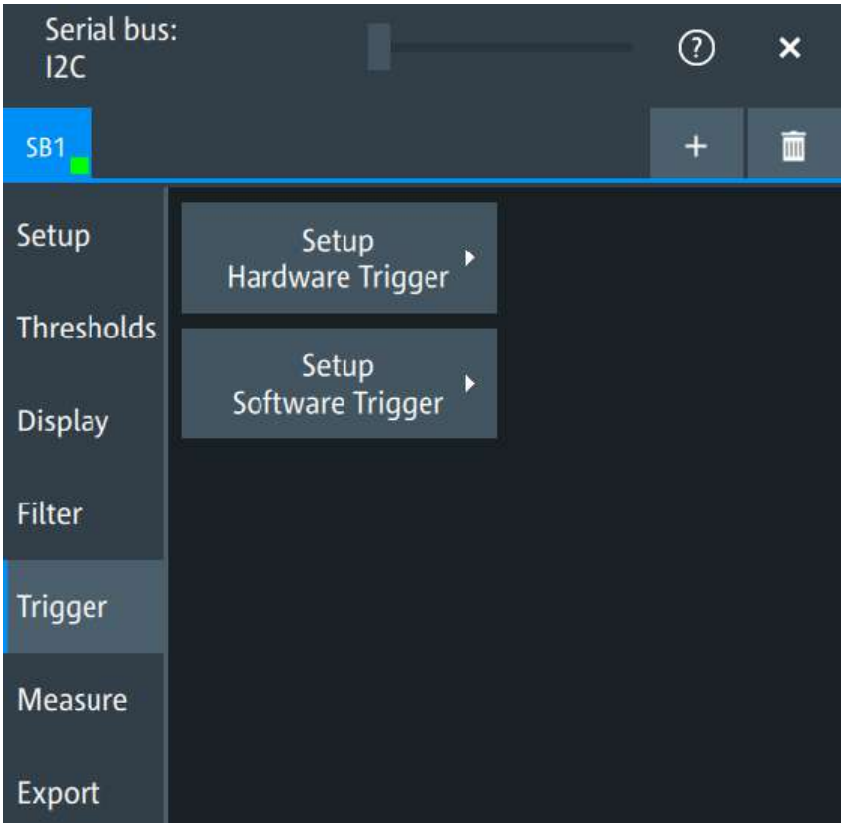
7. Select "Back".
8. Repeat the previous steps to add and define more fields.
9. Select "Back".
10. Enable "Show format" to turn on the format display.

    The "Decode results" table, filter and trigger tabs display the new fields instead of the original payload fields.

### 14.1.6 Trigger

The "Trigger" tab provides functionality to quickly set up protocol triggers and navigates to the "Trigger" dialogs for further settings.

A trigger is a function that determines when the instrument starts capturing a signal. There are two primary types of triggers on the MXO 3, depending on the selected protocol: software trigger and hardware trigger.




#### 14.1.6.1 Hardware trigger

A hardware trigger is a trigger that is generated by the oscilloscope's hardware circuitry.

The protocol hardware trigger is available only for the few protocols that support it (listed below). A single click of the "Setup Hardware Trigger" button in the protocol dialog is equivalent with the following actions in the "Trigger" dialog:

1. "Trigger on" > "Single event".
2. "Source" > selects the active serial bus, for example "Serial bus 1".
3. "Type" > selects the most simple protocol-specific trigger type:
   - I²C: "Start"
   - SPI: "Start of frame"
   - UART: "Start bit"
   - LIN: "Start of frame"
   - CAN: "Start of frame"

   This action also executes the "Type" command and resets the trigger settings.

   Optionally proceed with the trigger settings in the "Trigger" dialog.

#### 14.1.6.2 Software trigger

The protocol software trigger is a sequence "A → Serial bus" of events. Hence, it consists of two subsequent events: A-trigger and a serial bus trigger.

**Loss of settings**

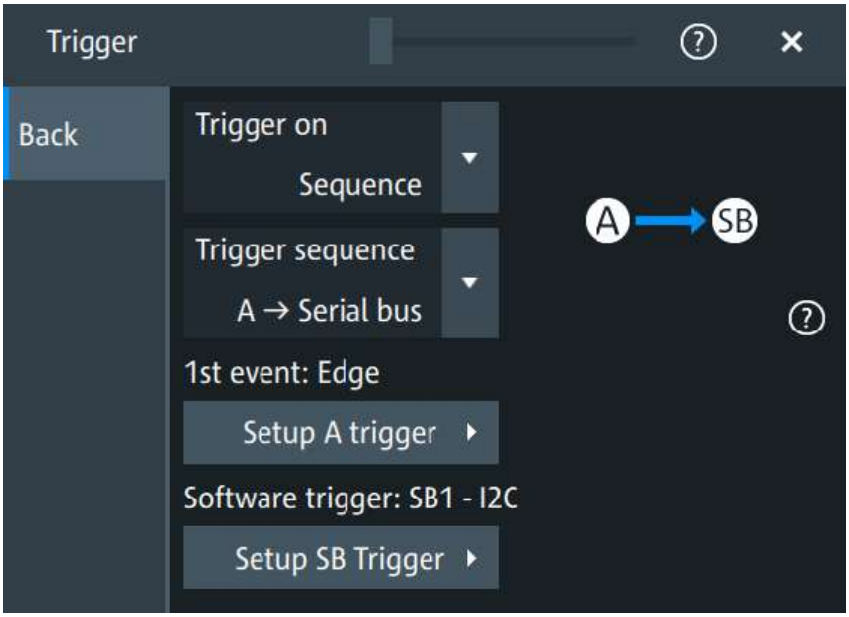
If you have already configured the SW trigger, do not click "Software Trigger" in the protocol dialog, because it overwrites your settings. Instead, use the "Trigger" dialog.

For an initial configuration, a single click of the "Software Trigger" button is equivalent with the following actions in the "Trigger" dialog:

1. "Trigger on" > "Sequence".
2. "Trigger sequence" > "A → Serial bus"
3. Configures the 1st event:
   a) Sets the "1st event" ("A trigger") to a protocol-specific trigger that is most suited for this protocol, for example "Edge".
   b) Sets the "Level" to the threshold level defined in the protocol dialog.
   c) Sets other parameters, depending on the event type and protocol requirements, for example "Slope" for "Edge", or "Time" for "Timeout".
4. Configures the software trigger:
   Preselects recommended frame types and error types, which you can change later.

After the A-trigger conditions have been met, the serial bus trigger with independent conditions is enabled. The oscilloscope waits until one or a specified number of serial bus trigger conditions occur. The serial bus trigger can only cause the trigger event if it occurs after the A-trigger.

All trigger sequences require that analog input channels C<n> are set as trigger sources for all events. The oscilloscope checks all trigger settings for compatibility and adjusts them if they do not fit.




To indicate the 1st (main, general) trigger position in the captured signal, the top of the display shows a triangular trigger marker (1 in Figure 14-5). If you enable displaying the results table (see "Show decode table" on page 493), the triggered frame is marked in the table by the letter "T" left of the frame index (2 in Figure 14-5):

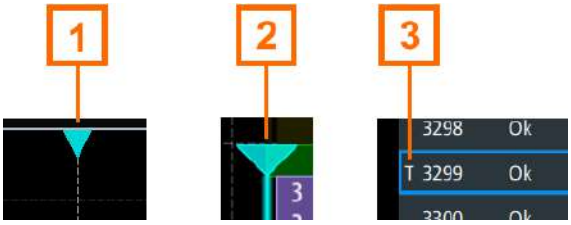



***Figure 14-5: Trigger markers***

1 = Triangular main trigger marker at the top of the waveform display, indicates the position of the single trigger or the position of the 1st trigger event of a trigger sequence
2 = Triangular secondary trigger marker with bold line in the zoom diagram of the decoded signal, indicates the 2nd event trigger position, hence the software trigger
3 = Letter "T" marks the 2nd event trigger position (the triggered frame) in the decode results table

The following scenarios are possible:

- 1st case, triggering on a *single event* while the source is a serial bus:
  - The main trigger marker indicates the single event trigger position.
  - There is no secondary trigger marker in the decoded signal diagram.
  - The "T" trigger marker in the protocol decode results table corresponds to the main trigger marker.

  This scenario uses the hardware trigger, which is not available for all protocols.
- 2nd case, triggering on a *single event* while the source is *not* a serial bus:
  - The main trigger marker indicates the single event trigger position.
  - There is no secondary trigger marker in the decoded signal diagram.

  - There is no "T" trigger marker in the protocol decode results table.
- 3rd case, triggering on a *sequence* of "Trigger A" and "serial bus event":
  - The main trigger marker indicates the 1st event trigger position.
  - The secondary trigger marker indicates the 2nd event trigger position (software trigger).
  - The "T" trigger marker in the protocol decode results table corresponds to the 2nd event trigger position (software trigger).

Figure 14-6 shows an example of these markers.

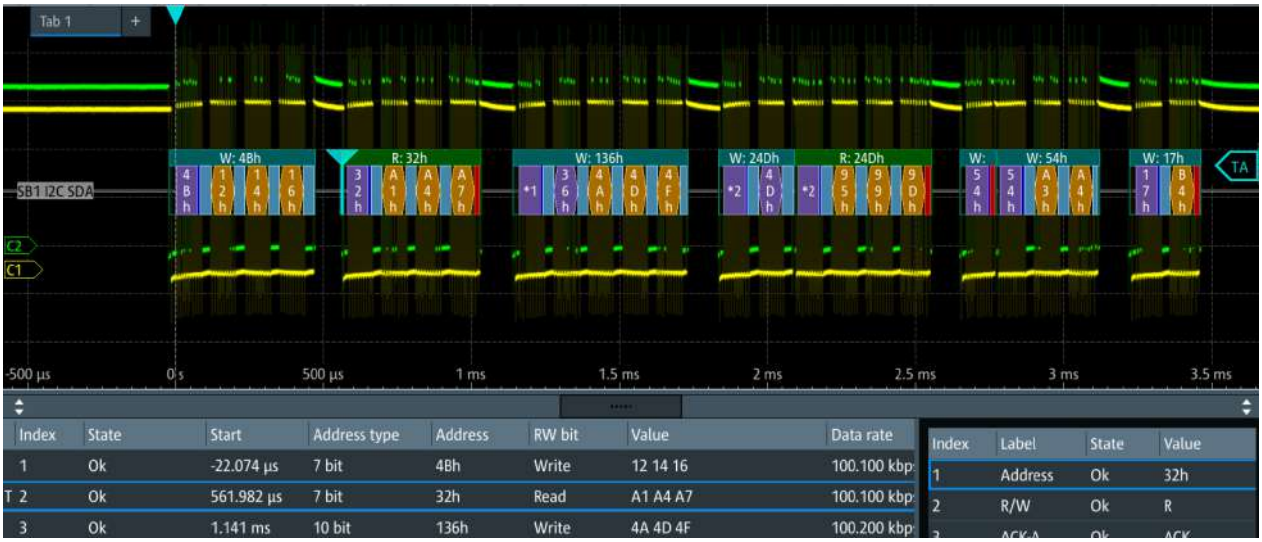



***Figure 14-6: Example of sequence trigger in I2C signal with "Trigger A" (interval > 1 ms) and software trigger (first "Read" frame)***

See also "Source" on page 165.

**Check all, Clear, Invert, Preset**

The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
The remote control commands for these functions are protocol-specific. Refer to the descriptions of the software trigger setup for each individual protocol.

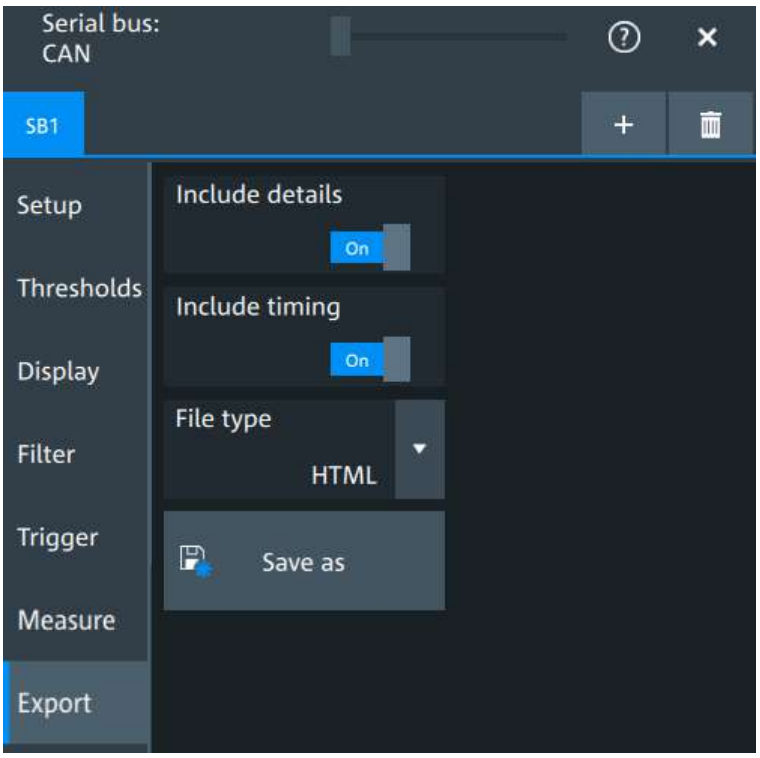
### 14.1.7 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7, "Protocol measurements (option R&S MXO3-K500)", on page 298.

### 14.1.8 Export

In the "Export results" tab, you can export the results and all details of the selected protocol.




**Include details**
If enabled, includes the detailed results for all frames in the export result file.

Remote command:
`SBUS<sb>:EXPResult:DETail` on page 1365

**Include timing**
If enabled, includes the frame timing in the export result file.

Remote command:
`SBUS<sb>:EXPResult:TIME` on page 1366

**File type**
Selects the file format.

| | |
|---|---|
| "HTML" | Saves the results as a webpage in hypertext markup language for display in a browser. |
| "CSV" | Saves the results as comma-separated values in a "`.csv`" file. |
| "XML" | Saves the results in a file compatible with an extended markup language. |

"Python" Saves the results in a Python-compatible ".py" file format.

Remote command:
SBUS<sb>:EXPResult:EXTension on page 1366

**Save as**
Opens a dialog box where you can select a filename and a path for the export results file.

Remote command:
SBUS<sb>:EXPResult:SAVE on page 1365
SBUS<sb>:EXPResult:PATH on page 1366

## 14.1.9 Performing protocol analysis

This section explains step by step how to configure and decode a protocol bus.

### 14.1.9.1 Configuring protocol signals

For configuration, assign the lines to the input channels, and define the active states and the logical thresholds.

1. Select the "+ Bus" activator in the bottom-right corner.




2. Select the bus tab that you want to configure, for example "SB1".
3. Select the "Protocol type" button. Select the protocol, for example "CAN".
4. Select "State" to enable the decoding.

   A CAN shortcut appears at the bottom-left corner. Select the shortcut for quick access to the respective protocol dialog settings.




5. Verify that all signals are visible on the screen. If a line is missing, adjust the vertical and/or horizontal scaling.
6. Configure the protocol specific details. They are described in the related sections.
7. Set the logical thresholds:
   a) Select the "Thresholds" tab.
   b) For each wire, set the threshold value.
   c) If necessary, set the hysteresis value for the wires.

d) Optionally, select "Show threshold lines".

### 14.1.9.2 Hardware triggering

A hardware trigger is available for the following protocols:

- CAN
- SPI
- I²C
- LIN
- UART

Prerequisites: A configured protocol signal. See Section 14.1.9.1, "Configuring protocol signals", on page 510.

1. Open "Menu" > "Trigger".
2. Set "Trigger on" to "Single event".
3. Ensure that the "Source" is set to the configured serial bus, e.g. "Serial bus 1".
4. Select the "Type".
5. If available, select "Set details" to define additional settings for more complex trigger types.

   The following figures illustrate examples for protocol-specific trigger configurations.

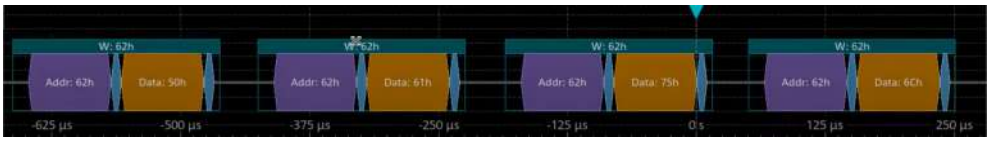



*Figure 14-7: I²C signal with trigger on Type=Data, Data=0x75 and Position=1*

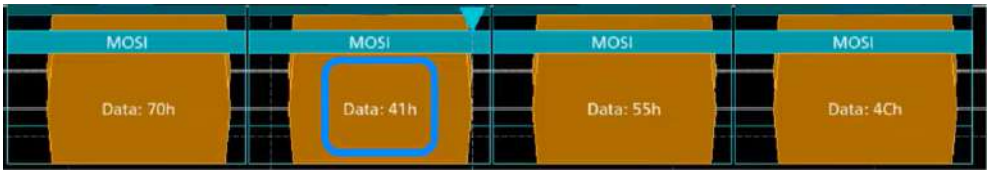



*Figure 14-8: SPI signal with trigger on Type= MOSI, Data pattern = 0x41.*

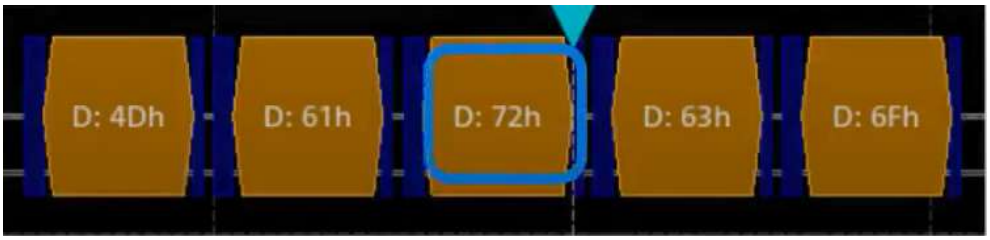



*Figure 14-9: UART signal with trigger on Type= Tx, Data =0x72 and Position=3*

#### 14.1.9.3 Decoding protocol signals

When the configuration of the serial bus is complete, the signal can be decoded:

1. Open the "Serial Bus" dialog for the respective bus.
2. In the "Setup" tab, enable "State".
3. In the "Display" tab, enable "Show decode table".

For a description of the display settings, see also Section 14.1.3.1, "Display settings", on page 492.

For a description of how to synchronize the displayed results, see Section 14.1.3.2, "Display synchronization", on page 494.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

#### 14.1.9.4 Exporting decode results

Once a decode has been completed, you can generate an export of all decode results.

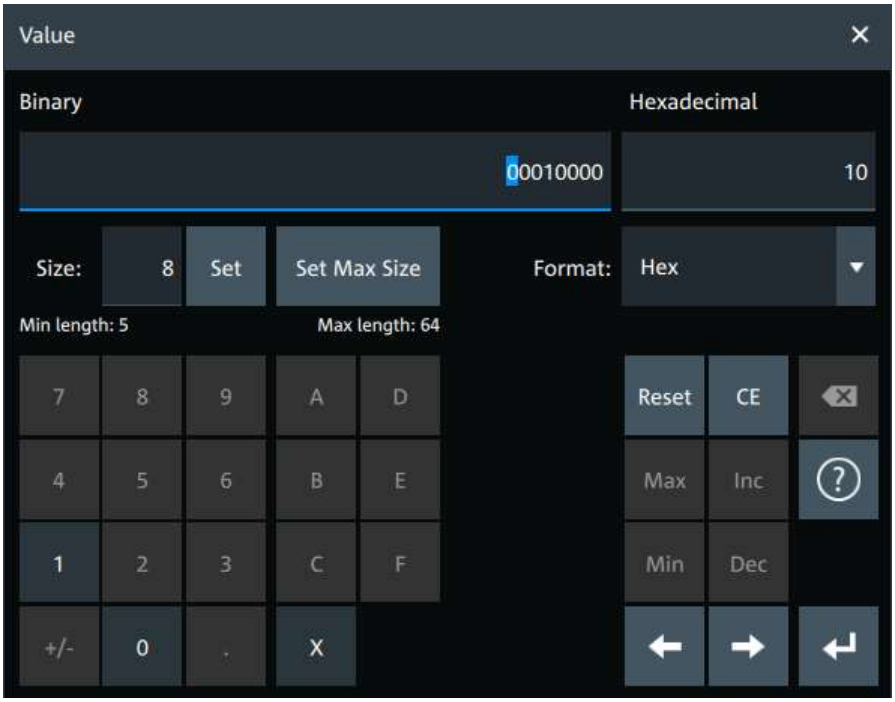
Prerequisites: A bus is configured for the 8b/10b protocol and decoded.

1. Select the protocol shortcut.
2. Select the "Export" tab.
3. If necessary, enable "Include details".
4. If necessary, enable "Include timing".
5. Select the "File type".
6. Select "Save as".
7. Enter a filename. Select "Save".

   An export file is saved in the "User data" directory.

### 14.1.10 Bit pattern editor

If you want to enter a specified address or data pattern, the pattern editor helps you to enter the pattern in various formats - decimal, hexadecimal, octal, binary and ASCII.




The editor at the top displays the pattern in two columns. The left column always shows binary data. For the right column, you can select the "Format". The default depends on the data specifics. You can edit data in the left or right column. The keypad adapts itself to the column format and only keys appropriate to the format are enabled.

The data is grouped and converted in bit groups. The size of a bit group depends on the address or data specifics and is set by the instrument. Groups are automatically separated by blanks. The maximum size of a bit group is 64 bit, the most common group size is 1 byte.

**Format-specific information**

- Binary: 0, 1 and X (do not care) is allowed.
- Octal: Each digit represents 3 bit.
- Hex: most common format in the right column.
- ASCII: In the ASCII column, "X" is the character X. The binary X (do not care) is not allowed. If an X is included in the binary value in the left column, the ASCII column displays "§" to indicate that the value is not defined.
- Unsigned: Decimal data format without a sign for positive or negative numbers. 3 digits are grouped, and the next group is separated by a comma, e.g. 653,848.
- Signed: Decimal format with a sign for positive or negative numbers. 3 digits are grouped, where the first bit represents the sign. You can use the 2's complement format. After a group of 3 digits the next group is separated by a comma, e.g. -653,848.

## 14.2 SPI bus (IC electronics, option R&S MXO3-K510)

The Serial Peripheral Interface SPI is used for communication with slow peripheral devices, in particular, for transmission of data streams.

For basic information on SPI, you can also refer to a video, available on the Rohde & Schwarz YouTube channel: Understanding SPI.

**Requirements**

For performing SPI decode measurements, you need the following:

- MXO 3 with 4 channels. The channels can be a combination of the following:
  - Analog channels (C1-C4)
  - Logic channels (D0-D15) (requires option R&S MXO3-B1)
- Option R&S MXO3-K510

- About the SPI protocol ..... 514
- SPI configuration ..... 515
- SPI filter ..... 519
- SPI format configuration ..... 521
- SPI hardware trigger ..... 527
- SPI software trigger ..... 528
- Measure ..... 531
- SPI decode results ..... 531

### 14.2.1 About the SPI protocol

A 4-channel instrument is required for full support of the SPI protocol, or the MSO option R&S MXO3-B1.

The Serial Peripheral Interface SPI is used for communication with slow peripheral devices, in particular, for transmission of data streams.

The main characteristics of SPI are:

- Main-subnode communication
- No device addressing. The sub is accessed by a chip select, or sub select line.
- No acknowledgment mechanism to confirm receipt of data
- Duplex capability

Most SPI buses have four lines, two data and two control lines:

- Clock line to all subs (SCLK)
- Sub select or Chip select line (SS or CS)
- Main data output, sub data input (MOSI or SDI)
- Main data input, sub data output (MISO or SDO)

When the main generates a clock and selects a sub device, data can be transferred in either or both directions simultaneously.

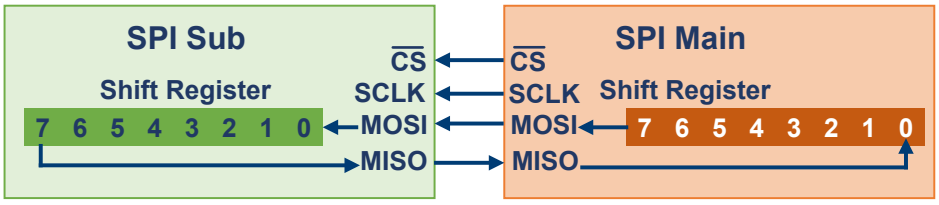



*Figure 14-10: Simple configuration of SPI bus*

The data bits of a message are grouped by the following criteria:

- A word contains several successive bits. The word length is defined in the protocol configuration.
- A frame contains several successive words, at least one word.

**Supported frame types**

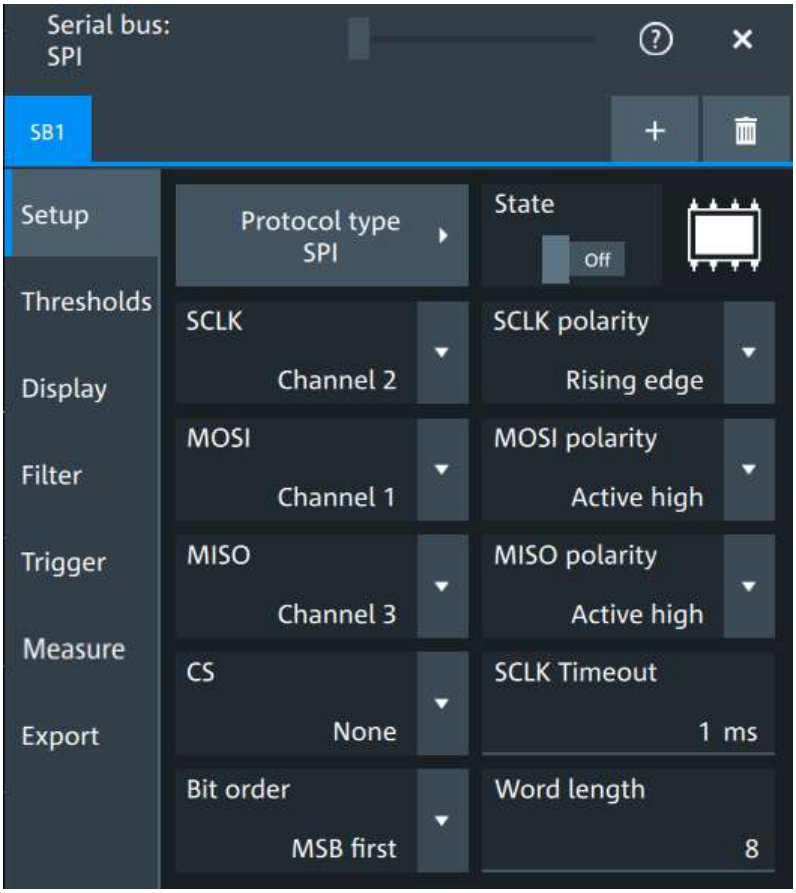
- MISO: Main data output, sub data input
- MOSI: Main data input, sub data output
- MISOMOSI: Combined view of both SPI data directions

## 14.2.2 SPI configuration

### 14.2.2.1 SPI configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "SPI" > "Setup".

Make sure that the tab of the correct serial bus is selected.




**Protocol type**
Selects the protocol type to be decoded.

Remote command:
`SBUS<sb>:TYPE` on page 1362

**State**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
`SBUS<sb>[:STATe]` on page 1362

**SCLK,MOSI,MISO,CS**
Sets the input channel for the respective line:

- SCLK: clock line
- MOSI line
- MISO line
- CS: chip select

Alternatively, digital channels can be used if MSO option R&S MXO3-B1 is installed. Digital and analog channels cannot be used at the same time.

For triggering on a serial bus, analog or digital channel sources are required.

Remote command:
SBUS<sb>:SPI:MISO:SOURce on page 1370
SBUS<sb>:SPI:MOSI:SOURce on page 1371
SBUS<sb>:SPI:SCLK:SOURce on page 1372
SBUS<sb>:SPI:CSELect:SOURce on page 1369

**Clock polarity**
Two settings define the clock mode: the clock polarity and the clock phase. Together, they determine the edges of the clock signal on which the data are driven and sampled.

A main/subnode pair must use the same parameter pair values to communicate. The clock phase defines the slope. It selects if data is stored with the rising or falling slope of the clock. The slope marks the begin of a new bit.

**Polarity: MOSI, MISO, CS**
Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

For triggering on a serial bus, analog or digital channel sources are required.

Remote command:
SBUS<sb>:SPI:MISO:POLarity on page 1370
SBUS<sb>:SPI:MOSI:POLarity on page 1371
SBUS<sb>:SPI:CSELect:POLarity on page 1369

**Bit order**
Selects the bit order, which determines if the data of the messages starts with MSB (most significant bit) or LSB (least significant bit).

Remote command:
SBUS<sb>:SPI:BORDer on page 1368

**Word length**
Sets the word length (or symbol size), which is the number of bits in a message. The maximum word length is 32 bit.

Remote command:
SBUS<sb>:SPI:WSIZe on page 1373

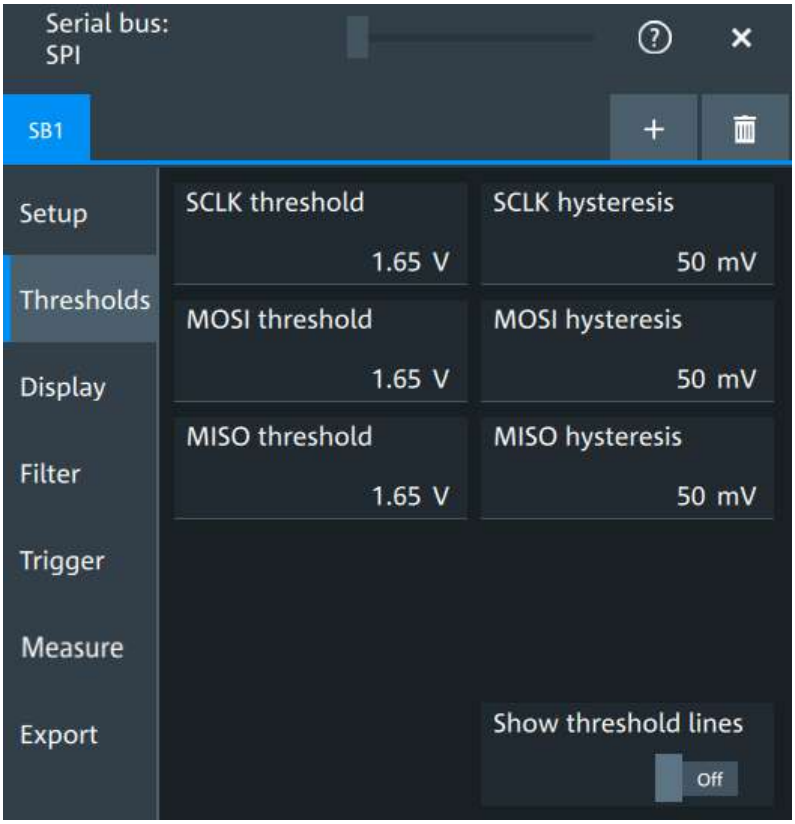
**Timeout**
Sets the minimum clock idle time if a timeout on the clock line SCLK is used as limiter between two frames.

This setting is only available, for "CS" = "None".

Remote command:
SBUS<sb>:SPI:TIMeout on page 1373

#### 14.2.2.2 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "SPI" > "Thresholds".




**Threshold**
Sets the threshold for the SCLK, MOSI and MISO channels. Enter the value directly in the fields.

Additional to the threshold, you can also set a hysteresis.

If the signal value on the line is higher than the threshold, the signal state is high. Otherwise, the signal state is considered low if the signal value is below the threshold.

The interpretation of high and low is defined by the "Polarity: MOSI, MISO, CS" on page 517.

Remote command:
`SBUS<sb>:SPI:MISO:HYSTeresis` on page 1369
`SBUS<sb>:SPI:MOSI:HYSTeresis` on page 1371
`SBUS<sb>:SPI:SCLK:HYSTeresis` on page 1372
`SBUS<sb>:SPI:CSELect:HYSTeresis` on page 1368
`SBUS<sb>:SPI:MISO:THReshold` on page 1370
`SBUS<sb>:SPI:MOSI:THReshold` on page 1371
`SBUS<sb>:SPI:SCLK:THReshold` on page 1372
`SBUS<sb>:SPI:CSELect:THReshold` on page 1369

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

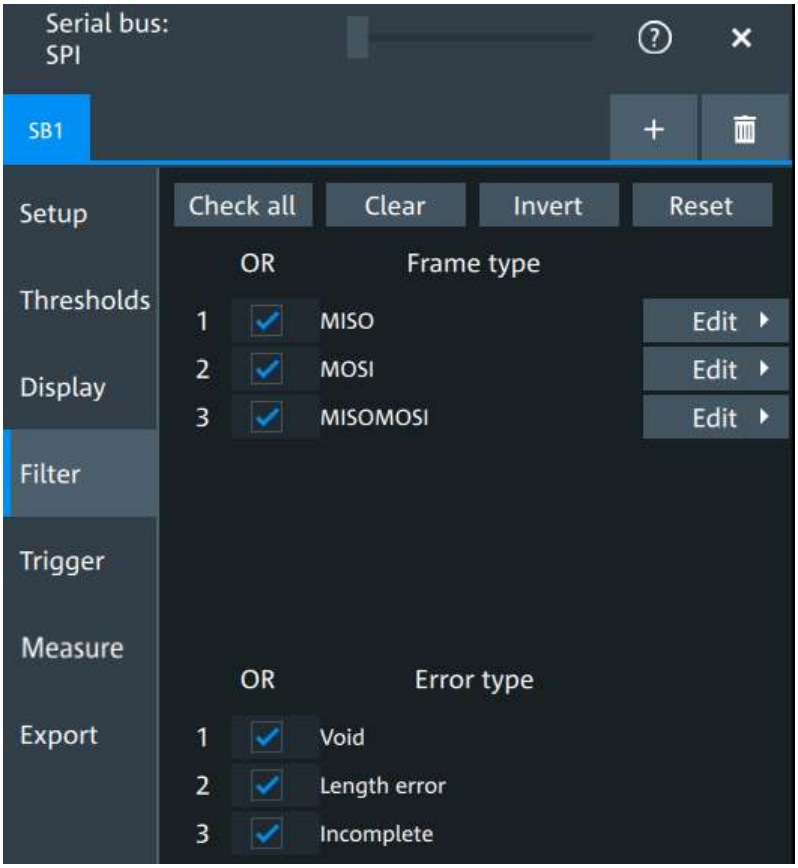
Remote command:
SBUS<sb>:THReshold on page 1364

#### 14.2.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

### 14.2.3 SPI filter

Access: "Menu" > "Apps" > "Protocol" tab > "SPI" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
SBUS<sb>:SPI:FILTer:CHKall on page 1375
SBUS<sb>:SPI:FILTer:CLR on page 1375
SBUS<sb>:SPI:FILTer:INVert on page 1375
SBUS<sb>:SPI:FILTer:RST on page 1376

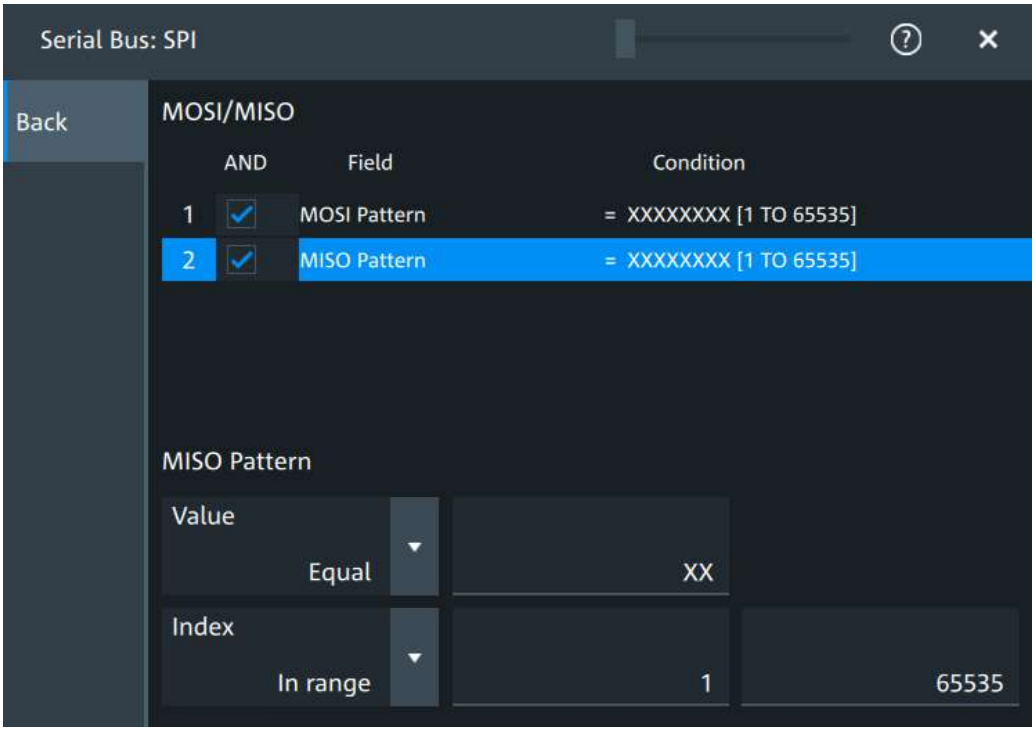
**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

The available frames are "MOSI", "MISO" and "MOSI/MISO".

Remote command:
SBUS<sb>:SPI:FILTer:FRAMe<fr>:ENABle on page 1378
SBUS<sb>:SPI:FILTer:FRENable on page 1378

**Edit**
Opens a dialog to define the details of the selected frame.




"Field" Enables the field type that you want to filter on for the selected frame. The available fields are "MOSI Pattern" and "MISO Pattern".

Remote command:
SBUS<sb>:SPI:FILTer:FIENable on page 1378
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:ENABle on page 1378

"Condition" Displays the value condition for the selected field.

Remote command:
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:BIT
on page 1376

"Data" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:DMAX
on page 1376
SBUS<sb>:SPI:FILTer:DMAX on page 1376
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:DMIN
on page 1377
SBUS<sb>:SPI:FILTer:DMIN on page 1377
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator
on page 1377
SBUS<sb>:SPI:FILTer:DOPerator on page 1377

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:IMAX
on page 1378
SBUS<sb>:SPI:FILTer:IMAX on page 1378
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:IMIN
on page 1379
SBUS<sb>:SPI:FILTer:IMIN on page 1379
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator
on page 1379
SBUS<sb>:SPI:FILTer:IOPerator on page 1379

**Edit**
Opens a dialog to define the details of the selected frame.

**Error type**
Enables filtering on the selected error type.

Available error types are "Void", "Length error" and "Incomplete".

Remote command:
SBUS<sb>:SPI:FILTer:ERRor<n>:ENABle on page 1377

### 14.2.4 SPI format configuration

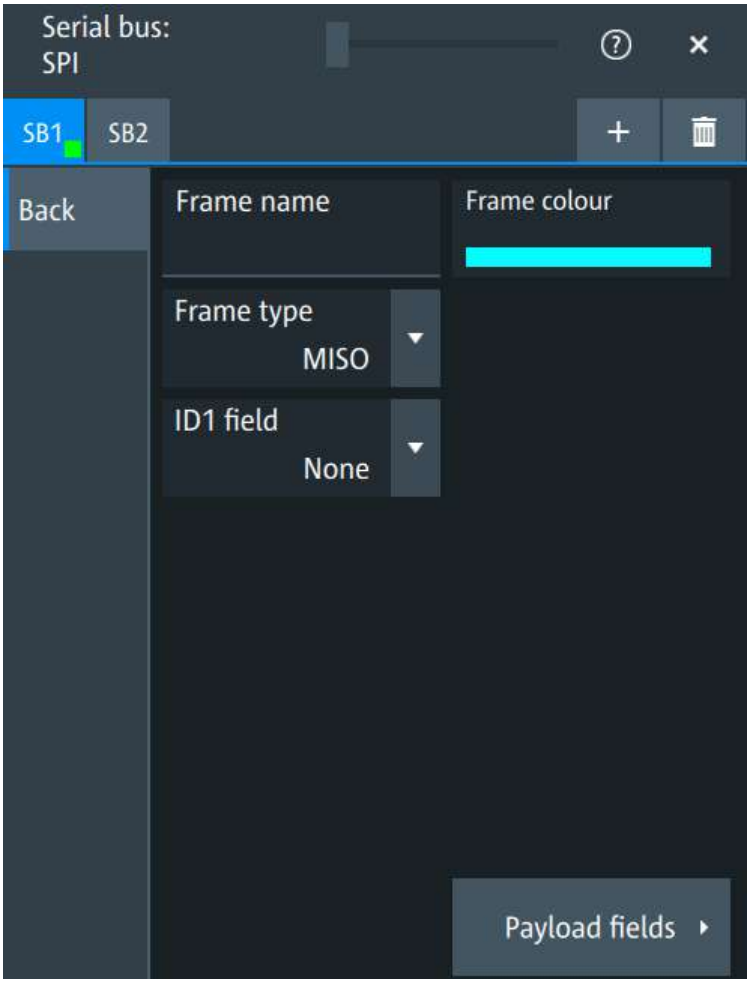
Access: "Menu" > "Apps" > "Protocol" tab > "SPI" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
SBUS<sb>:SPI:FORMat:SELect on page 1380

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:SPI:FORMat:ADDFrame on page 1381

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:SPI:FORMat:CLR on page 1381
SBUS<sb>:SPI:FORMat:FCOunt? on page 1382

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:SPI:FORMat:LOAD on page 1381

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:SPI:FORMat:SAVE on page 1381

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:SPI:FORMat:FRAMe<fr>:NAME on page 1382

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:SPI:FORMat:FRAMe<fr>:COLor on page 1383

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:SPI:FORMat:FRAMe<fr>:TYPE on page 1383

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID1Field on page 1383
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID2Field on page 1383
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID3Field on page 1383

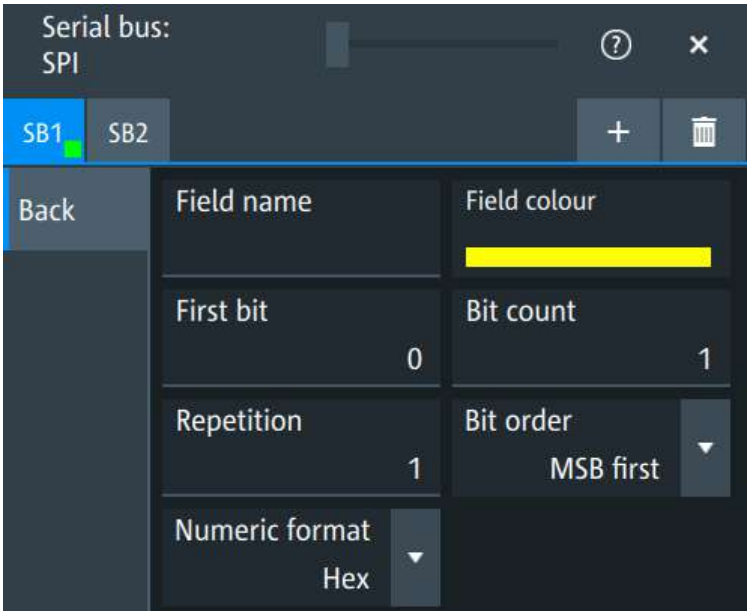
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID1Value on page 1384
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID2Value on page 1384
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID3Value on page 1384

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
`SBUS<sb>:SPI:FORMat:FRAMe<fr>:ADDField` on page 1382

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
`SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:NAME` on page 1384

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
`SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:COLor` on page 1384

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1385

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.
- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  - Bit count = 2: truncated. Result: =00
  - Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  - Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  - Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:BITCount on page 1385

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:REPetition on page 1385

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:BITorder on page 1386

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:

- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  – "=1HL111000": valid (read as binary)
  – "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  – "=1010": valid (read as HEX, with a total of 16 bits)
  – "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)
  – "=5X12": invalid
  – "=1H33": invalid

Remote command:
`SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FORMat` on page 1386

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
`SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ADDValue` on page 1387

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
`SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue` on page 1387

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
`SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel` on page 1388

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
`SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:UNIT` on page 1388

**Parameter A,B**
Defines the parameters of the linear function.

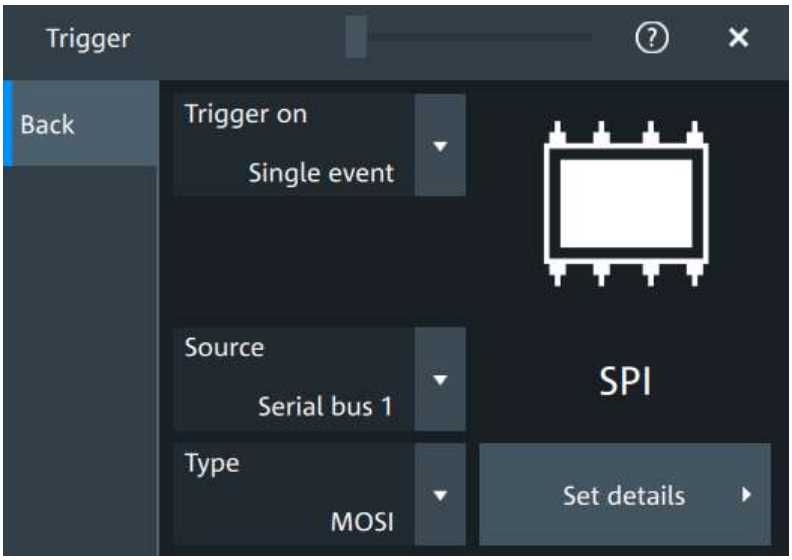
- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1388
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1388

### 14.2.5 SPI hardware trigger

Access: "Menu" > "Apps" > "Protocol" tab > "SPI" > "Trigger" tab > "Setup Hardware Trigger"



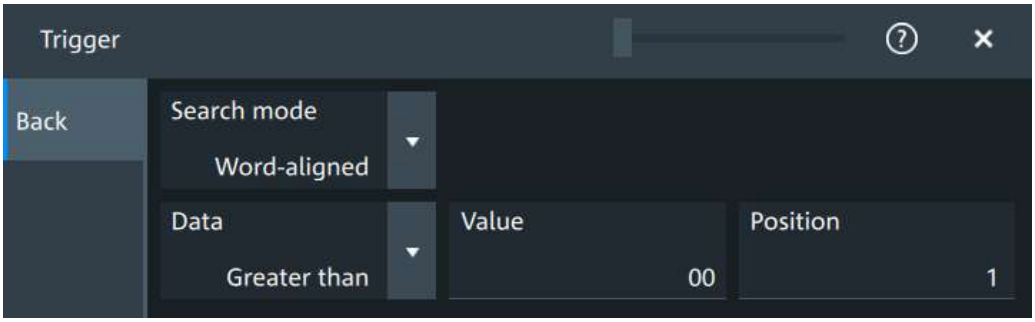

**Type**
Selects the trigger type for SPI analysis.

Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.

| | |
|---|---|
| "Start of frame" | Sets the trigger to the beginning of the frame. |
| "End of frame" | Sets the trigger to the end of the frame. |
| "MOSI" | Sets the trigger to a specified data pattern expected on the MOSI line.<br>Available, if there is a channel assigned to the "MOSI" line in the "Setup" tab. |
| "MISO" | Sets the trigger to a specified data pattern expected on the MISO line.<br>Available, if there is a channel assigned to the "MISO" line in the "Setup" tab. |

Remote command:
TRIGger:SBHW:SPI:TYPE on page 1389

**Data conditions**
The trigger on MOSI and MISO patterns is defined in the same way:




**Search mode ← Data conditions**
Defines how the specified data pattern is searched.

"Bit-aligned" Bit-by-bit: the pattern can start at any position in the message.

"Word-aligned" The pattern is matched only at word boundaries.

Remote command:
`TRIGger:SBHW:SPI:PALignment` on page 1390

**Data ← Data conditions**
Selects the operator condition for the data pattern and sets the data pattern.

Remote command:
`TRIGger:SBHW:SPI:FCONdition` on page 1390
`TRIGger:SBHW:SPI:DMINpattern` on page 1390

**Position ← Data conditions**
Sets the number of bits or words to be ignored before the first bit or word of interest.

Remote command:
`TRIGger:SBHW:SPI:DPOSition` on page 1390

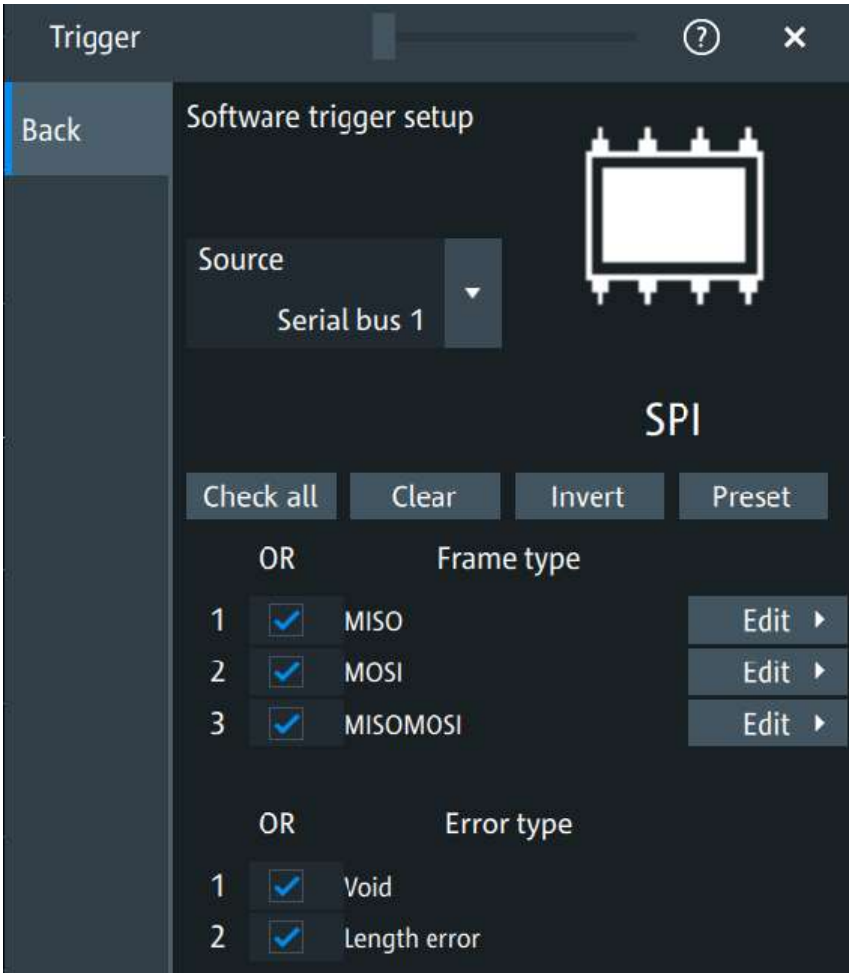
## 14.2.6 SPI software trigger

### 14.2.6.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.2.6.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "SPI" > "Trigger" tab > "Setup Software Trigger" > "Setup SB Trigger".




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:SPI:CHKall` on page 1391
`TRIGger:SBSW:SPI:CLR` on page 1392
`TRIGger:SBSW:SPI:INVert` on page 1392
`TRIGger:SBSW:SPI:RST` on page 1392

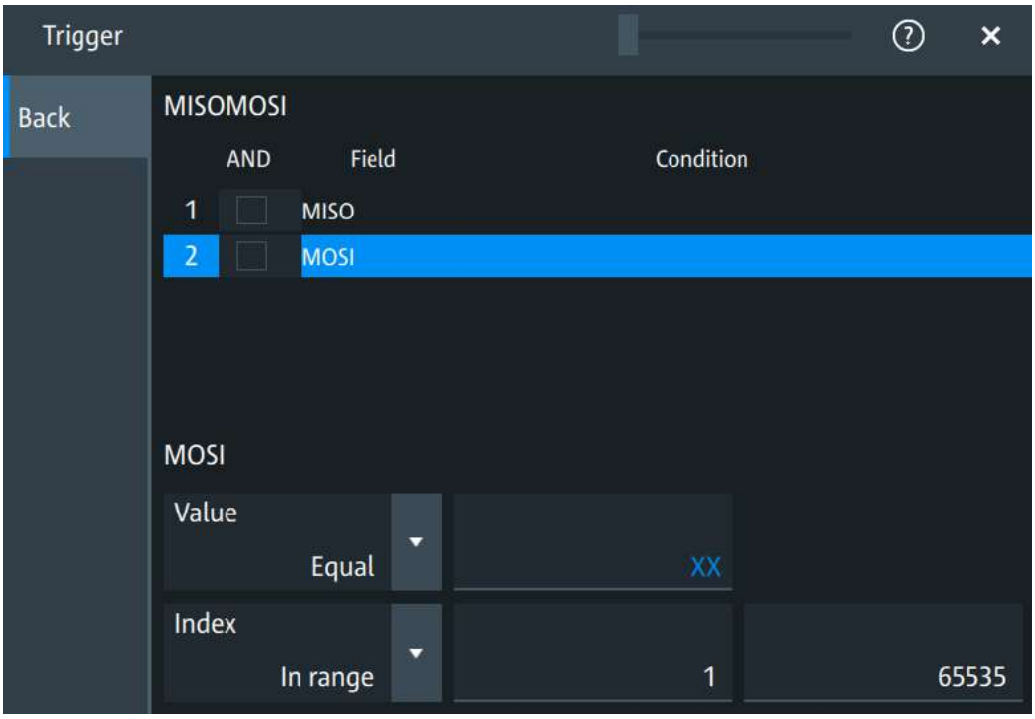
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: "MISO", "MOSI" and "MISO/MOSI".

Remote command:
`TRIGger:SBSW:SPI:FRENable` on page 1392
`TRIGger:SBSW:SPI:FRAMe<fr>:ENABle` on page 1392

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.




"Field" Enables the field type that you want to trigger on for the selected frame.
The following fields are available: "MOSI" and "MISO".

Remote command:
`TRIGger:SBSW:SPI:FIENable` on page 1394
`TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:ENABle` on page 1394

"Condition" Displays the value condition for the selected field.

Remote command:
`TRIGger:SBSW:SPI:BIT` on page 1392
`TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:BIT` on page 1392

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
TRIGger:SBSW:SPI:DMAX on page 1393
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DMAX on page 1393
TRIGger:SBSW:SPI:DMIN on page 1393
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DMIN on page 1393
TRIGger:SBSW:SPI:DOPerator on page 1394
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DOPerator on page 1394

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:SPI:IMAX on page 1394
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IMAX on page 1394
TRIGger:SBSW:SPI:IMIN on page 1395
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IMIN on page 1395
TRIGger:SBSW:SPI:IOPerator on page 1395
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IOPerator on page 1395

**Error type**
Enables triggering on the selected error type.

The following error types are available: "Void" and "Length error".

Remote command:
TRIGger:SBSW:SPI:ERENable on page 1395
TRIGger:SBSW:SPI:ERRor<m>:ENABle on page 1395

## 14.2.7 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

## 14.2.8 SPI decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

For more information on decoding SPI, you can also refer to the video, available on the Rohde & Schwarz YouTube channel: Decoding SPI with MXO Series Oscilloscopes.

**Decode results table**

The "Decode results" table contains information about all decoded frames.

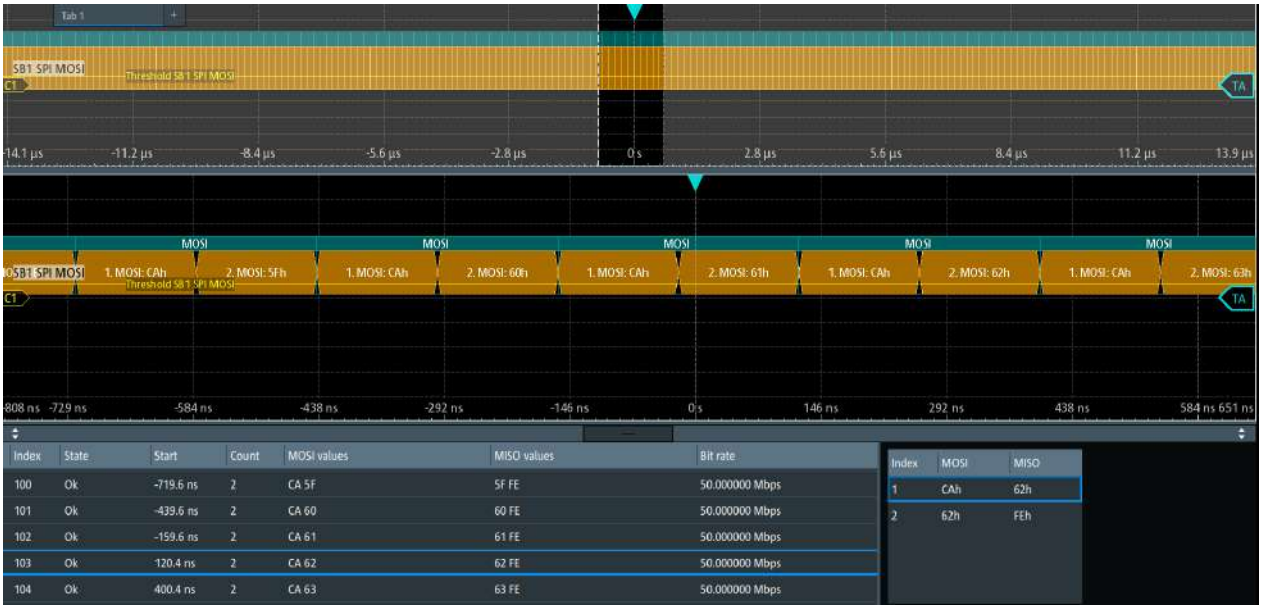



*Figure 14-11: Decoded SPI signal*

The decode results table contains information about all decoded frames.

*Table 14-1: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Index of the decoded frame |
| State | Overall state of the frame |
| Start | Times of frame start |
| Count | Number of words in the frame |
| MOSI values | Value of the MOSI data words. Select the data format in the "Display" tab. |
| MISO values | Value of the MISO data words. Select the data format in the "Display" tab. |
| Bit rate | Value of the bit rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

*Table 14-2: Content of the frame details table*

| Column | Description |
|---|---|
| Index | Index of the field |
| MOSI | Values of the MOSI fields |
| MISO | Values of the MISO fields |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- SBUS<sb>:FORMat on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.2.6, "Decode results", on page 1396.

To query the MOSI and MISO values, use the remote command SBUS<sb>:SPI:FRAMe<fr>:DATA?.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example SPI export file

```
Index,Start,Stop,State,Count,BitRate
1,-2.9e-05,-2.872e-05,'OK',2,50000000
2,-2.844e-05,-2.83e-05,'OK',1,50000000
3,-2.816e-05,-2.788e-05,'OK',2,50000000
4,-2.76e-05,-2.746e-05,'VOID',0,---

Details frame 1
Index,Start Time,Stop Time,MOSI Values,MISO Values
1,-2.9e-05,-2.886e-05,AFh,---
2,-2.886e-05,-2.872e-05,FEh,---

Details frame 2
Index,Start Time,Stop Time,MOSI Values,MISO Values
1,-2.844e-05,-2.83e-05,---,34h

Details frame 3
Index,Start Time,Stop Time,MOSI Values,MISO Values
1,-2.816e-05,-2.802e-05,CAh,F0h
2,-2.802e-05,-2.788e-05,05h,0Dh
```

## 14.3 QUAD-SPI bus (IC electronics, option R&S MXO3-K510)

QUAD-SPI also known as QSPI is a protocol that extends the standard SPI by using four data lanes instead of one. This protocol enables a faster data transfer.

### Requirements

For performing QUAD-SPI decode measurements, you need the following:

- MXO 3 with 4 channels. The channels can be a combination of the following:
  - Analog channels (C1-C4)
  - Logic channels (D0-D15) (requires option R&S MXO3-B1)

- Option R&S MXO3-K510

- About the QUAD-SPI protocol....534
- QUAD-SPI configuration....534
- QUAD SPI filter....537
- QUAD-SPI format configuration....540
- QUAD SPI software trigger....544
- Measure....547
- QUAD-SPI decode results....547

## 14.3.1 About the QUAD-SPI protocol

Quad Serial Peripheral Interface is an extension of the standard Serial Peripheral Interface (SPI). It enhances the data transfer capabilities by utilizing four data lines instead of the single data line used in traditional SPI. This expansion significantly increases the data throughput, making it ideal for applications requiring rapid data exchange, such as memory devices, display controllers, and sensors.

**Key features**

Key features of QSPI are:

- **Increased data throughput**: By using four data lines (IO0, IO1, IO2, IO3), QSPI can achieve higher data transfer rates compared to standard SPI. This makes it suitable for high-speed applications.
- **Enhanced flexibility**: QSPI supports various modes of operation, including single, dual, and quad modes, allowing for flexible communication tailored to specific needs.
- **Memory mapping**: one of the standout features of QSPI is its ability to map external flash memory directly into the memory space of a microcontroller, enabling efficient and seamless data access.

**Communication Lines**

QSPI uses 6 primary lines:

- SCLK (serial clock): generates the clock signal
- CS (chip select): selects the device for communication
- IO0-IO3 (data lines): four lines for data transfer.

## 14.3.2 QUAD-SPI configuration

### 14.3.2.1 QUAD-SPI configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "QUAD-SPI" > "Setup".

Make sure that the tab of the correct serial bus is selected.

**Protocol type**
Selects the protocol type to be decoded.

Remote command:
SBUS<sb>:TYPE on page 1362

**State**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
SBUS<sb>[:STATe] on page 1362

**CS, SCLK, IO n**
Sets the input channel for the respective line:

- CS: chip select
- SCLK: clock line
- IO0, IO1, IO2, IO3, IO4: data lines

Remote command:
SBUS<sb>:QSPI:CSEL:SOURce on page 1403
SBUS<sb>:QSPI:IOONe:SOURce on page 1404
SBUS<sb>:QSPI:IOTHree:SOURce on page 1406
SBUS<sb>:QSPI:IOTWo:SOURce on page 1408
SBUS<sb>:QSPI:IOZero:SOURce on page 1409
SBUS<sb>:QSPI:SCLK:SOURce on page 1410

**SCLK SDR Polarity**
Selects if the transmitted signal for the respective line is rising or falling.

Remote command:
SBUS<sb>:QSPI:SCLK:POLarity on page 1410

**Polarity: CS, IO<n>**
Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

Remote command:
SBUS<sb>:QSPI:CSEL:POLarity on page 1402
SBUS<sb>:QSPI:IOONe:POLarity on page 1404
SBUS<sb>:QSPI:IOTHree:POLarity on page 1405
SBUS<sb>:QSPI:IOTWo:POLarity on page 1407
SBUS<sb>:QSPI:IOZero:POLarity on page 1408

**Instruction**
Selects the instruction mode that defines how many lanes are used to transmit data.

"Single" Only one data line is used, IO0.

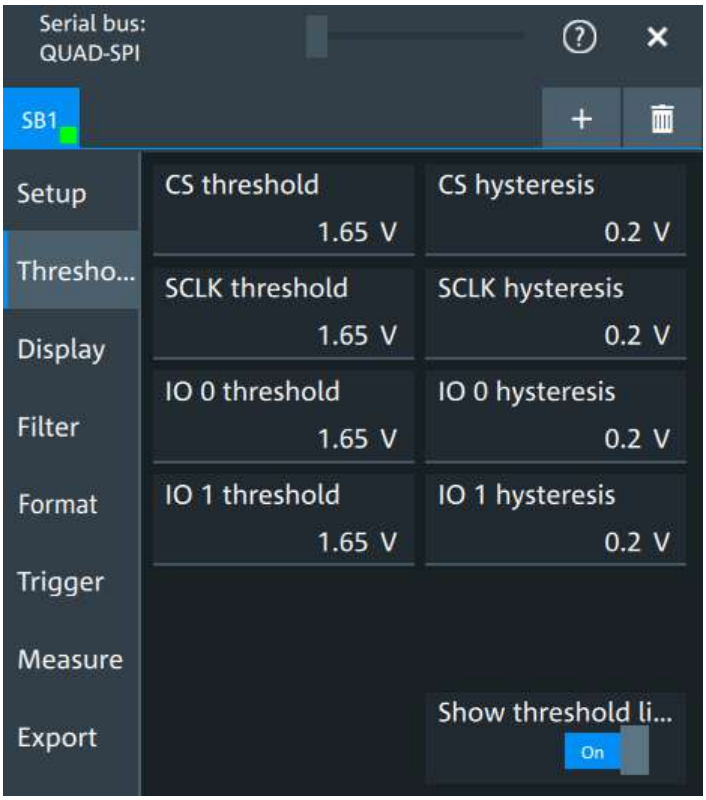
"Dual" Two lines are used for data transmission IO0 and IO1. This mode doubles the data rate compared to a single mode, as it can send or receive 2 bits per clock cycle.

"Quad" Four lines are used for data transmission IO0, IO1, IO2 and IO3. This mode provides the highest data transfer rate. This mode quadruples the data transfer rate compared to a single mode, as it can send or receive 4 bits per clock cycle.

Remote command:
`SBUS<sb>:QSPI:INSTruction` on page 1403

### 14.3.2.2 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "QUAD-SPI" > "Thresholds".




**CS/SCLK/IO<n> threshold**
Set the threshold for each signal channel. Enter the value directly in the fields.

Remote command:
`SBUS<sb>:QSPI:CSEL:THReshold` on page 1403
`SBUS<sb>:QSPI:IOONe:THReshold` on page 1405
`SBUS<sb>:QSPI:IOTWo:THReshold` on page 1408
`SBUS<sb>:QSPI:IOTHree:THReshold` on page 1406
`SBUS<sb>:QSPI:IOZero:THReshold` on page 1409
`SBUS<sb>:QSPI:SCLK:THReshold` on page 1411

**CS/SCLK/IO<n> hysteresis**
Set the hysteresis for each signal channel. Enter the value directly in the fields.

Remote command:
`SBUS<sb>:QSPI:CSEL:HYSTeresis` on page 1402
`SBUS<sb>:QSPI:IOONe:HYSTeresis` on page 1403
`SBUS<sb>:QSPI:IOTWo:HYSTeresis` on page 1407
`SBUS<sb>:QSPI:IOTHree:HYSTeresis` on page 1405
`SBUS<sb>:QSPI:IOZero:HYSTeresis` on page 1408
`SBUS<sb>:QSPI:SCLK:HYSTeresis` on page 1410

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
`SBUS<sb>:THReshold` on page 1364

#### 14.3.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

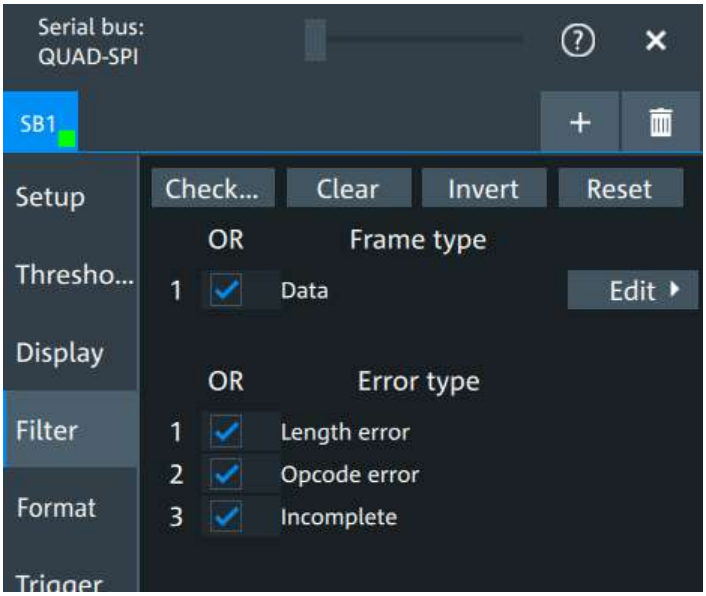
**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

| | |
|---|---|
| "Off" | Disables the display of the decode layer. |
| "Edges" | Enables the display of all edges for each lane. |
| "Bits" | Enables the display of all bits for each lane. |
| "Merged bits" | Enables the display of the bits merged across multiple lanes. |
| "Words" | Enables the display of all words. |

### 14.3.3 QUAD SPI filter

Access: "Menu" > "Apps" > "Protocol" tab > "QUAD-SPI" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
`SBUS<sb>:QSPI:FILTer:CHKall` on page 1412
`SBUS<sb>:QSPI:FILTer:CLR` on page 1412
`SBUS<sb>:QSPI:FILTer:INVert` on page 1412
`SBUS<sb>:QSPI:FILTer:RST` on page 1412

**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

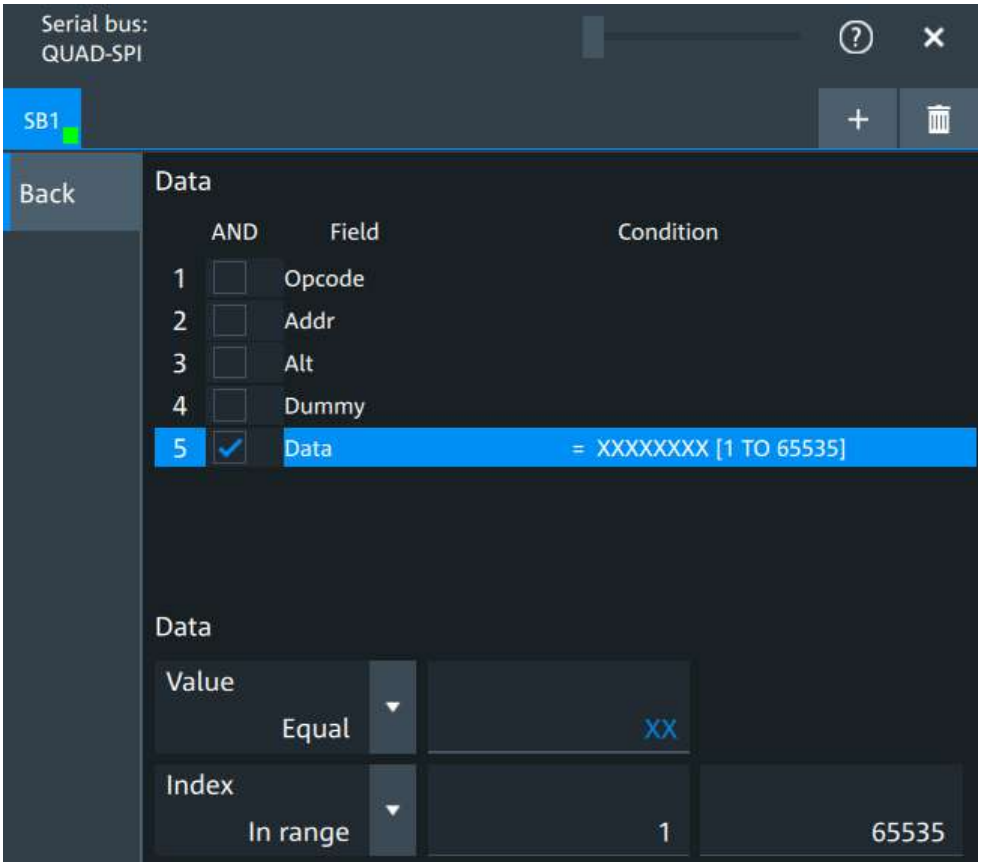
For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

The available frame types are the opcodes and an "Undescribed", to support custom coding.

If you have created own opcodes, they also appear in this list.

See Section 14.1.5, "Format", on page 497.

**Edit**
Opens a dialog to define the details of the selected frame.




"Field" Enables the field type that you want to filter on for the selected frame.

Depending on the frame type, the available fields are:

- "Opcode"
- "Addr"
- "Alt"
- "Dummy"
- "Data"

Remote command:
`SBUS<sb>:QSPI:FILTer:FIENable` on page 1415
`SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:ENABle` on page 1415

"Condition" Displays the value condition for the selected field.

Remote command:
`SBUS<sb>:QSPI:FILTer:BIT` on page 1412
`SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:BIT` on page 1412

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:QSPI:FILTer:DMAX on page 1413
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1413
SBUS<sb>:QSPI:FILTer:DMIN on page 1413
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1413
SBUS<sb>:QSPI:FILTer:DOPerator on page 1414
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1414

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:QSPI:FILTer:IMAX on page 1414
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1414
SBUS<sb>:QSPI:FILTer:IMIN on page 1414
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1414
SBUS<sb>:QSPI:FILTer:IOPerator on page 1415
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1415

**Error type**
Enables filtering on the selected error type.

Available error types are "Length error", "Opcode error", "Incomplete".

Remote command:
SBUS<sb>:QSPI:FILTer:ERENable on page 1415
SBUS<sb>:QSPI:FILTer:ERRor<n>:ENABle on page 1415

### 14.3.4 QUAD-SPI format configuration

Access: "Menu" > "Apps" > "Protocol" tab > "QUAD-SPI" > "Format".

See Section 14.1.5, "Format", on page 497.

**Add**
Adds a new item to the opcode list.

Remote command:
SBUS<sb>:QSPI:OPCode:APPend on page 1417

**Delete**
Deletes the currently selected opcode from the list.

Remote command:
SBUS<sb>:QSPI:OPCode:DELete on page 1417

**Delete all**
Deletes all defined opcodes from the list.

Remote command:
SBUS<sb>:QSPI:OPCode:DALL on page 1417

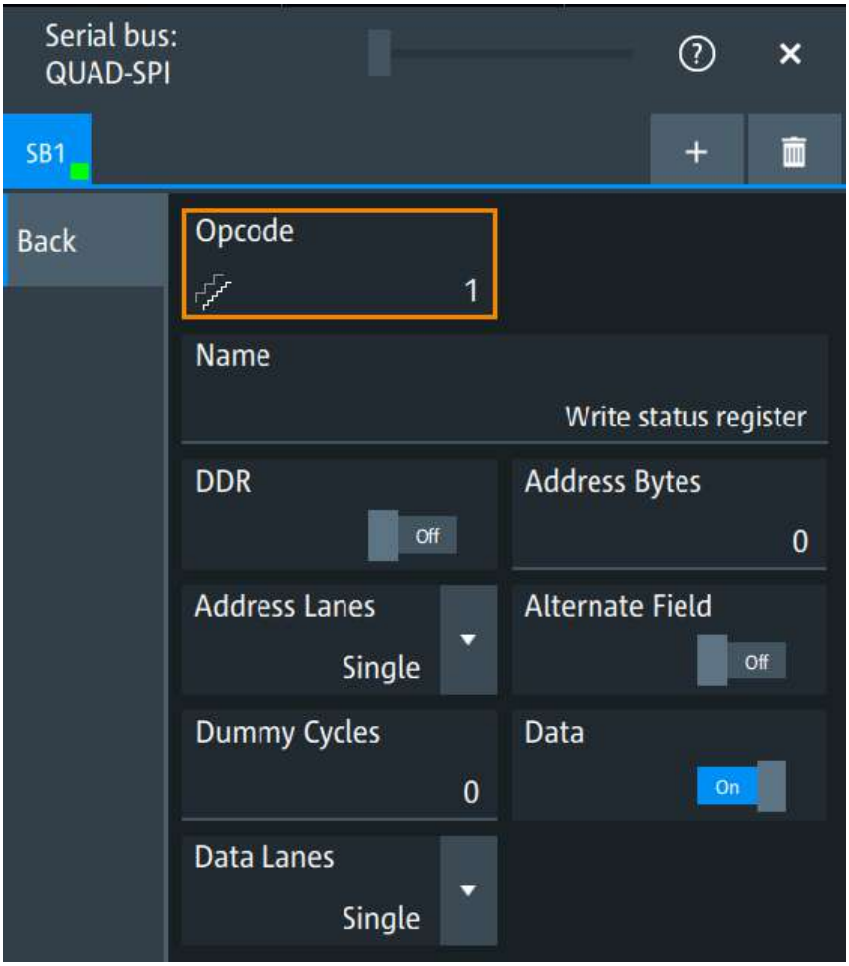
**Open, Save as**
Opens a dialog box where you can select a filename and a path for the export results file.

Remote command:
SBUS<sb>:QSPI:LDOPcode on page 1417
SBUS<sb>:QSPI:SVOP on page 1421

**Preset**
Resets the opcode fields to the predefined values.

Remote command:
SBUS<sb>:QSPI:OPCode:RESet on page 1417

**Opcode settings**




You can define the following settings for each opcode separately:

**Opcode ← Opcode settings**
Sets the opcode value.

Remote command:
`SBUS<sb>:QSPI:OPCode:ITEM<n>:CODE` on page 1419

**Name ← Opcode settings**
Sets the name for the opcode.

Remote command:
`SBUS<sb>:QSPI:OPCode:ITEM<n>:NAME` on page 1421

**DDR ← Opcode settings**
If enabled, the data is transferred on both the rising and falling edges of the clock signal. This method effectively doubles the data transfer rate compared to a single data rate (SDR), which transfers data only on one clock edge.

Remote command:
`SBUS<sb>:QSPI:OPCode:ITEM<n>:DDR` on page 1420

**Address Bytes ← Opcode settings**
Sets the address byte. It specifies the location in the flash memory where the operation (e.g., read, write) is performed.

Remote command:
SBUS<sb>:QSPI:OPCode:ITEM<n>:ADBYtes on page 1418

**Address Lanes ← Opcode settings**
Selects how many lines or lanes are used to send the address bytes to the flash memory.

"Single" Address is sent over one serial line.

"Dual" Address is sent over two lines.

"Quad" Address is sent over four lines.

Remote command:
SBUS<sb>:QSPI:OPCode:ITEM<n>:ADLanes on page 1418

**Alternative Field ← Opcode settings**
Enable, if an "Alt" field is available. The alternative field can be implemented according to the preferences of the manufacturer of the device under test, for example a flash memory device.

Remote command:
SBUS<sb>:QSPI:OPCode:ITEM<n>:ALT on page 1419

**Dummy Cycles ← Opcode settings**
Sets the number of dummy cycles. Dummy cycles are clock cycles inserted after the address or other command sequences but before data transfer begins. These cycles allow the flash memory device additional time to perform internal operations or latch onto the correct data to ensure accurate read or write operations.

Remote command:
SBUS<sb>:QSPI:OPCode:ITEM<n>:DMCYcles on page 1420

**Data ← Opcode settings**
Enable, if data is being transferred in the opcode operation.

Remote command:
SBUS<sb>:QSPI:OPCode:ITEM<n>:DATA on page 1419

**Data Lanes ← Opcode settings**
Selects how many lanes are used for the data transfer.

The data lanes refer to the physical connections through which data is transmitted between the main (typically a microcontroller) and the secondary (typically a flash memory device).

"Single" Data is sent over one serial line.

"Dual" Data is sent over two lines.

"Quad" Data is sent over four lines.

Remote command:
SBUS<sb>:QSPI:OPCode:ITEM<n>:DTLanes on page 1420

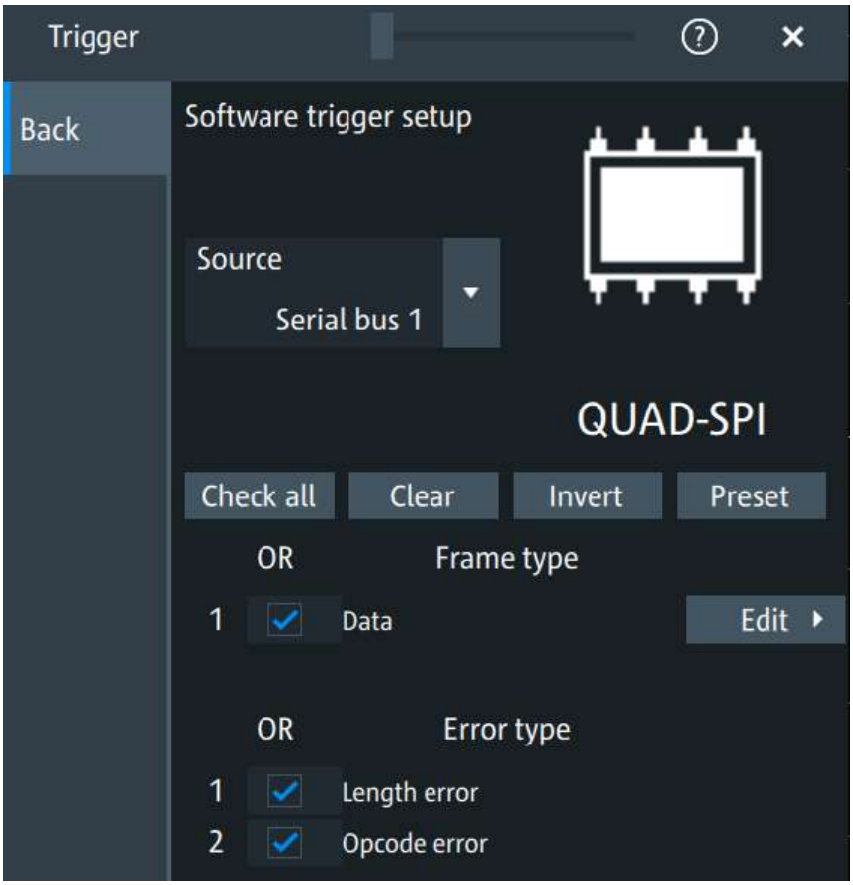
## 14.3.5 QUAD SPI software trigger

### 14.3.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.3.5.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "QUAD-SPI" > "Trigger" tab > "Setup Software Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:QSPI:CHKall` on page 1422
`TRIGger:SBSW:QSPI:CLR` on page 1422
`TRIGger:SBSW:QSPI:INVert` on page 1422
`TRIGger:SBSW:QSPI:RST` on page 1423

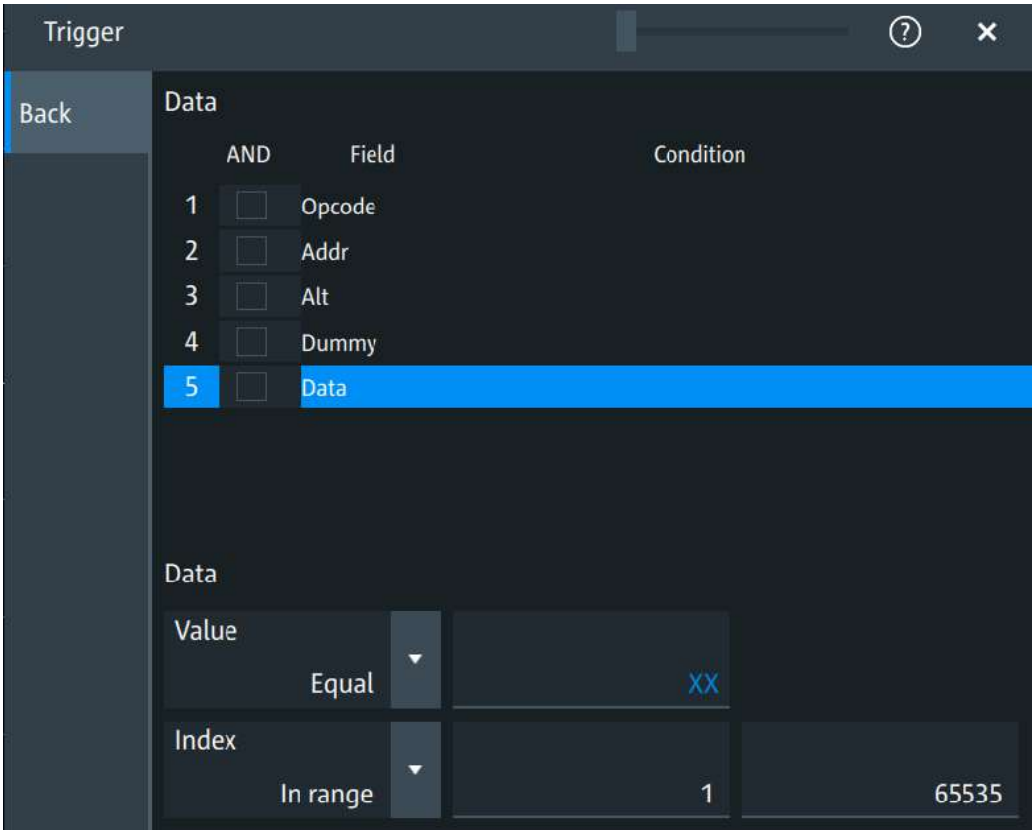
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The available frame is "Data".

Remote command:
`TRIGger:SBSW:QSPI:FRENable` on page 1423
`TRIGger:SBSW:QSPI:FRAMe<fr>:ENABle` on page 1423

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.




"Field" Enables the field type that you want to trigger on for the selected frame.
The following fields are available: "Opcode", "Addr", "Alt", "Dummy" and "Data".

Remote command:
`TRIGger:SBSW:QSPI:FIENable` on page 1425
`TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:ENABle` on page 1425

"Condition" Displays the value condition for the selected field.

Remote command:
`TRIGger:SBSW:QSPI:BIT` on page 1423
`TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:BIT` on page 1423

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`TRIGger:SBSW:QSPI:DMAX` on page 1424
`TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DMAX` on page 1424
`TRIGger:SBSW:QSPI:DMIN` on page 1424
`TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DMIN` on page 1424
`TRIGger:SBSW:QSPI:DOPerator` on page 1424
`TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DOPerator` on page 1424

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:QSPI:IMAX on page 1425
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IMAX on page 1425
TRIGger:SBSW:QSPI:IMIN on page 1426
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IMIN on page 1426
TRIGger:SBSW:QSPI:IOPerator on page 1426
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IOPerator on page 1426

**Error type**
Enables triggering on the selected error type.

The following error types are available: "Length error" and "Opcode error".

Remote command:
TRIGger:SBSW:QSPI:ERENable on page 1426
TRIGger:SBSW:QSPI:ERRor<m>:ENABle on page 1426

### 14.3.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.3.7 QUAD-SPI decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

**Decode results table**

The "Decode results" table contains information about all decoded frames.

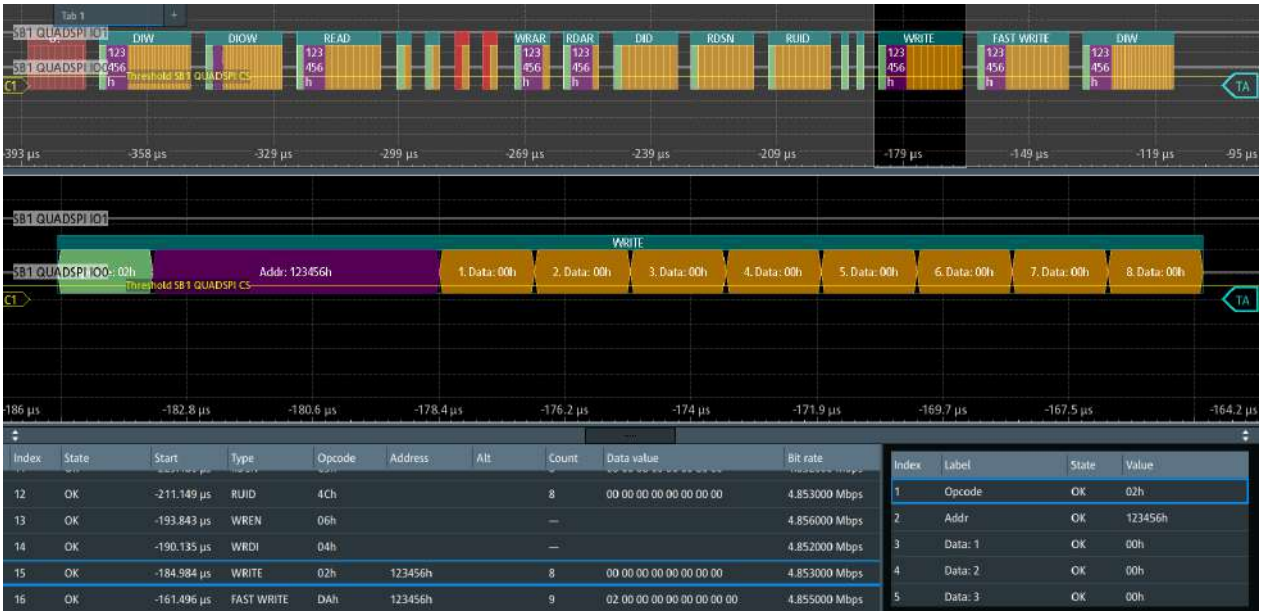



***Figure 14-12: Decoded QUAD-SPI signal***

The decode results table contains information about all decoded frames.

***Table 14-3: Content of the decode results table***

| Column | Description |
|---|---|
| Index | Index of the decoded frame |
| State | Overall state of the frame |
| Start | Times of frame start |
| Opcode | Operation code value |
| Address | Address value |
| Alt | Value of the alternate byte |
| Count | Number of words in the frame |
| Data value | Value of the data frame. Select the data format in the "Display" tab. |
| Bit rate | Value of the bit rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

***Table 14-4: Content of the frame details table***

| Column | Description |
|---|---|
| Index | Number of the decoded field |
| Label | Name of the field |
| State | State of the field |
| Value | Value of the field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.3.5, "Decode results", on page 1427.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example QUAD-SPI export file

```
Index,Start,Stop,State,Opcode,Address,Alt,Count,Bit rate
1,-7.04e-05,-5.28e-05,'OK',0Ah,12345678h,0Ch,4,5000000
2,-5.28e-05,-3.52e-05,'OK',0Ah,12345678h,0Ch,4,5000000
3,-3.52e-05,-1.76e-05,'OK',0Ah,12345678h,0Ch,4,5000000
4,-1.76e-05,0,'OK',0Ah,12345678h,0Ch,4,5000000

Details frame 1
Index,Data
1,52h
2,53h
3,54h
4,55h

Details frame 2
Index,Data
1,56h
2,57h
3,58h
4,59h
```

## 14.4 OCTO-SPI bus (IC electronics, option R&S MXO3-K510)

### Requirements

For performing OCTO-SPI decode measurements, you need the following:

- MXO 3 with 8 channels. The channels can be a combination of the following:
  - Analog channels (C1-C8)
  - Logic channels (D0-D15) (requires option R&S MXO3-B1)

- Option R&S MXO3-K510

- About the OCTO-SPI protocol....550
- OCTO-SPI configuration....551
- OCTO SPI filter....555
- OCTO-SPI format configuration....557
- OCTO SPI software trigger....561
- Measure....563
- OCTO-SPI decode results....564

## 14.4.1 About the OCTO-SPI protocol

Octal Serial Peripheral Interface (Octo-SPI or OSPI) is an extension of the classic Serial Peripheral Interface (SPI) protocol that employs eight bidirectional data lines (IO0-IO7) instead of the single MOSI/MISO pair used in conventional SPI. By transferring up to 8 bits per clock edge or 16 bits when dual-data-rate (DDR) is enabled, OSPI can deliver throughput that is roughly twice that of QUAD-SPI while still using the same basic command-address-data transaction format.

### Key features

Key features of OSPI are:

- **Very high data throughput**: Up to several gigabits per second, depending on clock frequency (often 100 to 200 MHz) and DDR operation.
- **Enhanced Flexibility**: Devices can be addressed in single-, dual-, quad-, or octal-mode, allowing to start with a simple 1-bit interface and later switch to 8-bit for maximum speed.
- **Memory Mapping**: external OSPI flash can be mapped directly into the microcontroller's address space, enabling execute-in-place (XIP) code and fast random reads without CPU-managed byte-wise transfers.
- **Extended address space**: OSPI controllers typically support 24-, 32-, or even 40-bit address fields, making it suitable for very large flash devices.
- **Optional dual-data-rate (DDR)**: Sampling on both clock edges doubles the effective bit rate without increasing the pin count.

### Communication Lines

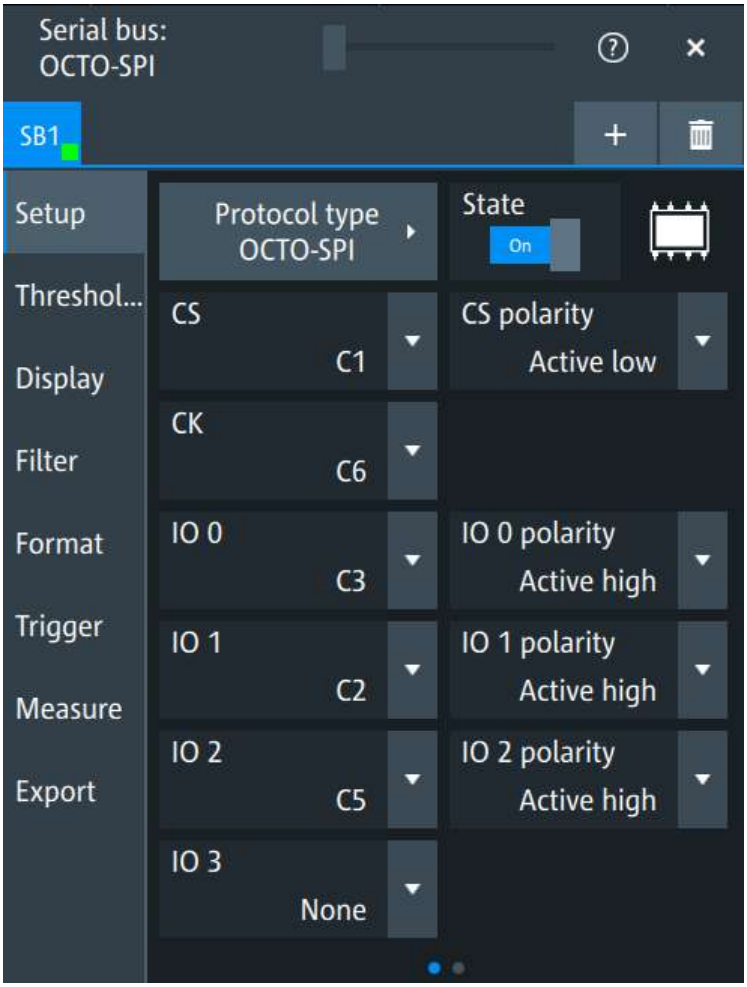
OSPI uses up to 11 primary lines:

- CK (clock line): Generates the clock signal.
- CS (chip select line): Selects the device for communication.
- IO0-IO7 (data lines): Eight bidirectional I/O pins that carry command, address, and data bits.
- DS (data strobe): Optional line that marks the valid data edge when DDR is used.

## 14.4.2 OCTO-SPI configuration

### 14.4.2.1 OCTO-SPI configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "OCTO-SPI" > "Setup".




Make sure that the tab of the correct serial bus is selected.

**Protocol type**
Selects the protocol type to be decoded.

Remote command:
`SBUS<sb>:TYPE` on page 1362

**State**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
SBUS<sb>[:STATe] on page 1362

**CS, SCLK, IO n**
Sets the input channel for the respective line:

- CS: chip select
- CK: clock line
- IO0-IO7: data lines
- DS: data strobe line (optional, for DDR)

Remote command:
SBUS<sb>:OSPI:CSEL:SOURce on page 1435
SBUS<sb>:OSPI:CK:SOURce on page 1436
SBUS<sb>:OSPI:IOZero:SOURce on page 1438
SBUS<sb>:OSPI:IOONe:SOURce on page 1438
SBUS<sb>:OSPI:IOTWo:SOURce on page 1438
SBUS<sb>:OSPI:IOTHree:SOURce on page 1438
SBUS<sb>:OSPI:IOFR:SOURce on page 1439
SBUS<sb>:OSPI:IOFV:SOURce on page 1439
SBUS<sb>:OSPI:IOSX:SOURce on page 1439
SBUS<sb>:OSPI:IOSV:SOURce on page 1439
SBUS<sb>:OSPI:DS:SOURce on page 1440

**Polarity: CS, IO<n>**
Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

Remote command:
SBUS<sb>:OSPI:CSEL:POLarity on page 1435
SBUS<sb>:OSPI:IOZero:POLarity on page 1437
SBUS<sb>:OSPI:IOONe:POLarity on page 1437
SBUS<sb>:OSPI:IOTWo:POLarity on page 1437
SBUS<sb>:OSPI:IOTHree:POLarity on page 1437
SBUS<sb>:OSPI:IOFR:POLarity on page 1437
SBUS<sb>:OSPI:IOFV:POLarity on page 1437
SBUS<sb>:OSPI:IOSX:POLarity on page 1437
SBUS<sb>:OSPI:IOSV:POLarity on page 1437

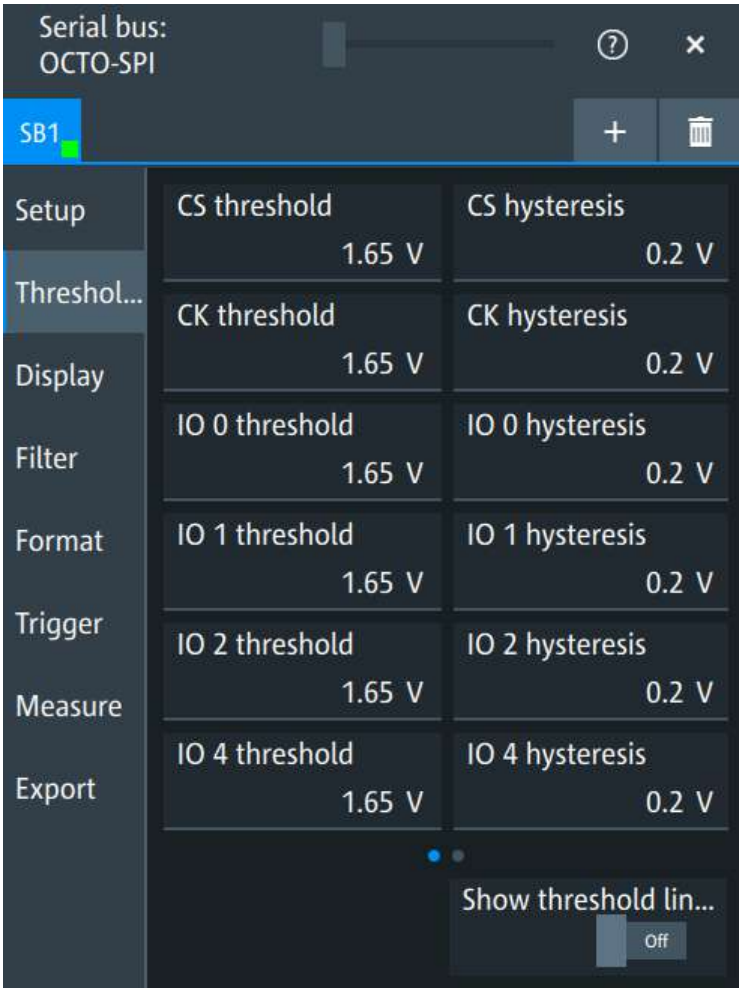
**Instruction**
Selects the instruction mode that defines how many lines are used to transmit data.

| | |
|---|---|
| "1S", "2S", "4S", "8S" | Defines the number of lines in single data rate mode, i.e. 8S = “8-line-Single-Data-Rate” (octal-SDR) |
| "1D", "2D", "4D", "8D" | Defines the number of lines in dual data rate mode, i.e. 4D = “4-line-Dual-Data-Rate” (quad-DDR) |

Remote command:
SBUS<sb>:OSPI:MODE on page 1436

#### 14.4.2.2 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "OCTO-SPI" > "Thresholds".




**CS/CK/IO<n>/DS threshold**
Set the threshold for each signal channel. Enter the value directly in the fields.

Remote command:
`SBUS<sb>:OSPI:CSEL:THReshold` on page 1435
`SBUS<sb>:OSPI:CK:THReshold` on page 1437
`SBUS<sb>:OSPI:IOZero:THReshold` on page 1439
`SBUS<sb>:OSPI:IOONe:THReshold` on page 1439
`SBUS<sb>:OSPI:IOTWo:THReshold` on page 1439
`SBUS<sb>:OSPI:IOTHree:THReshold` on page 1439
`SBUS<sb>:OSPI:IOFR:THReshold` on page 1439
`SBUS<sb>:OSPI:IOFV:THReshold` on page 1439
`SBUS<sb>:OSPI:IOSX:THReshold` on page 1439
`SBUS<sb>:OSPI:IOSV:THReshold` on page 1439
`SBUS<sb>:OSPI:DS:THReshold` on page 1440

**CS/CK/IO<n>/DS hysteresis**
Set the hysteresis for each signal channel. Enter the value directly in the fields.

Remote command:
SBUS<sb>:OSPI:CSEL:HYSTeresis on page 1435
SBUS<sb>:OSPI:CK:HYSTeresis on page 1436
SBUS<sb>:OSPI:IOZero:HYSTeresis on page 1437
SBUS<sb>:OSPI:IOONe:HYSTeresis on page 1437
SBUS<sb>:OSPI:IOTWo:HYSTeresis on page 1437
SBUS<sb>:OSPI:IOTHree:HYSTeresis on page 1437
SBUS<sb>:OSPI:IOFR:HYSTeresis on page 1437
SBUS<sb>:OSPI:IOFV:HYSTeresis on page 1437
SBUS<sb>:OSPI:IOSX:HYSTeresis on page 1437
SBUS<sb>:OSPI:IOSV:HYSTeresis on page 1437
SBUS<sb>:OSPI:DS:HYSTeresis on page 1439

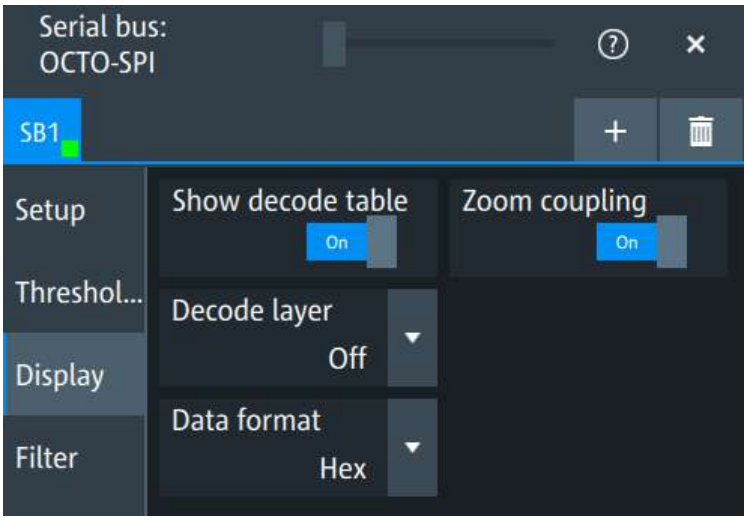
**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
SBUS<sb>:THReshold on page 1364

### 14.4.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.



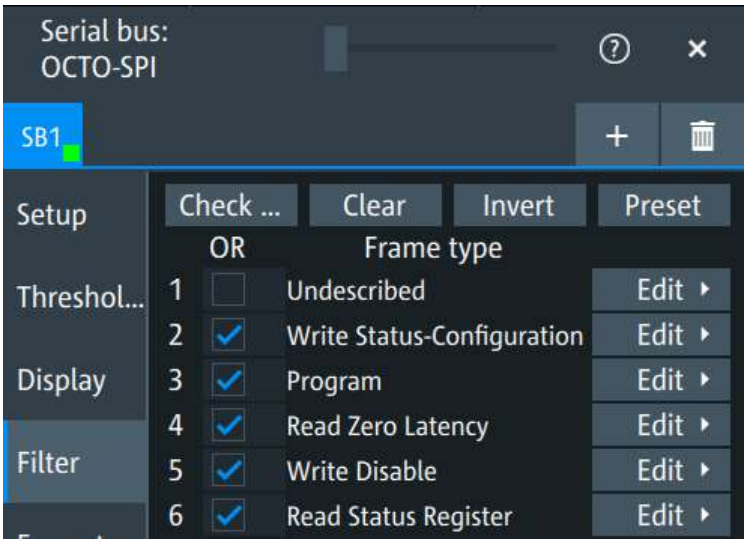

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

| | |
|---|---|
| "Off" | Disables the display of the decode layer. |
| "Edges" | Enables the display of all edges for each line. |
| "Bits" | Enables the display of all bits for each line. |
| "Merged bits" | Enables the display of the bits merged across multiple lines. |
| "Words" | Enables the display of all words. |

### 14.4.3 OCTO SPI filter

Access: "Menu" > "Apps" > "Protocol" tab > "OCTO-SPI" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

| | |
|---|---|
| "Check all" | Enables the filter for all available frames and error types. |
| "Clear" | Disables the filter for all available frames and error types. |
| "Invert" | Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa. |
| "Preset" | Presets the state of the selected frames and error types. |

Remote command:
`SBUS<sb>:OSPI:FILTer:CHKall` on page 1441
`SBUS<sb>:OSPI:FILTer:CLR` on page 1441
`SBUS<sb>:OSPI:FILTer:INVert` on page 1441
`SBUS<sb>:OSPI:FILTer:RST` on page 1442

**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

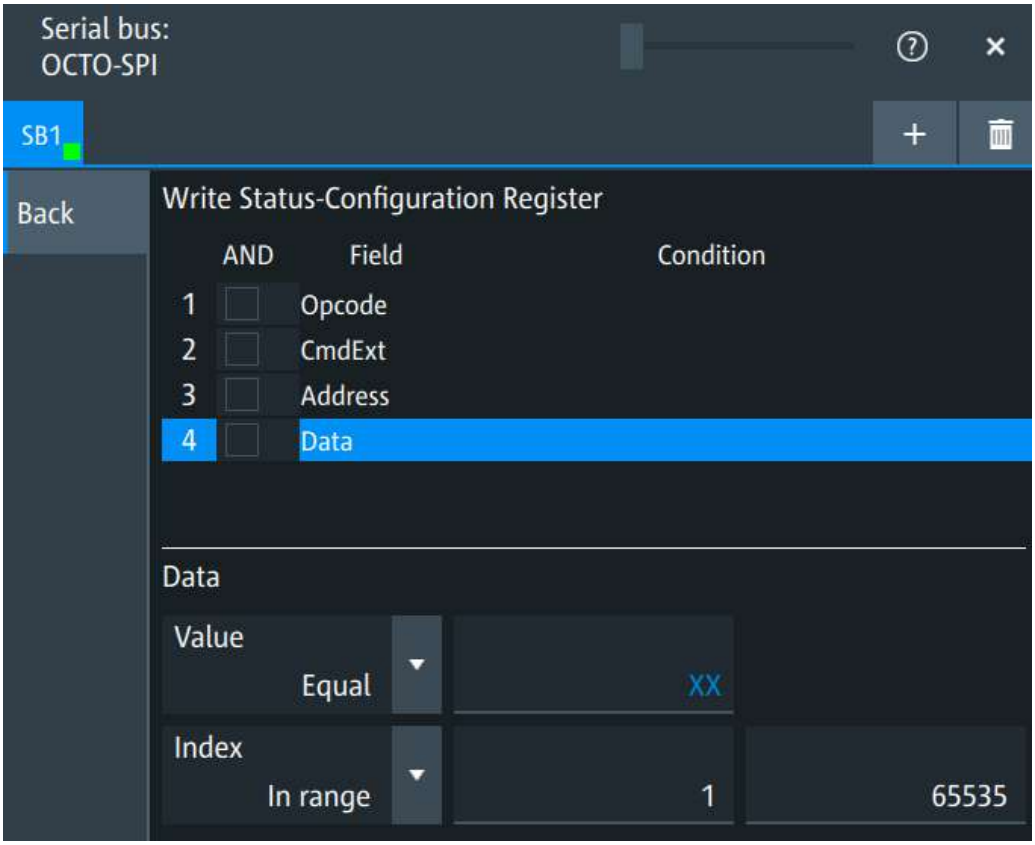
For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

The available frame types are the opcodes and an "Undescribed", to support custom coding.

If you have created own opcodes, they also appear in this list.

See Section 14.1.5, "Format", on page 497.

**Edit**
Opens a dialog to define the details of the selected frame.




"Field"
Enables the field type that you want to filter on for the selected frame.

Depending on the frame type, the available fields are:

- "Opcode"
- "Addr"
- "CmdExt"
- "Data"

Remote command:
`SBUS<sb>:OSPI:FILTer:FIENable` on page 1445
`SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:ENABle` on page 1445

"Condition" Displays the value condition for the selected field.

Remote command:
SBUS<sb>:OSPI:FILTer:BIT on page 1442
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:BIT on page 1442

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:OSPI:FILTer:DMAX on page 1442
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1442
SBUS<sb>:OSPI:FILTer:DMIN on page 1443
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1443
SBUS<sb>:OSPI:FILTer:DOPerator on page 1443
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1443

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:OSPI:FILTer:IMAX on page 1443
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1443
SBUS<sb>:OSPI:FILTer:IMIN on page 1444
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1444
SBUS<sb>:OSPI:FILTer:IOPerator on page 1444
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1444

**Error type**
Enables filtering on the selected error type.

Available error types are "Length error", "Opcode error", "Incomplete".

Remote command:
SBUS<sb>:OSPI:FILTer:ERENable on page 1445
SBUS<sb>:OSPI:FILTer:ERRor<n>:ENABle on page 1445

### 14.4.4 OCTO-SPI format configuration

Access: "Menu" > "Apps" > "Protocol" tab > "OCTO-SPI" > "Format".

See Section 14.1.5, "Format", on page 497.

**Add**
Adds a new item to the opcode list.

Remote command:
SBUS<sb>:OSPI:OPCode:APPend on page 1446

**Delete**
Deletes the currently selected opcode from the list.

Remote command:
SBUS<sb>:OSPI:OPCode:DELete on page 1446

**Delete all**
Deletes all defined opcodes from the list.

Remote command:
SBUS<sb>:OSPI:OPCode:DALL on page 1446

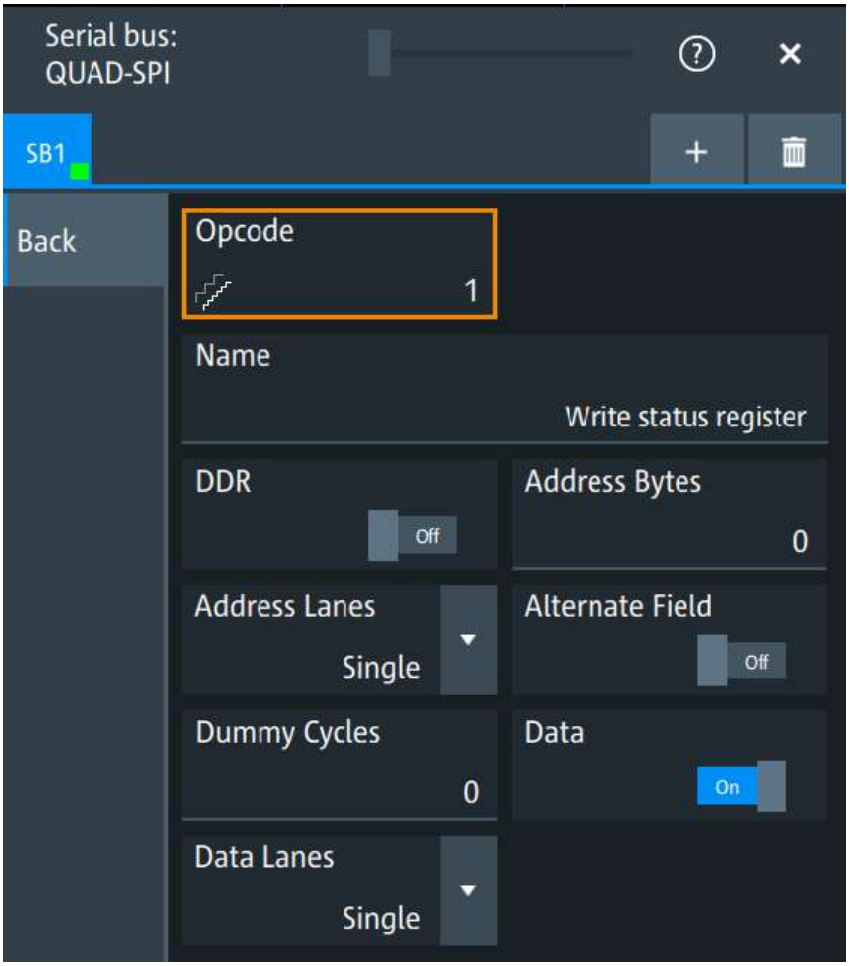
**Open, Save as**
Opens a dialog box where you can select a filename and a path for the export results file.

Remote command:
SBUS<sb>:OSPI:LDOPcode on page 1447
SBUS<sb>:OSPI:SVOP on page 1447

**Preset**
Resets the opcode fields to the predefined values.

Remote command:
SBUS<sb>:OSPI:OPCode:RESet on page 1447

**Opcode settings**




You can define the following settings for each opcode separately:

**Command ← Opcode settings**
Sets the opcode value.

Remote command:
`SBUS<sb>:OSPI:OPCode:ITEM<n>:CODE` on page 1448

**Name ← Opcode settings**
Sets the name for the opcode.

Remote command:
`SBUS<sb>:OSPI:OPCode:ITEM<n>:NAME` on page 1448

**Command extension ← Opcode settings**
If enabled, an extra extension byte is sent immediately after the command opcode.

Remote command:
`SBUS<sb>:OSPI:OPCode:ITEM<n>:CMMode` on page 1448

**Cmd. ext.mode ← Opcode settings**
Selects how many lines are used for sending the command extension byte.

"1S", "2S", "4S", "8S" Number of lines in single data rate mode, i.e. 8S = “8-line-Single-Data-Rate” (octal-SDR)

"1D", "2D", "4D", "8D" Number of lines in dual data rate mode, i.e. 4D = “4-line-Dual-Data-Rate” (quad-DDR)

Remote command:
`SBUS<sb>:OSPI:OPCode:ITEM<n>:CMEXtension` on page 1449

**Address Bytes ← Opcode settings**
Sets the address byte. It specifies the location in the flash memory where the operation (e.g., read, write) is performed.

Remote command:
`SBUS<sb>:OSPI:OPCode:ITEM<n>:ADBYtes` on page 1449

**Address mode ← Opcode settings**
Selects how many lines are used to send the address bytes to the flash memory.

"1S", "2S", "4S", "8S" Number of lines in single data rate mode, i.e. 8S = “8-line-Single-Data-Rate” (octal-SDR)

"1D", "2D", "4D", "8D" Number of lines in dual data rate mode, i.e. 4D = “4-line-Dual-Data-Rate” (quad-DDR)

Remote command:
`SBUS<sb>:OSPI:OPCode:ITEM<n>:ADMode` on page 1449

**Latency Cycles ← Opcode settings**
Sets the number of latency cycles. Latency cycles are clock cycles inserted after the address or other command sequences but before data transfer begins. These cycles allow the flash memory device additional time to perform internal operations or latch onto the correct data to ensure accurate read or write operations.

Remote command:
`SBUS<sb>:OSPI:OPCode:ITEM<n>:LACYcles` on page 1450

**Data type ← Opcode settings**
Determines the direction of the data flow for the payload after the command and address are sent.

"Read data" Controller is requesting information from the device.

"Write data" Controller is sending information to the device.

"No data" Controller sends a command and sometimes an address, but no actual payload is transferred.

Remote command:
`SBUS<sb>:OSPI:OPCode:ITEM<n>:DTTYpe` on page 1450

**Data mode ← Opcode settings**
Selects how many lines are used for the data transfer.

The data lines refer to the physical connections through which data is transmitted between the main (typically a microcontroller) and the secondary (typically a flash memory device).

"1S", "2S", "4S", "8S" Number of lines in single data rate mode, i.e. 8S = “8-line-Single-Data-Rate” (octal-SDR)

"1D", "2D", "4D", "8D" Number of lines in dual data rate mode, i.e. 4D = “4-line-Dual-Data-Rate” (quad-DDR)

Remote command:
`SBUS<sb>:OSPI:OPCode:ITEM<n>:DTMode` on page 1451

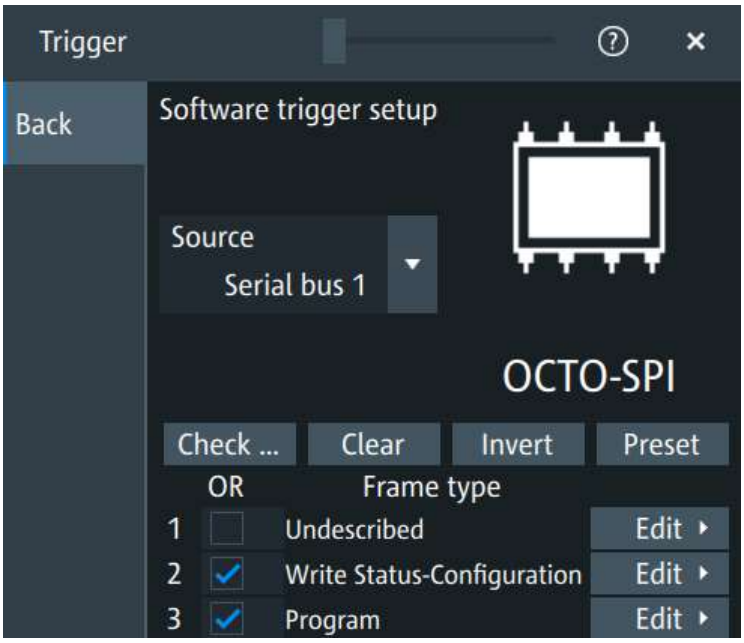
## 14.4.5 OCTO SPI software trigger

### 14.4.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.4.5.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "OCTO-SPI" > "Trigger" tab > "Setup Software Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:OSPI:CHKall` on page 1452
`TRIGger:SBSW:OSPI:CLR` on page 1452
`TRIGger:SBSW:OSPI:INVert` on page 1452
`TRIGger:SBSW:OSPI:RST` on page 1452

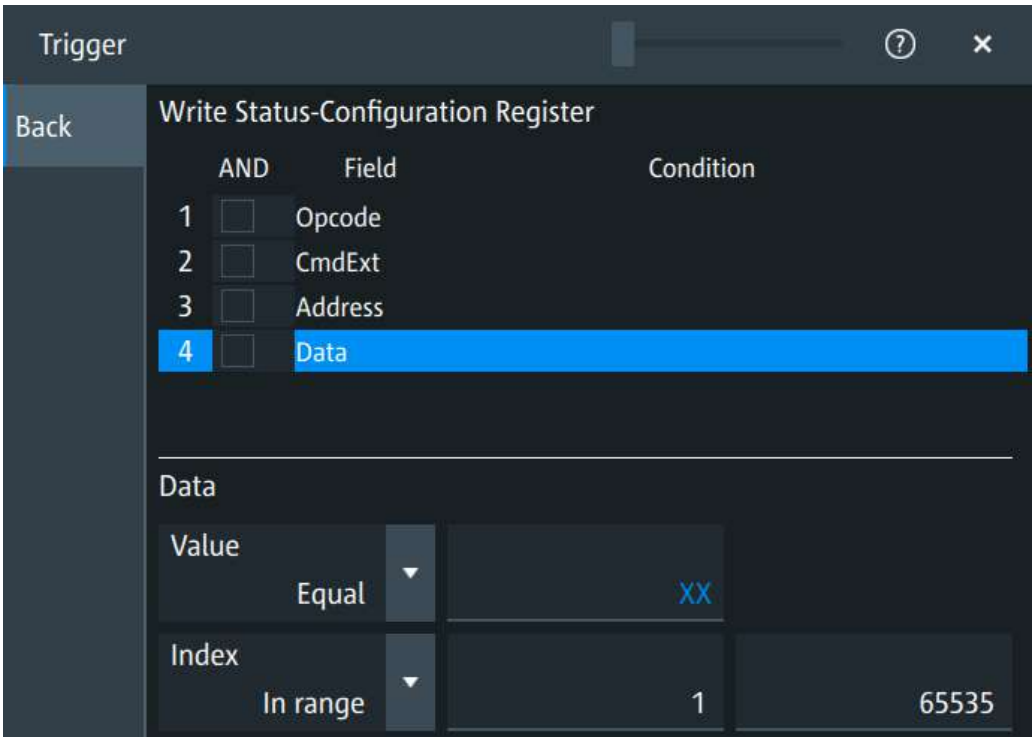
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The available frame is "Data".

Remote command:
`TRIGger:SBSW:OSPI:FRENable` on page 1453
`TRIGger:SBSW:OSPI:FRAMe<fr>:ENABle` on page 1453

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.



"Field" Enables the field type that you want to trigger on for the selected frame.
The following fields are available: "Opcode", "Addr", "Alt", "Dummy" and "Data".

Remote command:
TRIGger:SBSW:OSPI:FIENable on page 1454
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:ENABle on page 1454

"Condition" Displays the value condition for the selected field.

Remote command:
TRIGger:SBSW:OSPI:BIT on page 1453
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:BIT on page 1453

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
TRIGger:SBSW:OSPI:DMAX on page 1453
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:DMAX on page 1453
TRIGger:SBSW:OSPI:DMIN on page 1454
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:DMIN on page 1454
TRIGger:SBSW:OSPI:DOPerator on page 1454
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:DOPerator on page 1454

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:OSPI:IMAX on page 1455
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:IMAX on page 1455
TRIGger:SBSW:OSPI:IMIN on page 1455
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:IMIN on page 1455
TRIGger:SBSW:OSPI:IOPerator on page 1456
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:IOPerator on page 1456

**Error type**
Enables triggering on the selected error type.

The following error types are available: "Length error" and "Opcode error".

Remote command:
TRIGger:SBSW:OSPI:ERENable on page 1456
TRIGger:SBSW:OSPI:ERRor<m>:ENABle on page 1456

### 14.4.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

## 14.4.7 OCTO-SPI decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

### Decode results table

The "Decode results" table contains information about all decoded frames.

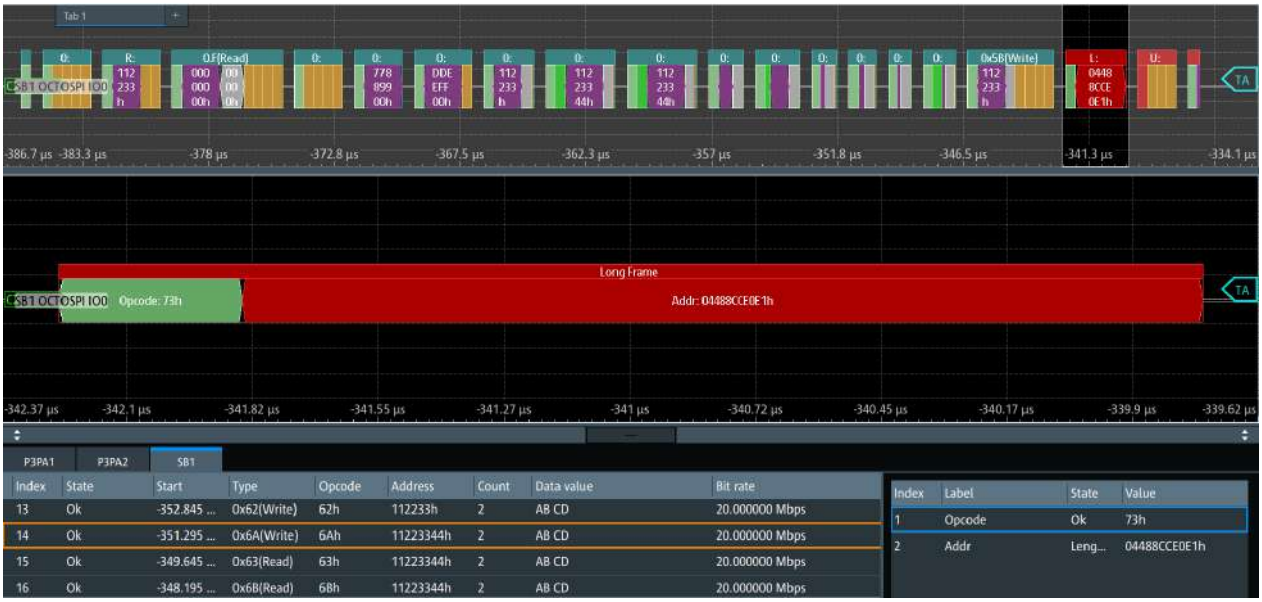



*Figure 14-13: Decoded OCTO-SPI signal*

The decode results table contains information about all decoded frames.

*Table 14-5: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Index of the decoded frame |
| State | Overall state of the frame |
| Start | Time of frame start |
| Type | Frame type |
| Opcode | Operation code value |
| Address | Address value |
| Count | Number of words in the frame |
| Data value | Value of the data frame. Select the data format in the "Display" tab. |
| Bit rate | Value of the bit rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

*Table 14-6: Content of the frame details table*

| Column | Description |
|---|---|
| Index | Number of the decoded field |
| Label | Name of the field |
| State | State of the field |
| Value | Value of the field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.4.5, "Decode results", on page 1457.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example OCTO-SPI export file

```
Index,Start,Stop,State,Type,Command,CommandExt,Address,Count,Bit rate
1,-0.000385847,-0.000385447,'OK','0.A',00h,---,---,---,20000000
2,-0.000384947,-0.000382945,'OK','0.B(Read)',FFh,---,---,4,20000000
3,-0.000382448,-0.00038004,'OK','Read Zero Latency',03h,---,112233h,2,20000000
4,-0.000379547,-0.000374945,'OK','0.F(Read)',01h,---,00000000h,4,20000000


Details frame 2
Index,Label,State,Value
1,'Data: 1','OK',01h
2,'Data: 2','OK',02h
3,'Data: 3','OK',03h
4,'Data: 4','OK',04h
```

## 14.5 I²C (IC electronics, option R&S MXO3-K510)

The Inter-Integrated Circuit is a simple, low-bandwidth, low-speed protocol used for communication between on-board devices, for example, in LCD and LED drivers, RAM, EEPROM, and others.

For basic information on I²C, you can also refer to a video, available on the Rohde & Schwarz YouTube channel: Understanding I2C.

**Requirements**

For performing I²C decode measurements, you need the following:

- MXO 3 with 2 available channels. The channels can be a combination of the following:
  - Analog channels (C1-C4)
  - Logic channels (D0-D15) (requires option R&S MXO3-B1)
- Option R&S MXO3-K510

- About the I²C protocol ..... 566
- I²C configuration ..... 568
- I²C filter ..... 572
- I2C format configuration ..... 575
- I²C hardware trigger ..... 580
- I²C software trigger ..... 584
- Measure ..... 587
- I²C decode results ..... 588

## 14.5.1 About the I²C protocol

This section provides an overview of protocol characteristics, data format, address types and trigger possibilities. For detailed information, read the "I2C-bus specification and user manual" available on the NXP manuals webpage at http://www.nxp.com/.

**I²C characteristics**

The main characteristics of I²C are:

- Two-wire design: serial clock (SCL) and serial data (SDA) lines
- Controller/ target communication: the controller generates the clock and addresses the targets. Targets receive the address and the clock. Both controller and targets can transmit and receive data.
- Addressing scheme: each target device is addressable by a unique address. Multiple target devices can be linked together and can be addressed by the same controller.
- Read/write bit: specifies if the controller reads (=1) or writes (=0) the data.
- Acknowledge: takes place after every byte. The receiver of the address or data sends the acknowledge bit to the transmitter.

The MXO 3 supports all operating speed modes: high-speed, fast mode plus, fast mode, and standard mode.

### Data transfer

The format of a simple I²C message (frame) with 7-bit addressing consists of the following parts:

- Start condition: a falling slope on SDA while SCL is high
- 7-bit address of the target device that is either written to or read from
- R/W bit: specifies if the data is written to or read from the target
- ACKnowledge bits: is issued by the receiver of the previous byte if the transfer was successful
  Exception: At read access, the controller terminates the data transmission with a NACK bit after the last byte.
- Data: several data bytes with an ACK bit after every byte
- Stop condition: a rising slope on SDA while SCL is high

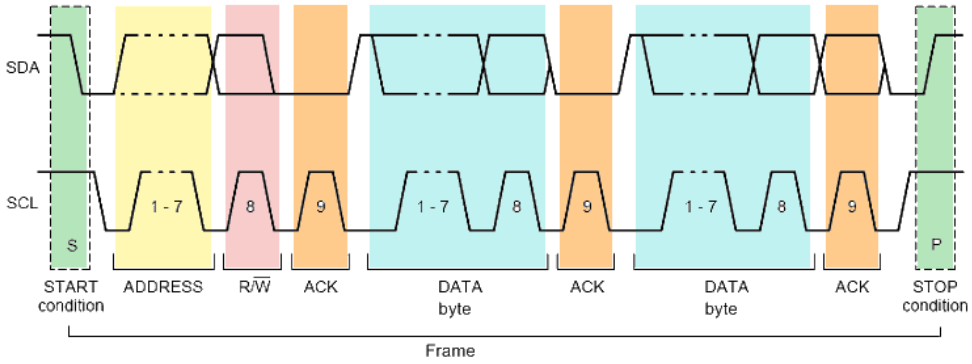



*Figure 14-14: I2C write access with a 7-bit address*

### Address types: 7-bit and 10-bit

Target addresses can be 7 bits or 10 bits long. A 7-bit address requires 1 byte, 7 bits for the address followed by the R/W bit.

A 10-bit address for write access requires 2 bytes: the first byte starts with the reserved sequence 11110, followed by the two MSB of the address and the write bit. The second byte contains the remaining 8 LSB of the address. The target acknowledges each address byte.




*Figure 14-15: 10-bit address, write access*

A 10-bit address for read access requires 3 bytes. The first 2 bytes are identical to the write access address. The third byte repeats the address bits of the first byte and sets the read bit.

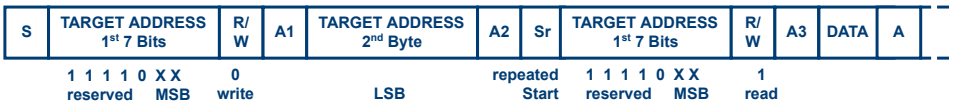



*Figure 14-16: 10-bit address, read access*

### Trigger

The MXO 3 can trigger on various parts of I²C messages. You must connect the data and clock lines to the input channels; triggering on math and reference waveforms is not possible.

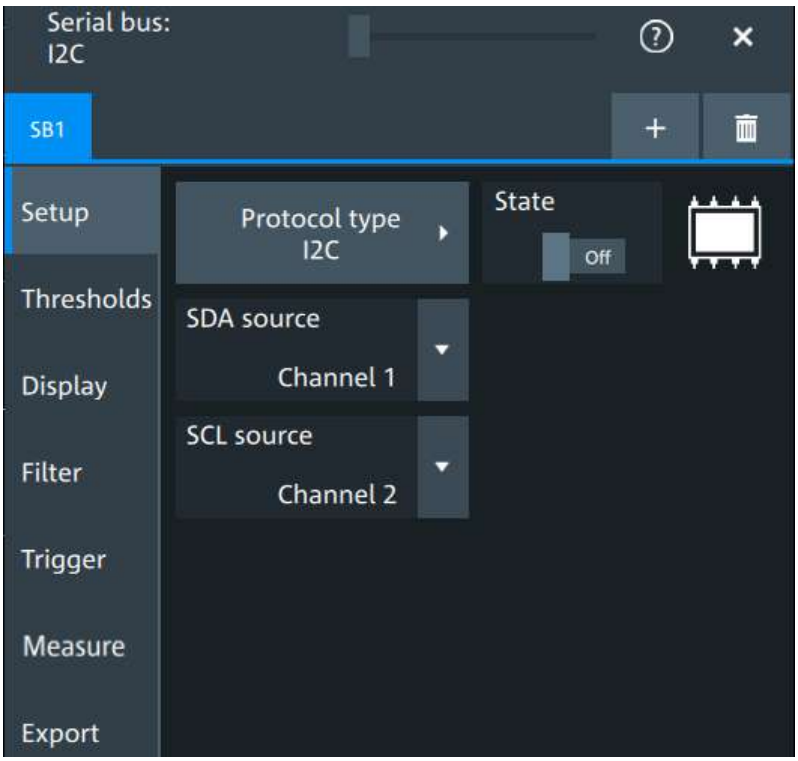
You can trigger on:

- Start or stop condition
- Repeated start condition
- Transfer direction (read or write)
- Bytes with missing acknowledge bit
- Specific data pattern in the message

## 14.5.2 I²C configuration

### 14.5.2.1 I²C configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "I2C" > "Setup".




Make sure that the tab of the correct serial bus is selected.

**Protocol type**
Selects the protocol type to be decoded.

Remote command:
`SBUS<sb>:TYPE` on page 1362

**State**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
`SBUS<sb>[:STATe]` on page 1362

**SDA source**
Sets the source channel to which the data line is connected.

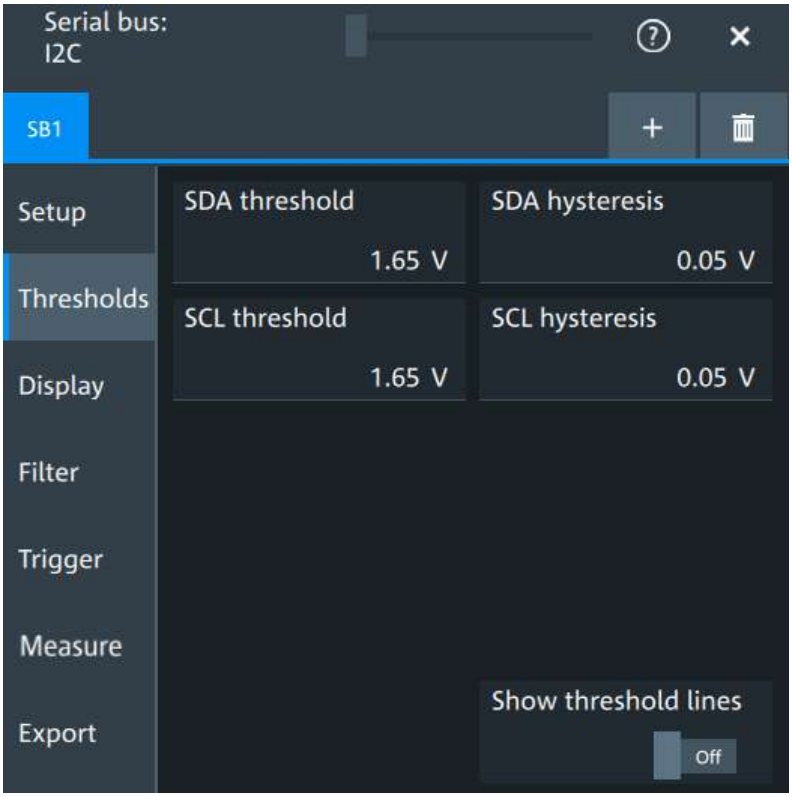
Remote command:
`SBUS<sb>:I2C:SDA:SOURce` on page 1465

**SCL source**
Selects the source channel to which the clock line is connected.

Remote command:
`SBUS<sb>:I2C:SCL:SOURce` on page 1465

#### 14.5.2.2 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "I2C" > "Thresholds".




**SDA/SCL threshold, SDA/SCL hysteresis**
Set the threshold for the SDA and SCL channels. Enter the value directly in the fields.

Additional to the threshold, you can also set a hysteresis for each channel.

Remote command:
`SBUS<sb>:I2C:SCL:THReshold` on page 1465
`SBUS<sb>:I2C:SDA:THReshold` on page 1466
`SBUS<sb>:I2C:SCL:HYSTeresis` on page 1464
`SBUS<sb>:I2C:SDA:HYSTeresis` on page 1465

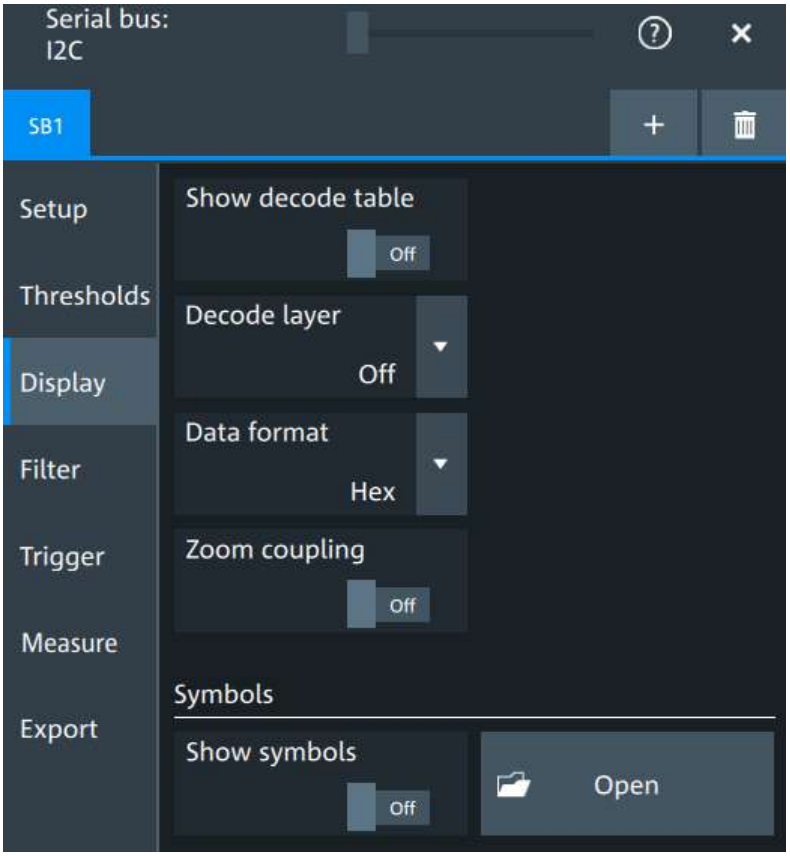
**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
`SBUS<sb>:THReshold` on page 1364

#### 14.5.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.




**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

See "Decode layer" on page 493.

**Data format**
See "Data format" on page 493.

**Zoom coupling**
See "Zoom coupling" on page 494.

**Show symbols**
Symbol lists are protocol-specific. Label lists for I2C are available in CSV format.

For details, see I²C symbols.

Remote command:
`SBUS<sb>:I2C:SYMBols` on page 1466
`SBUS<sb>:I2C:NEWList` on page 1467

#### 14.5.2.4 I²C symbols

For the I²C protocol, you can upload symbol lists containing IDs and a symbolic name for each node. You can load a file in one of the supported formats. Symbol lists for I²C are available in CSV format.

An I²C label file contains three values for each address:

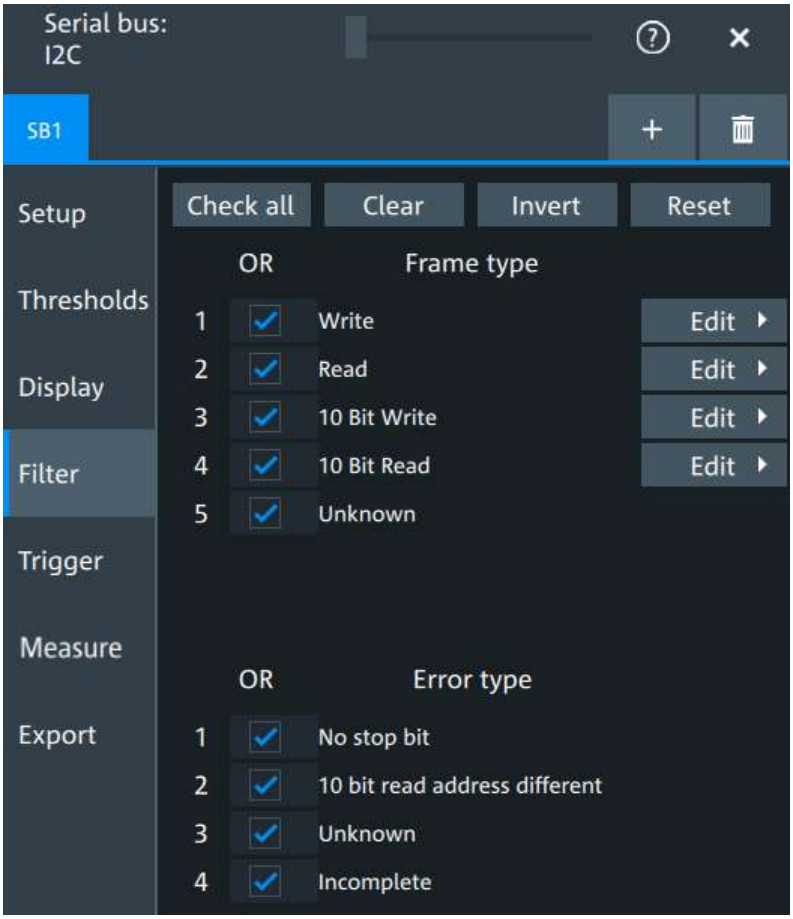
- Address type, 7-bit or 10-bit long
- Address value
- Symbolic label: name of the address, specifying its function in the bus network.

**Example: I²C CSV file**

```
# ------------------------------------------------------------------------
@FILE_VERSION  = 1.00
@PROTOCOL_NAME = i2c
# ------------------------------------------------------------------------
# Labels for I2C protocol
#   Column order: Identifier type, Identifier value, Label
# ------------------------------------------------------------------------
7,0x1E,Voltage
7,38h,Pressure
7,2Ah,Temperature
7,16h,Speed
7,118,Acceleration
7,07h,HighSpeed_Master_0x3
7,51h,EEPROM
10,3A2h,DeviceSetup
10,1A3h,GatewayStatus
10,06Eh,LeftSensor
# ------------------------------------------------------------------------
```

### 14.5.3 I²C filter

Access: "Menu" > "Apps" > "Protocol" tab > "I2C" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

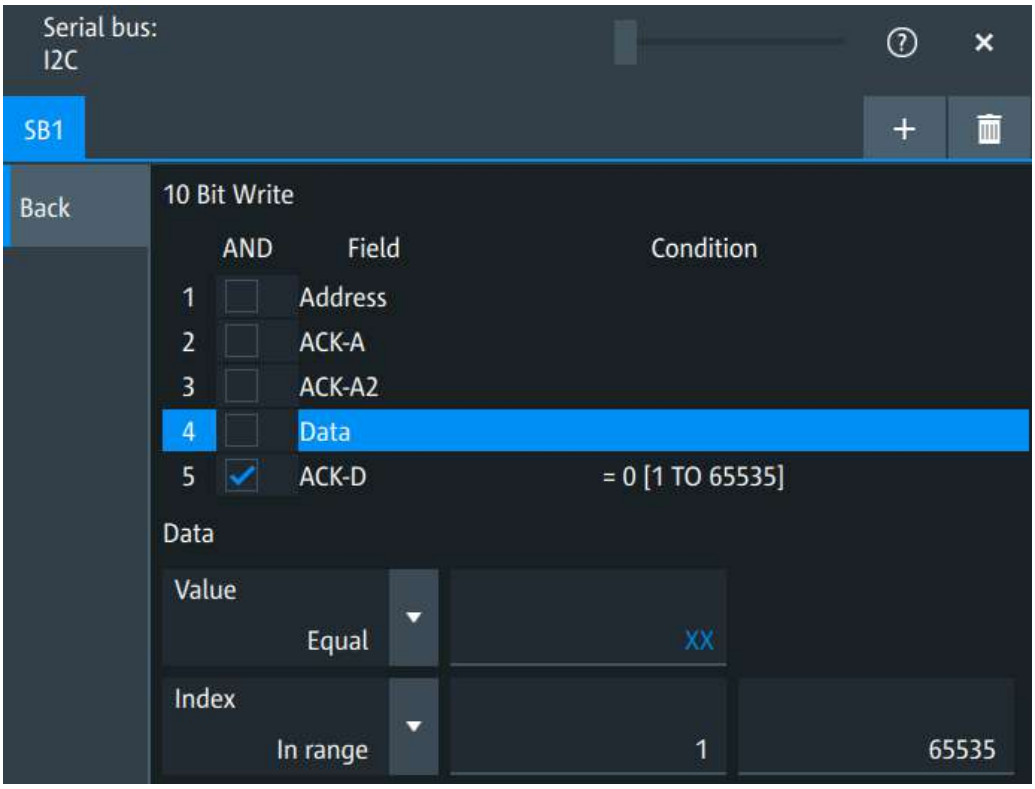
Remote command:
`SBUS<sb>:I2C:FILTer:CHKall` on page 1468
`SBUS<sb>:I2C:FILTer:CLR` on page 1468
`SBUS<sb>:I2C:FILTer:INVert` on page 1468
`SBUS<sb>:I2C:FILTer:RST` on page 1468

**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

Remote command:
SBUS<sb>:I2C:FILTer:FRENable on page 1469
SBUS<sb>:I2C:FILTer:FRAMe<fr>:ENABle on page 1469

**Edit**
Opens a dialog to define the details of the selected frame.




"Field" Enables the field type that you want to filter on for the selected frame. The available fields are "Address", "R/W", "ACK_A", "ACK_A2", "Data", and "ACK_D".

Remote command:
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:ENABle on page 1472
SBUS<sb>:I2C:FILTer:FIENable on page 1472

"Condition" Displays the value condition for the selected field.

Remote command:
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:BIT on page 1472

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1469
SBUS<sb>:I2C:FILTer:DMAX on page 1469
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1469
SBUS<sb>:I2C:FILTer:DMIN on page 1469
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1470
SBUS<sb>:I2C:FILTer:DOPerator on page 1470

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1471
SBUS<sb>:I2C:FILTer:IMAX on page 1471
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1471
SBUS<sb>:I2C:FILTer:IMIN on page 1471
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1471
SBUS<sb>:I2C:FILTer:IOPerator on page 1471

**Error type**
Enables filtering on the selected error type.

Remote command:
SBUS<sb>:I2C:FILTer:ERENable on page 1470
SBUS<sb>:I2C:FILTer:ERRor<n>:ENABle on page 1470

### 14.5.4 I2C format configuration

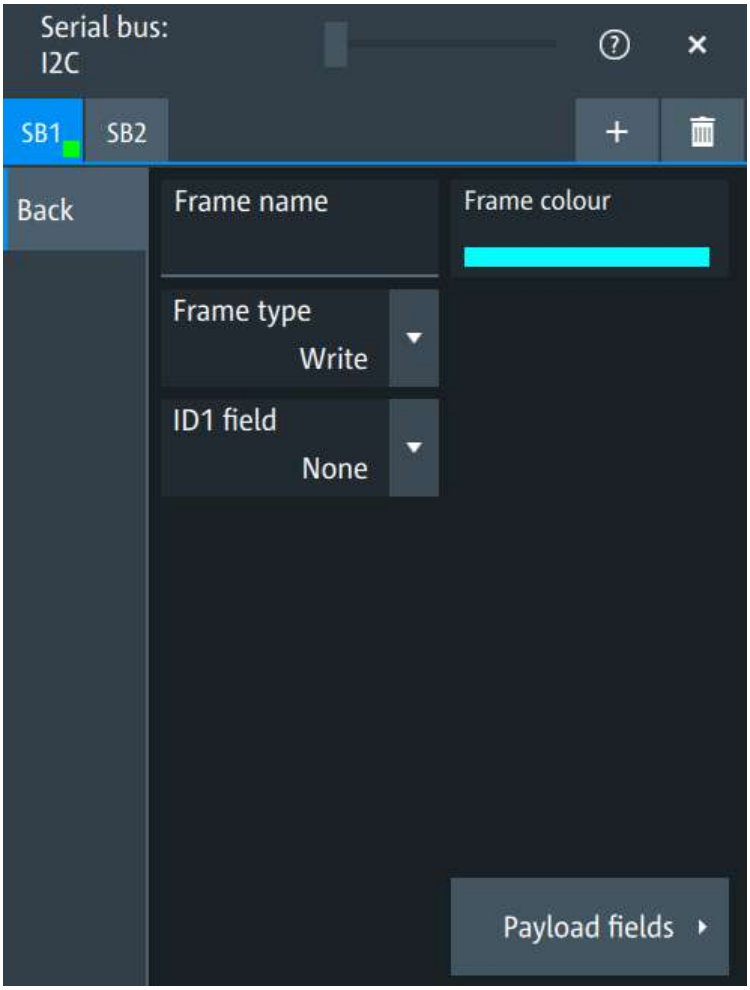
Access: "Menu" > "Apps" > "Protocol" tab > "I2C" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
SBUS<sb>:I2C:FORMat:SELect on page 1473

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:I2C:FORMat:ADDFrame on page 1473

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:I2C:FORMat:CLR on page 1474
SBUS<sb>:I2C:FORMat:FCOunt? on page 1474

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:I2C:FORMat:LOAD on page 1474

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:I2C:FORMat:SAVE on page 1474

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:NAME on page 1475

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:COLor on page 1476

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:TYPE on page 1476

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID1Field on page 1476
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID2Field on page 1476
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID3Field on page 1476

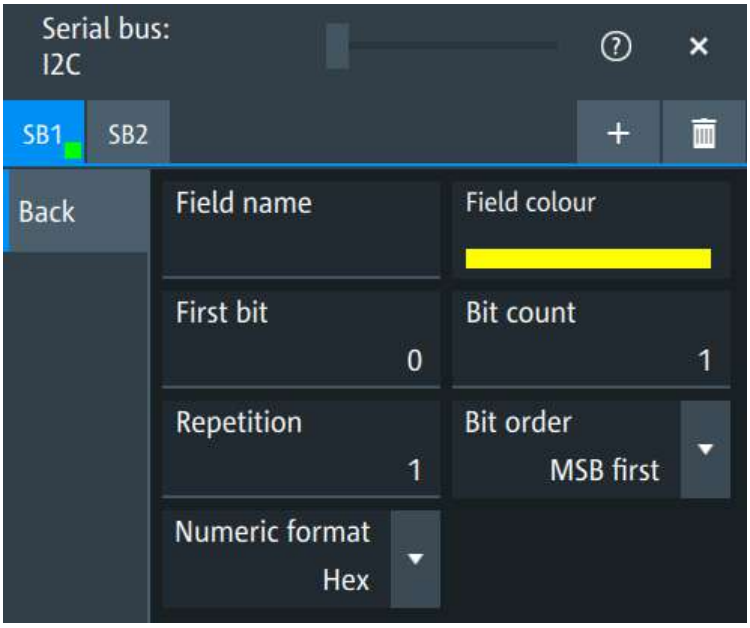
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID1Value on page 1477
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID2Value on page 1477
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID3Value on page 1477

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ADDField on page 1475

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1477

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1477

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1478

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.

- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  – Bit count = 2: truncated. Result: =00
  – Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  – Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  – Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1478

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
`SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:REPetition` on page 1478

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
`SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:BITorder` on page 1479

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:
- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  – "=1HL111000": valid (read as binary)
  – "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  – "=1010": valid (read as HEX, with a total of 16 bits)
  – "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)

– "=5X12": invalid
– "=1H33": invalid

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1479

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1480

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1480

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1481

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1481

**Parameter A,B**
Defines the parameters of the linear function.

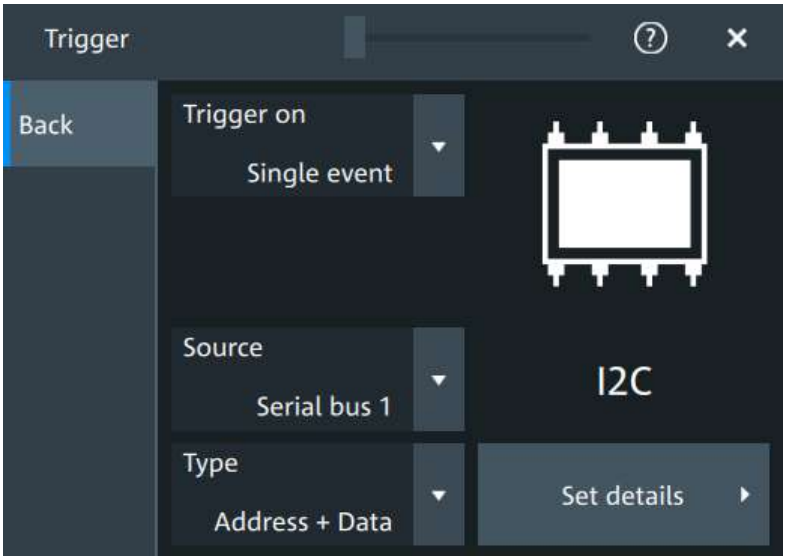
- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1481
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1481

### 14.5.5 I²C hardware trigger

Access: "Menu" > "Apps" > "Protocol" tab > "I2C" > "Trigger" tab > "Setup Hardware Trigger"




**Type**
Selects the trigger type for I²C analysis.

Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.

Remote command:
`TRIGger:SBHW:I2C:TYPE` on page 1482

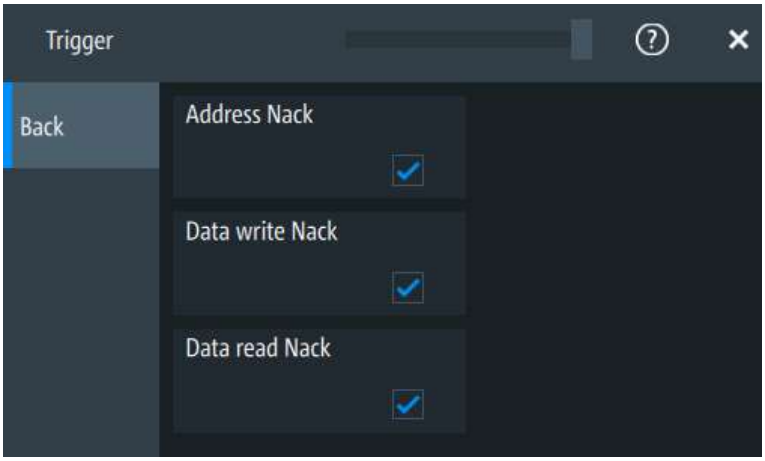
**Start ← Type**
Sets the trigger to the start of the message. The start condition is a falling edge on SDA while SCL is high. The trigger instant is the falling edge of the SDA line.

You can change the SDA and SCL lines here if necessary.

**Repeated start ← Type**
Sets the trigger to a repeated start - when the start condition occurs without previous stop condition. Repeated start conditions occur when a controller exchanges multiple messages with a target device without releasing the bus.

**Stop ← Type**
Sets the trigger to the end of the message. The stop condition is a rising slope on SDA while SCL is high.

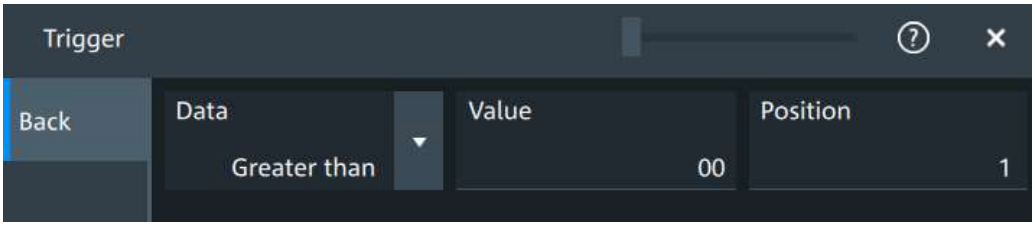
**No Ack ← Type**
Missing acknowledge bit: the instrument triggers if the data line remains HIGH during the clock pulse following a transmitted byte.




**Address ← Type**
Sets the trigger to one specific address condition or a combination of address conditions. The trigger time is the falling clock edge of the acknowledge bit after the address.




**Data ← Type**
Sets the trigger to one specific data condition or a combination of address conditions.




**Address and data ← Type**
Sets the trigger to a combination of address and data condition.




**No Ack conditions**
Selects which missing acknowledge bits is detected if the trigger type is set to "No Ack".

"Address Nack"
No secondary recognizes the address.

"Data write Nack"
The addressed secondary does not accept the data.

"Data read Nack"
Marks the end of the read process when the primary reads data from the secondary. This Nack is sent according to the protocol definition, it is not an error.

Remote command:
`TRIGger:SBHW:I2C:ADNack` on page 1484
`TRIGger:SBHW:I2C:DWNack` on page 1485
`TRIGger:SBHW:I2C:DRNack` on page 1485

**Address type**
Sets the address length to be triggered on: 7 bit or 10 bit.

Remote command:
`TRIGger:SBHW:I2C:AMODe` on page 1484

**R/W bit**
Toggles the trigger condition between read and write access of the primary. Select "Either" if the transfer direction is not relevant for the trigger condition.

Remote command:
`TRIGger:SBHW:I2C:ACCess` on page 1483

**Address**
The trigger address setup consists of a comparison condition and one or two address patterns.

Defines the bit pattern of the secondary device address. The length of the entry is adjusted to the selected address type.

"Condition" Sets the comparison condition to a specific value or a range.

"From" Specifies the value or sets the start value of a range.

"To" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger:SBHW:I2C:ADDRess` on page 1483
`TRIGger:SBHW:I2C:ADDTo` on page 1484

**Data**
Specifies the trigger conditions for the data bit pattern.

"Condition" Sets the comparison condition to a specific value or a range.

"Value" Specifies the value or sets the start value of a range.
Enter the bytes in msb first bit order. The maximum pattern length is 64 bit. Waveform data is compared with the pattern byte-by-byte.

Remote command:
`TRIGger:SBHW:I2C:DCONdition` on page 1484
`TRIGger:SBHW:I2C:DMIN` on page 1485

**Position**
Sets the number of data bytes to be skipped after the address.

Remote command:
`TRIGger:SBHW:I2C:DPOSition` on page 1485

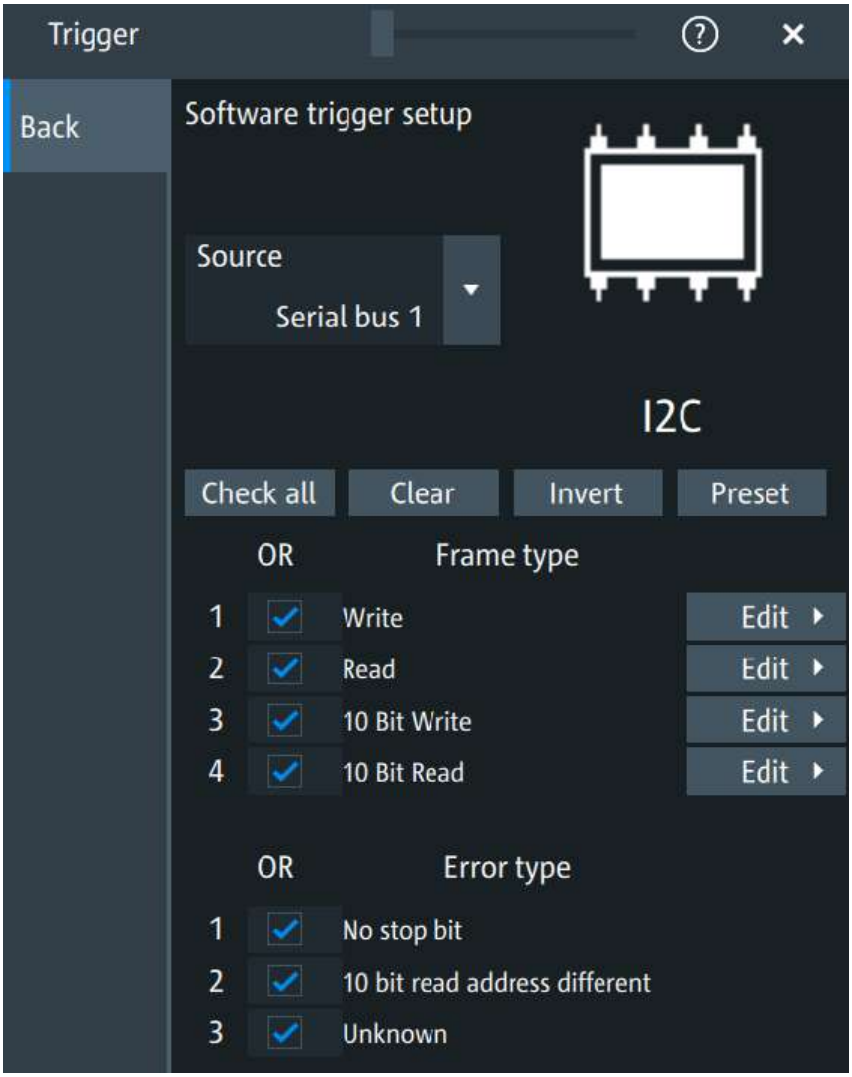
## 14.5.6 I²C software trigger

### 14.5.6.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.5.6.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "I2C" > "Trigger" tab > "Setup Software Trigger" > "Setup SB Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:I2C:CHKall` on page 1486
`TRIGger:SBSW:I2C:CLR` on page 1487

TRIGger:SBSW:I2C:INVert on page 1487
TRIGger:SBSW:I2C:RST on page 1487

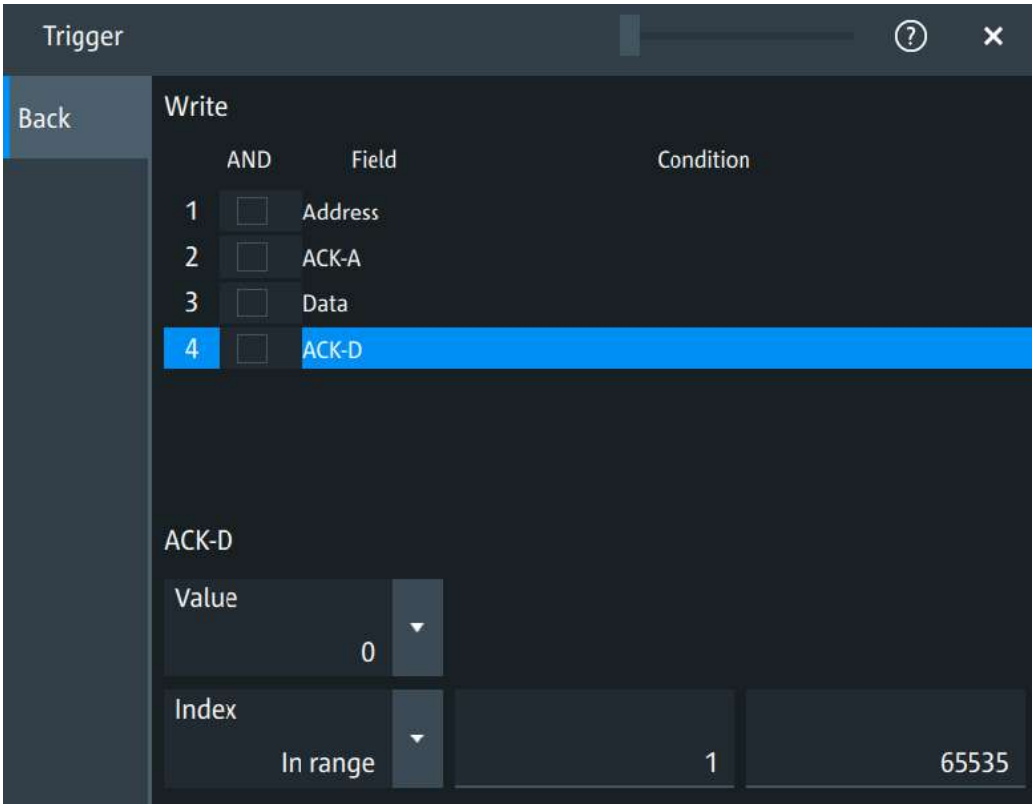
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: "Write", "Read", "10 Bit Write", "10 Bit Read".

Remote command:
TRIGger:SBSW:I2C:FRENable on page 1487
TRIGger:SBSW:I2C:FRAMe<fr>:ENABle on page 1487

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.



"Field" Enables the field type that you want to trigger on for the selected frame.
The available fields are "Address", "ACK-A", "ACK-A2", "Data", "ACK-D".

Remote command:
TRIGger:SBSW:I2C:FIENable on page 1489
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:ENABle on page 1489

"Condition" Displays the value condition for the selected field.

Remote command:
TRIGger:SBSW:I2C:BIT on page 1487
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:BIT on page 1487

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
TRIGger:SBSW:I2C:DMAX on page 1488
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DMAX on page 1488
TRIGger:SBSW:I2C:DMIN on page 1488
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DMIN on page 1488
TRIGger:SBSW:I2C:DOPerator on page 1489
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DOPerator on page 1489

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:I2C:IMAX on page 1489
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IMAX on page 1489
TRIGger:SBSW:I2C:IMIN on page 1490
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IMIN on page 1490
TRIGger:SBSW:I2C:IOPerator on page 1490
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IOPerator on page 1490

**Error type**
Enables triggering on the selected error type.

Remote command:
TRIGger:SBSW:I2C:ERENable on page 1490
TRIGger:SBSW:I2C:ERRor<m>:ENABle on page 1490

### 14.5.7 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.5.8 I²C decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

For more information on decoding I3C, you can also refer to the video, available on the Rohde & Schwarz YouTube channel: Decoding I2C with MXO Series Oscilloscope.

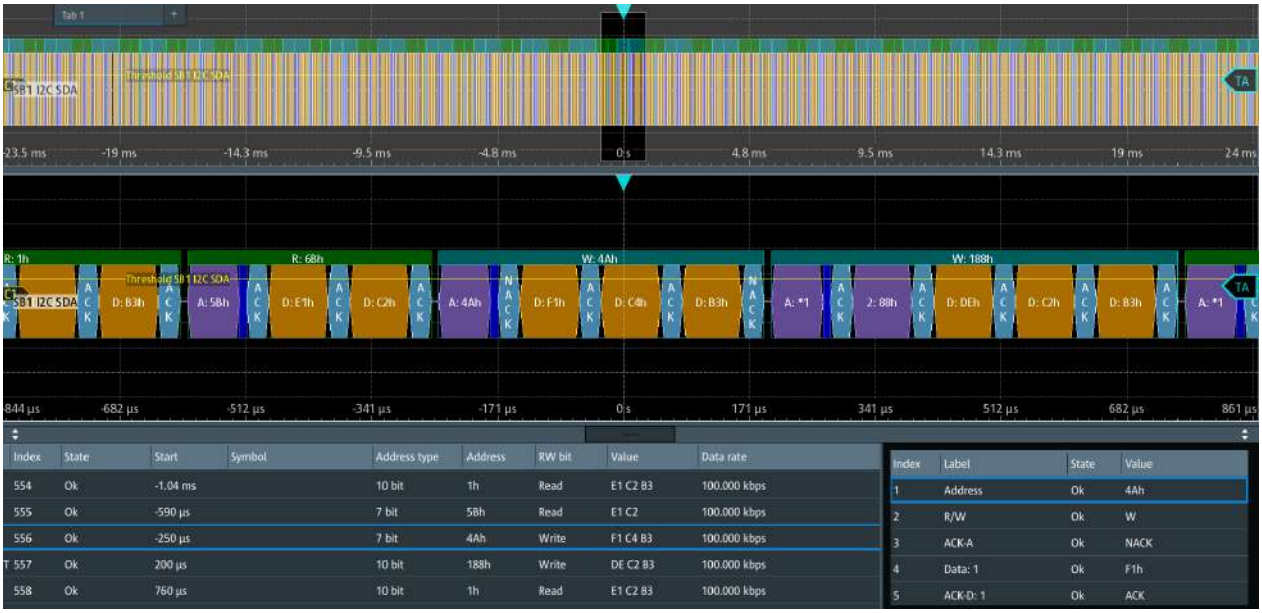



*Figure 14-17: Decoded I2C signal*

The decode results table contains information about all decoded frames.

*Table 14-7: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Index of the decoded frame |
| State | Overall state of the frame.<br>"Insuffcient waveform length" indicates that the frame is not completely contained in the acquisition. Change the horizontal scale, or move the reference point to the left to get a longer acquisition. |
| Start | Time of frame start |
| Symbol | Symbolic label of the frame. Available only, if "Show symbols" is enabled in the "Display" tab and a symbols list is loaded. |
| Address type | Address length, 7 bit or 10 bit |
| Address | Hexadecimal value of the address |
| RW bit | Value of the read/write bit |
| Value | The actual data payload values. Select the data format in the "Display" tab. |
| Data rate | Value of the data rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

***Table 14-8: Content of the frame details table***

| Item | Description |
|---|---|
| Index | Number of the decoded field |
| Label | Label (name) of each word of the data field |
| State | State of each word of the data field, for example "OK" or "Incomplete" |
| Value | Value of each word of the data field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.5.6, "Decode results", on page 1491.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example of an I²C export file

```
Index,Start,Stop,State,Address type,Address value,R/W bit,Bit rate
1,-0.25,-0.24945,'OK','BIT10',188h,'WRIT',100000
2,-0.24944,-0.249,'OK','BIT10',1h,'READ',100000
3,-0.24899,-0.24866,'OK','BIT7',5Bh,'READ',100000
4,-0.24865,-0.24821,'OK','BIT7',4Ah,'WRIT',100000
5,-0.2482,-0.24765,'OK','BIT10',188h,'WRIT',100000
6,-0.24764,-0.2472,'OK','BIT10',1h,'READ',100000

Details frame 1
Index,Formatted data,Ack bit start,Ack bit
1,DEh,-0.2497,'ACK'
2,C2h,-0.24959,'ACK'
3,B3h,-0.24948,'ACK'

Details frame 2
Index,Formatted data,Ack bit start,Ack bit
1,E1h,-0.24925,'ACK'
2,C2h,-0.24914,'ACK'
3,B3h,-0.24903,'ACK'

Details frame 3
Index,Formatted data,Ack bit start,Ack bit
```

```
1,E1h,-0.2488,'ACK'
2,C2h,-0.24869,'ACK'

Details frame 4
Index,Formatted data,Ack bit start,Ack bit
1,F1h,-0.24846,'ACK'
2,C4h,-0.24835,'ACK'
3,B3h,-0.24824,'NACK'

Details frame 5
Index,Formatted data,Ack bit start,Ack bit
1,DEh,-0.2479,'ACK'
2,C2h,-0.24779,'ACK'
3,B3h,-0.24768,'ACK'
```

## 14.6 I3C (IC electronics, option R&S MXO3-K550)

The Improved Inter-Integrated Circuit (I3C) is a low-speed MIPI protocol used for communication between chips and signaling patterns by a 2-wire serial data bus.

**Requirements**

For performing I3C decode measurements, you need the following:

- MXO 3 with 2 available channels. The channels can be a combination of the following:
  - Analog channels (C1-C4)
  - Logic channels (D0-D15) (requires option R&S MXO3-B1)
- Option R&S MXO3-K550

- About the I3C protocol ..... 590
- I3C configuration ..... 592
- I3C filter ..... 596
- I3C format configuration ..... 599
- I3C software trigger ..... 605
- Measure ..... 607
- I3C decode results ..... 607

### 14.6.1 About the I3C protocol

This section provides an overview of protocol characteristics, data format, address types and trigger possibilities. For detailed information, read the "MIPI I3C and I3C Basic" specifications, available at www.mipi.org/groups/I3C.

### I3C characteristics

The Improved Inter-Integrated Circuit (I3C) protocol is a high-speed, low-power, and low-latency interface standard. Typical applications are mobile devices, IoT devices and automotive systems, for example. It incorporates the capabilities of I²C and SPI, while offering several advantages over these protocols.

The main characteristics of I3C are:

- Significantly lower energy consumption and higher data rate than I²C, but backward compatible with it.
- 2-wire operation with a serial clock line (SCL) and a serial data line (SDA) simplifies data exchange at low pin count and reduced signal paths.
- Supports multiple devices on the same bus, enabling a multi-drop topology.
- Optional messaging modes (HDR-DDR or HDR ternary) with high data rate up to 33.33 Mbit/s.
- In-band interrupt support without requiring a separate interrupt line.
- Hot-join support for dynamic device attachment and detachment.
- Synchronous timing support and asynchronous time stamping.
- Broadcast transactions allow sending commands or data to all devices on the bus simultaneously.

The MXO 3 supports all operating speed modes: legacy I²C modes (see I²C characteristics), SDR standard mode, DDR mode and ternary mode.

### Data transfer

An I3C transaction consists of the following phases:

- **Start condition**: The controlling device sends a start condition to initiate the transaction.
- **Device addressing**: The controlling device sends the address of the responding device to select the target device.
- **Command**: The controlling device sends a command to the responding device, specifying the operation to be performed (e.g., read or write).
- **Data transfer**: The responding device responds with the requested data or sends data to the controlling device.
- **Stop condition**: The controlling device sends a stop condition to terminate the transaction.

The address format is the same as in I²C: 7 bits of address, 1 bit of RnW, and 1 bit of ACK/NACK. See Section 14.5.1, "About the I²C protocol", on page 566.

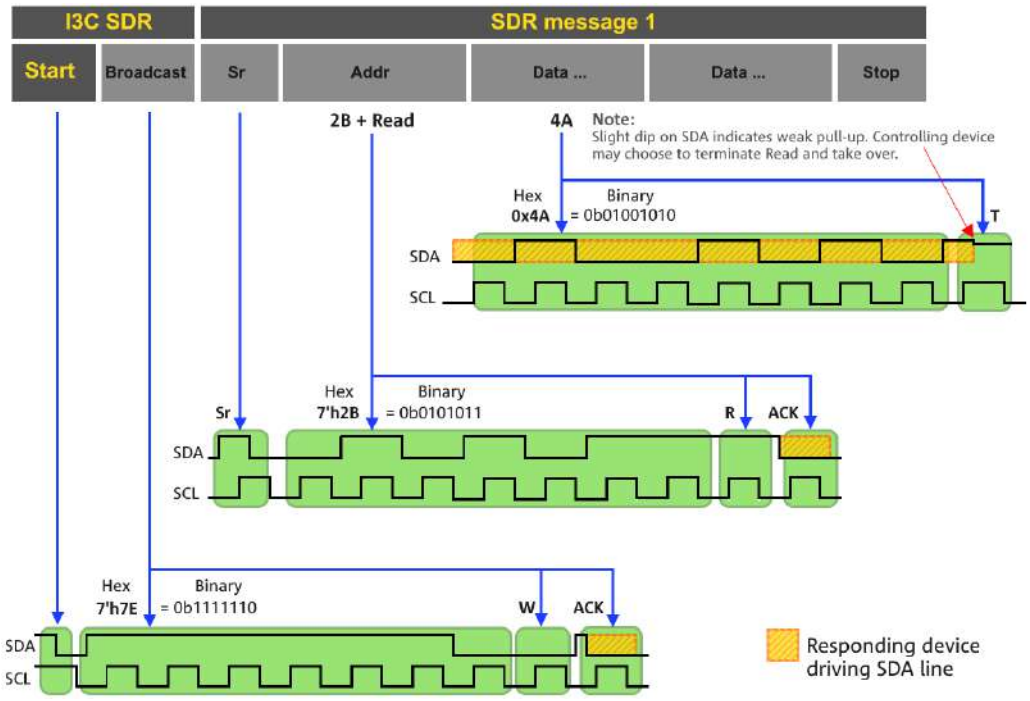



*Figure 14-18: Example of using I3C coding SDR*

### Trigger

The MXO 3 can trigger on various parts of I3C messages. You must connect the data and clock lines to the input channels. Triggering on math and reference waveforms is not possible.

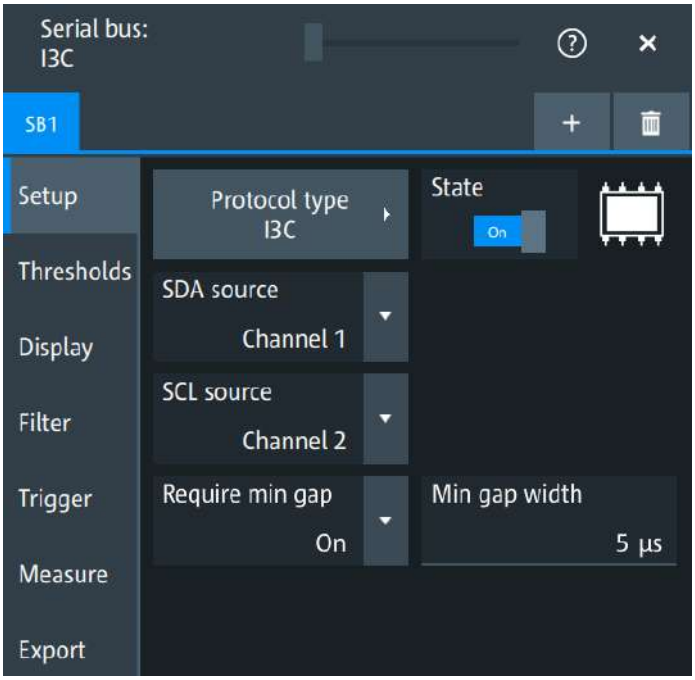
You can trigger on:

- Start or stop condition
- Repeated start condition
- Transfer direction (read or write)
- Bytes with missing acknowledge bit
- Specific target address or address range
- Specific data pattern in the message

## 14.6.2 I3C configuration

### 14.6.2.1 I3C configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "I3C" > "Setup".




Make sure that the tab of the correct serial bus is selected.

**Protocol type**
Selects the protocol type to be decoded.

Remote command:
`SBUS<sb>:TYPE` on page 1362

**State**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
`SBUS<sb>[:STATe]` on page 1362

**SDA source**
Sets the source channel to which the data line is connected.

Remote command:
`SBUS<sb>:I3C:SDA:SOURce` on page 1501

**SCL source**
Selects the source channel to which the clock line is connected.

Remote command:
`SBUS<sb>:I3C:SCL:SOURce` on page 1500

**Require min gap**
If enabled, the application requires a minimum gap on the left-hand side of the screen before the 1st decoded frame. Enabling this feature can prevent errors due to truncation of an HDR-type frame.

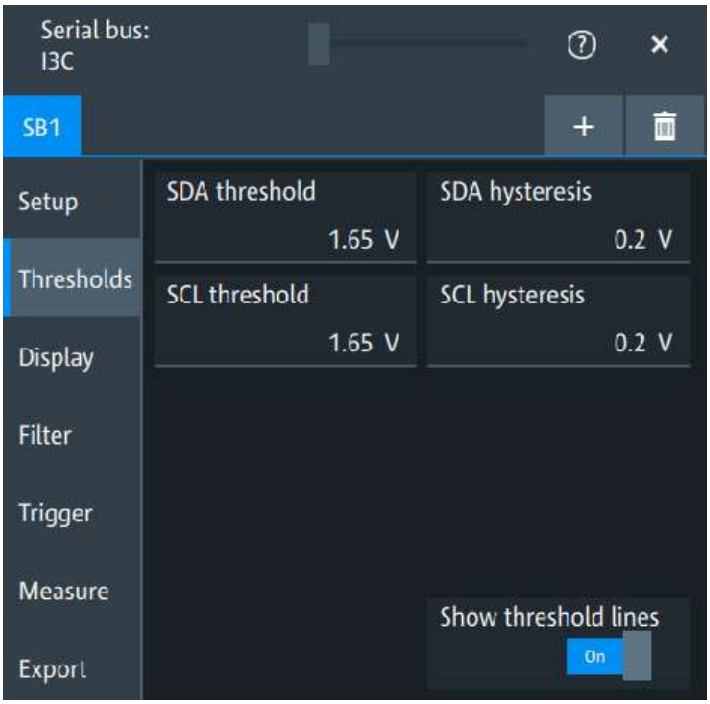
Remote command:
`SBUS<sb>:I3C:MINGap:SELect` on page 1502

**Min gap width**
Specifies the minimum gap width in units of time of the Require min gap, if enabled.

Remote command:
`SBUS<sb>:I3C:MINGap:WIDTh` on page 1502

### 14.6.2.2 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "I2C" > "Thresholds".




**SDA/SCL threshold, SDA/SCL hysteresis**
Set the threshold for the SDA and SCL channels. Enter the value directly in the fields.

Additional to the threshold, you can also set a hysteresis for each channel.

Remote command:
`SBUS<sb>:I3C:SCL:THReshold` on page 1501
`SBUS<sb>:I3C:SDA:THReshold` on page 1502
`SBUS<sb>:I3C:SCL:HYSTeresis` on page 1500
`SBUS<sb>:I3C:SDA:HYSTeresis` on page 1501

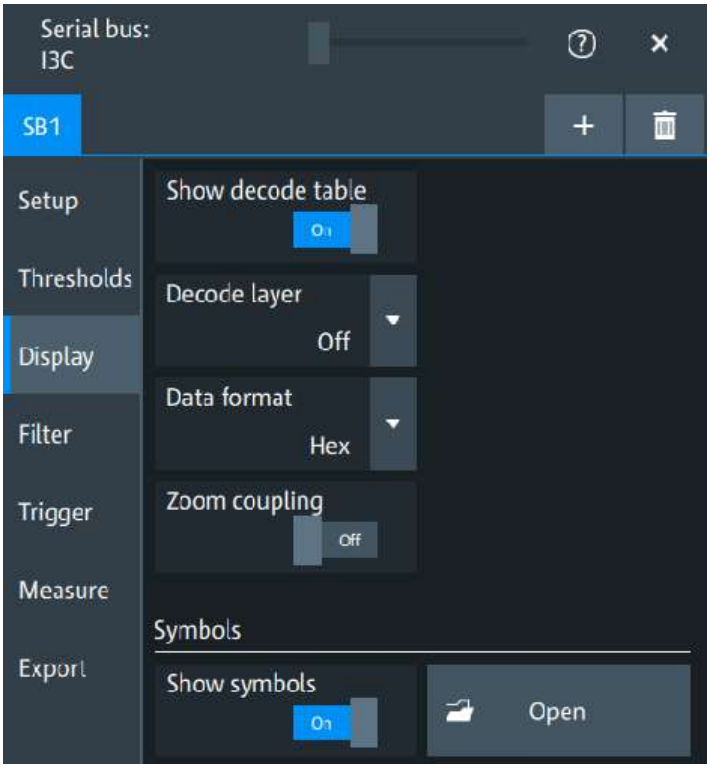
**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
SBUS<sb>:THReshold on page 1364

### 14.6.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.




**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

See "Decode layer" on page 493.

**Data format**
See "Data format" on page 493.

**Zoom coupling**
See "Zoom coupling" on page 494.

**Show symbols**
Symbol lists are protocol-specific. Label lists for I3C are available in CSV format.

For details, see I3C symbols.

Remote command:
SBUS<sb>:I3C:SYMBols on page 1503
SBUS<sb>:I3C:NEWList on page 1503

#### 14.6.2.4 I3C symbols

For the I3C protocol, you can upload symbol lists containing IDs and a symbolic name for each node. You can load a file in one of the supported formats. Symbol lists for I3C are available in CSV format.

For each address, an I3C label file contains three values (or optionally two values, see "Address type" on page 596):

- Address type, 7-bit or 10-bit long
- Address value
- Symbolic label: name of the address, specifying its function in the bus network.

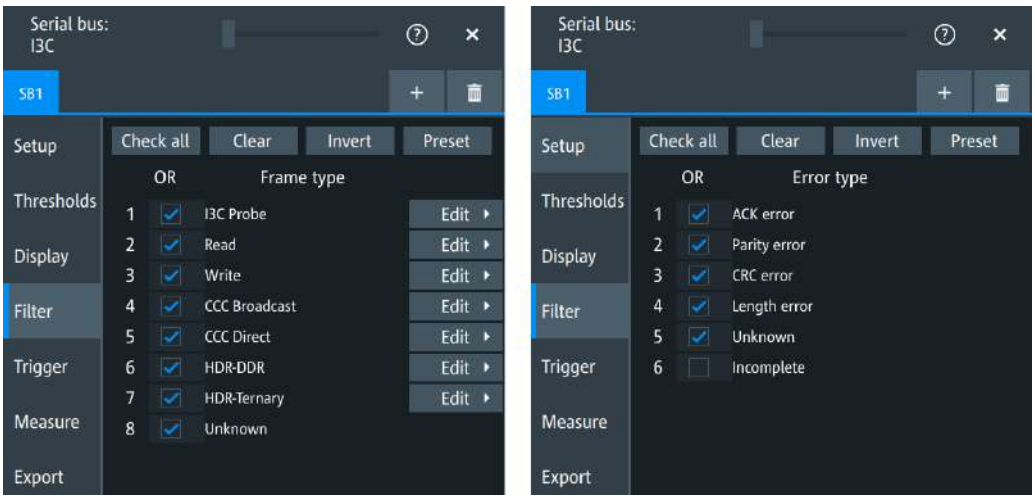
**Example: I3C CSV file**

```
# ----------------------------------------------------------------------------
@FILE_VERSION  = 1.00
@PROTOCOL_NAME = i3c
# ----------------------------------------------------------------------------
# Labels for I3C protocol
#   Column order: Identifier type, Identifier value, Label
# ----------------------------------------------------------------------------
7,0x1E,Voltage
7,38h,Pressure
7,2Ah,Temperature
7,16h,Speed
7,118,Acceleration
7,07h,HighSpeed_Master_0x3
7,51h,EEPROM
10,3A2h,DeviceSetup
10,1A3h,GatewayStatus
10,06Eh,LeftSensor
# ----------------------------------------------------------------------------
```

**Address type**
You can omit the address type, because I3C addresses are always 7-bit. Hence, the firmware also supports lines with only two entries: address value and symbolic label. For example, instead of "`7,0x1E,Voltage`", also "`0x1E,Voltage`" is valid.

### 14.6.3 I3C filter

Access: "Menu" > "Apps" > "Protocol" tab > "I3C" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

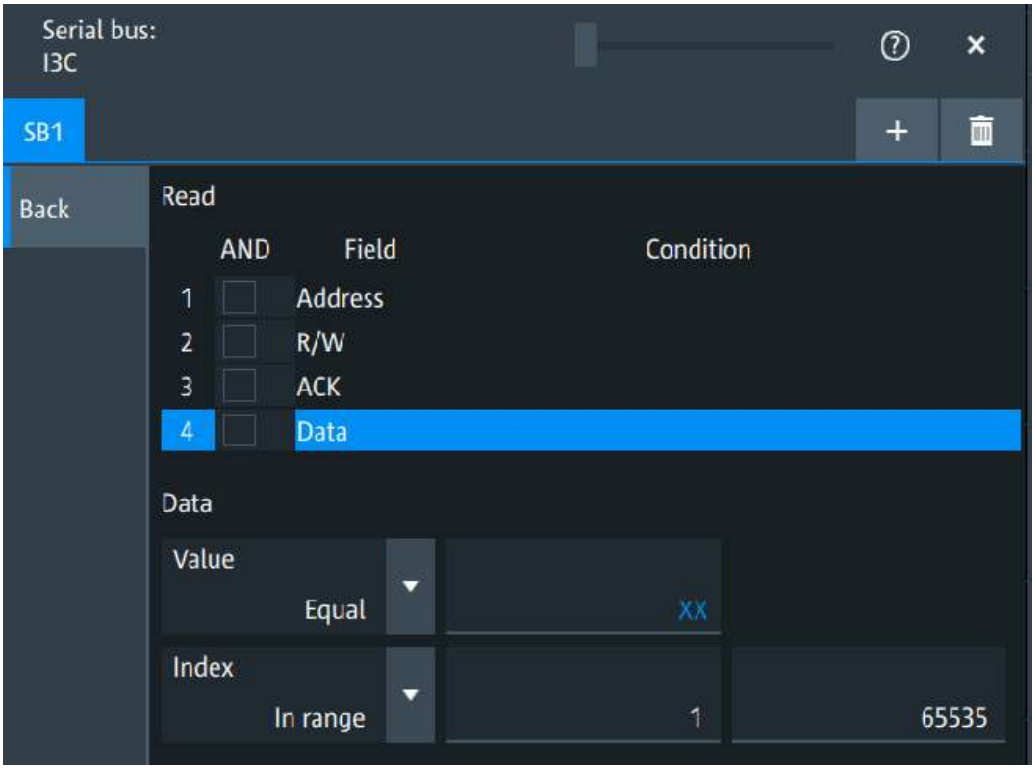
Remote command:
`SBUS<sb>:I3C:FILTer:CHKall` on page 1504
`SBUS<sb>:I3C:FILTer:CLR` on page 1505
`SBUS<sb>:I3C:FILTer:INVert` on page 1505
`SBUS<sb>:I3C:FILTer:RST` on page 1505

**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

Remote command:
`SBUS<sb>:I3C:FILTer:FRENable` on page 1505
`SBUS<sb>:I3C:FILTer:FRAMe<fr>:ENABle` on page 1505

**Edit**
Opens a dialog to define the details of the selected frame.




"Field" Enables the field type that you want to filter on for the selected frame. The available fields are "ACK", "Address", "CCC", "Command", "CRC", "Data", "Def Byte", "P", "R/W" and "Reserved".

Remote command:
`SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:ENABle` on page 1509
`SBUS<sb>:I3C:FILTer:FIENable` on page 1509

"Condition" Displays the value condition for the selected field.

Remote command:
`SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:BIT` on page 1508

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DMAX` on page 1506
`SBUS<sb>:I3C:FILTer:DMAX` on page 1506
`SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DMIN` on page 1506
`SBUS<sb>:I3C:FILTer:DMIN` on page 1506
`SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DOPerator` on page 1506
`SBUS<sb>:I3C:FILTer:DOPerator` on page 1506

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1507
SBUS<sb>:I3C:FILTer:IMAX on page 1507
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1508
SBUS<sb>:I3C:FILTer:IMIN on page 1508
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1508
SBUS<sb>:I3C:FILTer:IOPerator on page 1508

**Error type**
Enables filtering on the selected error type.

Remote command:
SBUS<sb>:I3C:FILTer:ERENable on page 1507
SBUS<sb>:I3C:FILTer:ERRor<n>:ENABle on page 1507

### 14.6.4 I3C format configuration

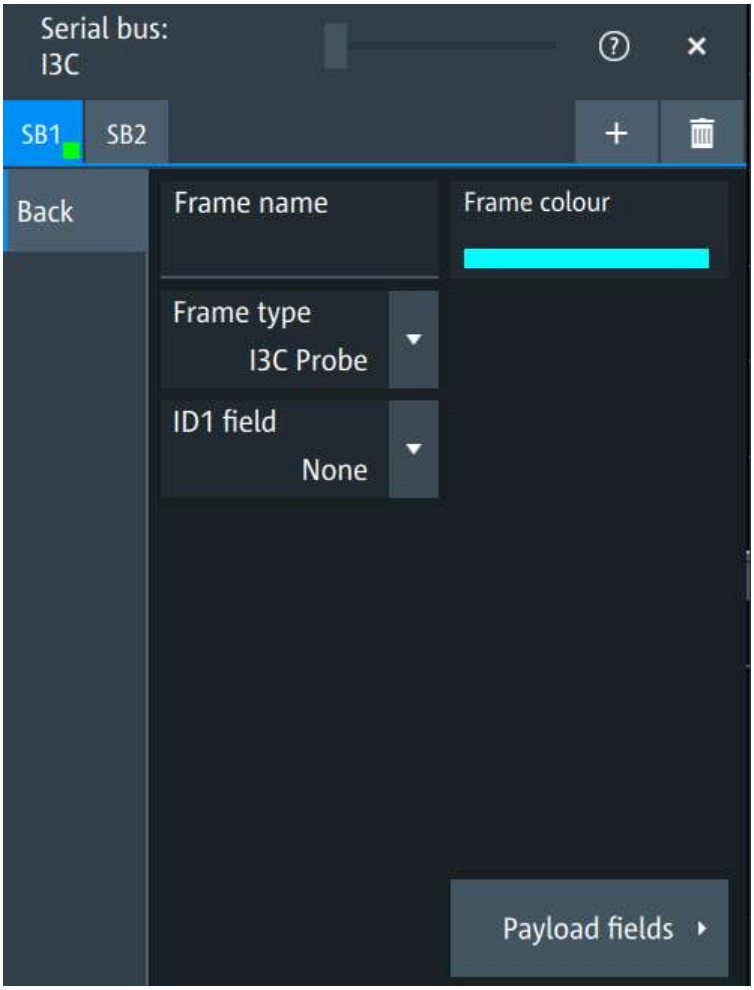
Access: "Menu" > "Apps" > "Protocol" tab > "I3C" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
SBUS<sb>:I3C:FORMat:SELect on page 1510

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:I3C:FORMat:ADDFrame on page 1510

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:I3C:FORMat:CLR on page 1511
SBUS<sb>:I3C:FORMat:FCOunt? on page 1511

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:I3C:FORMat:LOAD on page 1510

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:I3C:FORMat:SAVE on page 1510

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:NAME on page 1512

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:COLor on page 1512

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:TYPE on page 1512

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID1Field on page 1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID2Field on page 1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID3Field on page 1513

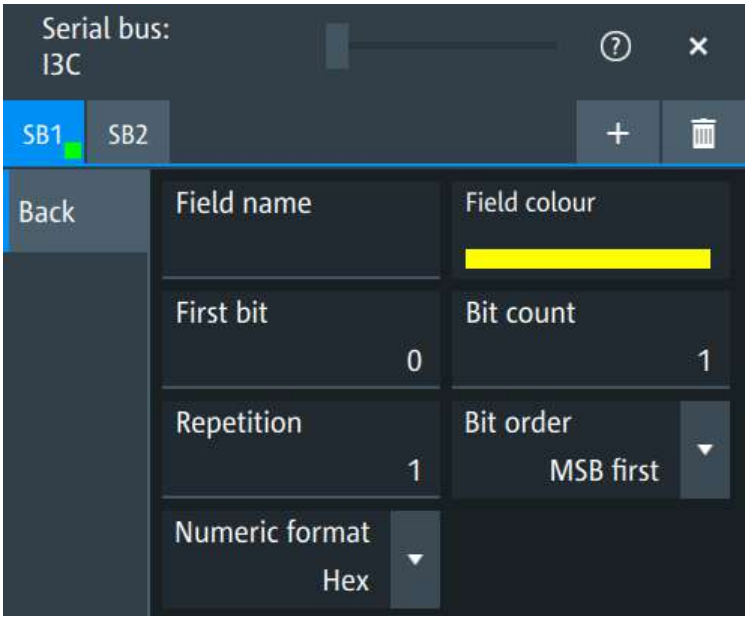
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID1Value on page 1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID2Value on page 1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID3Value on page 1513

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ADDField on page 1511

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1513

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1514

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1514

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.

- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  – Bit count = 2: truncated. Result: =00
  – Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  – Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  – Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1514

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
`SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:REPetition` on page 1515

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
`SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:BITorder` on page 1515

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:
- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  – "=1HL111000": valid (read as binary)
  – "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  – "=1010": valid (read as HEX, with a total of 16 bits)
  – "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)

– "=5X12": invalid
– "=1H33": invalid

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1516

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1516

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1517

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1517

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1517

**Parameter A,B**
Defines the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1518
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1518

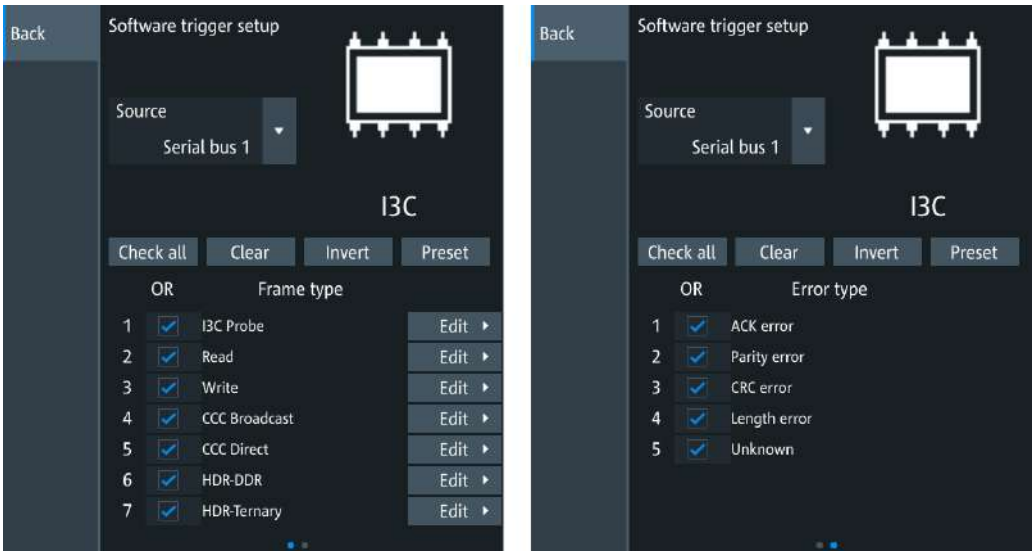
## 14.6.5 I3C software trigger

### 14.6.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.6.5.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "I3C" > "Trigger" tab > "Setup SB Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:I3C:CHKall` on page 1519
`TRIGger:SBSW:I3C:CLR` on page 1519
`TRIGger:SBSW:I3C:INVert` on page 1519
`TRIGger:SBSW:I3C:RST` on page 1520

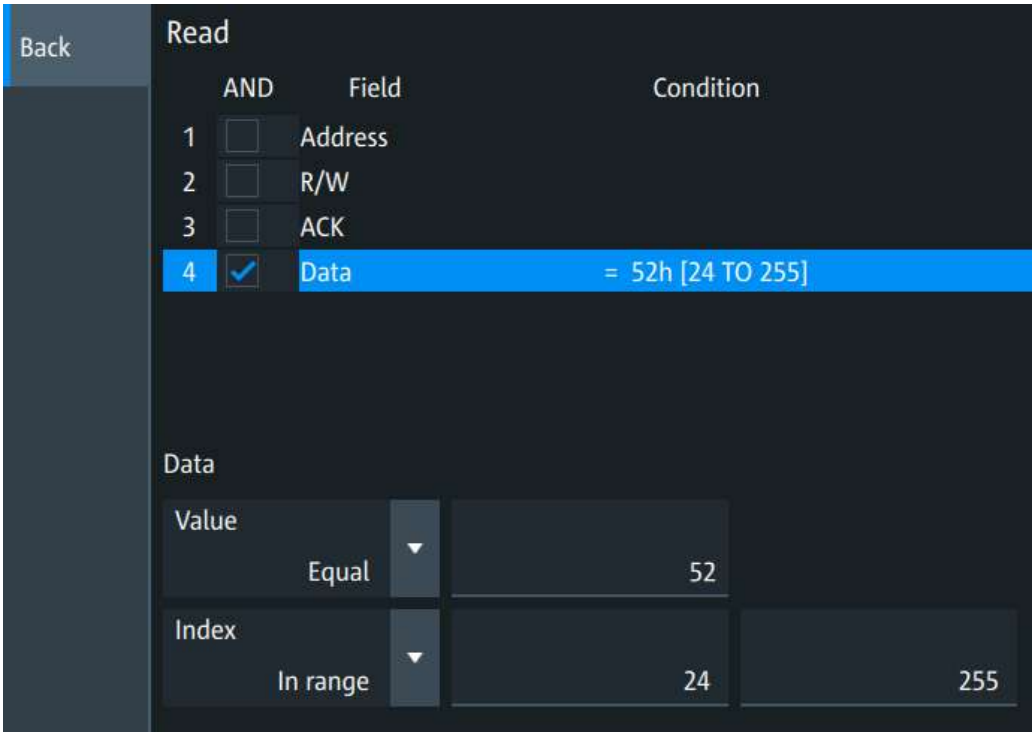
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: "CCC Broadcast", "CCC Direct", "HDR-DDR", "HDR-Ternary", "I3C Probe", "Read" and "Write".

Remote command:
`TRIGger:SBSW:I3C:FRENable` on page 1520
`TRIGger:SBSW:I3C:FRAMe<fr>:ENABle` on page 1520

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.




"Field" Enables the field type that you want to trigger on for the selected frame.
The available fields are "ACK", "Address", "CCC", "Command", "CRC", "Data", "Def Byte", "P", "R/W" and "Reserved".

Remote command:
`TRIGger:SBSW:I3C:FIENable` on page 1522
`TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:ENABle` on page 1522

"Condition" Displays the value condition for the selected field.

Remote command:
TRIGger:SBSW:I3C:BIT on page 1520
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:BIT on page 1520

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
TRIGger:SBSW:I3C:DMAX on page 1520
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DMAX on page 1520
TRIGger:SBSW:I3C:DMIN on page 1521
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DMIN on page 1521
TRIGger:SBSW:I3C:DOPerator on page 1521
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DOPerator on page 1521

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:I3C:IMAX on page 1522
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IMAX on page 1522
TRIGger:SBSW:I3C:IMIN on page 1522
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IMIN on page 1522
TRIGger:SBSW:I3C:IOPerator on page 1523
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IOPerator on page 1523

**Error type**
Enables triggering on the selected error type.

Remote command:
TRIGger:SBSW:I3C:ERENable on page 1523
TRIGger:SBSW:I3C:ERRor<m>:ENABle on page 1523

### 14.6.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.6.7 I3C decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

For more information on decoding I3C, you can also refer to the video, available on the Rohde & Schwarz YouTube channel: Decoding I2C with MXO Series Oscilloscope.

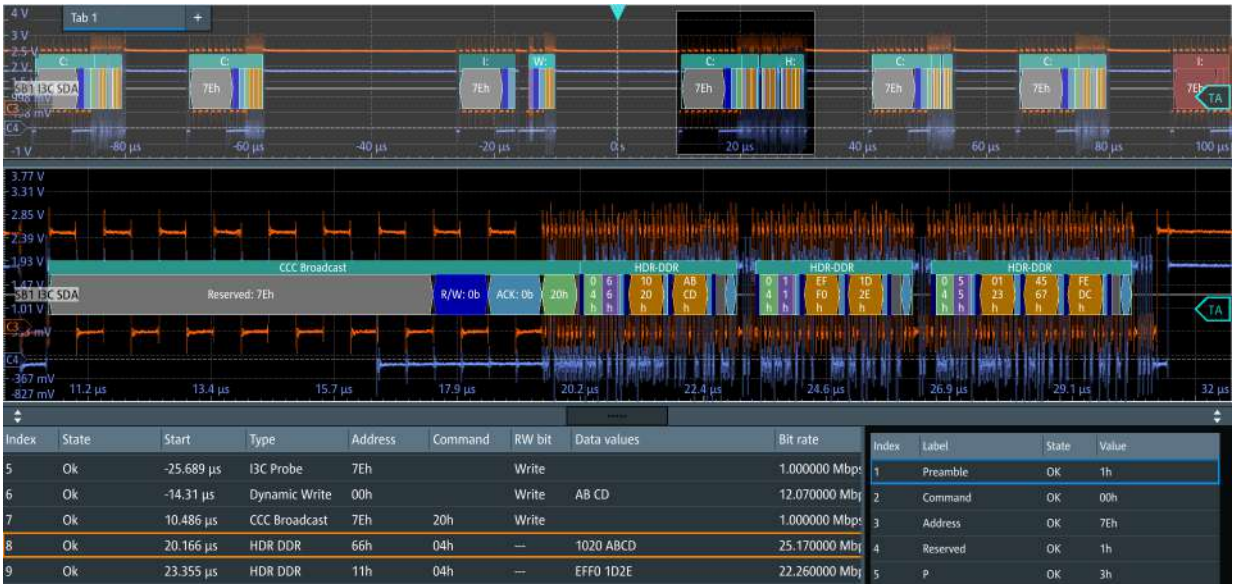



***Figure 14-19: Decoded I3C signal***

Purple = Address byte
Blue = R/W bit
Cyan = ACK bit
Red = NACK bit
Gold = Data byte
Turquoise = Write frame
Green = Read frame
Crimson red = Incomplete frame

The decode results table contains information about all decoded frames.

***Table 14-9: Content of the decode results table***

| Column | Description |
|---|---|
| Index | Index of the decoded frame |
| State | Overall state of the frame.<br>"Incomplete" indicates that the frame is not completely contained in the acquisition. Change the horizontal scale, or move the reference point to the left to get a longer acquisition. |
| Start | Time of frame start |
| Type | Frame type |
| Symbol | Symbolic label of the frame. Available only, if "Show symbols" is enabled in the "Display" tab and a symbols list is loaded. |
| Address | Value of the address field of the frame, if available |
| Command | Value of the command field of the frame, if available |
| RW bit | Value of the read/write bit |
| Data values | The actual data payload values. Select the data format in the "Display" tab. |
| Bit rate | Calculated bit rate for this frame |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

***Table 14-10: Content of the frame details table***

| Item | Description |
|---|---|
| Index | Number of the decoded field |
| Label | Label (name) of each word of the data field |
| State | State of each word of the data field, for example "OK" or "Incomplete" |
| Value | Value of each word of the data field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.6.5, "Decode results", on page 1524.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example of an I3C export file

```
Index,Start,Stop,State,Type,Address,Command,RWBit,CRC,Bit-rate
1,-4.38003e-05,-3.19206e-05,'OK','BRDC',7Eh,00h,'WRIT',---,1000000
2,3.45864e-08,8.9946e-06,'OK','PROB',7Eh,---,'WRIT',---,1000000
3,1.14144e-05,1.54345e-05,'OK','WRIT',00h,---,'WRIT',---,12070000
4,3.62096e-05,4.58891e-05,'OK','BRDC',7Eh,20h,'WRIT',---,1000000
5,4.58891e-05,4.87292e-05,'OK','HDDR',66h,04h,---,05h,25170000
6,4.9079e-05,5.19291e-05,'OK','HDDR',11h,04h,---,1Fh,22270000
7,5.2279e-05,5.59291e-05,'OK','HDDR',55h,04h,---,1Bh,22280000

Details frame 1
Index,Label,State,Value
1,'Data: 1','OK',12h
2,'Data: 2','OK',34h
3,'Data: 3','OK',56h

Details frame 3
Index,Label,State,Value
1,'Data: 1','OK',ABh
2,'Data: 2','OK',CDh

Details frame 5
Index,Label,State,Value
```

```
1,'Data: 1','OK',1020h
2,'Data: 2','OK',ABCDh

Details frame 6
Index,Label,State,Value
1,'Data: 1','OK',EFF0h
2,'Data: 2','OK',1D2Eh

Details frame 7
Index,Label,State,Value
1,'Data: 1','OK',0123h
2,'Data: 2','OK',4567h
3,'Data: 3','OK',FEDCh
```

## 14.7 UART (IC electronics, option R&S MXO3-K510)

The Universal Asynchronous Receiver/Transmitter UART converts a word of data into serial data, and vice versa. It is the base of many serial protocols such as RS-232.

For basic information on UART, you can also refer to a video, available on the Rohde & Schwarz YouTube channel: Understanding UART.

**Requirements**

For performing UART decode measurements, you need the following:

- MXO 3 with 2 available channels. The channels can be a combination of the following:
  - Analog channels (C1-C4)
  - Logic channels (D0-D15) (requires option R&S MXO3-B1)
- Two probes
- Option R&S MXO3-K510


- About the UART / RS-232 interface ..... 610
- UART configuration ..... 611
- UART filter ..... 615
- UART format configuration ..... 617
- UART hardware trigger ..... 623
- UART software trigger ..... 624
- Measure ..... 626
- UART decode results ..... 627
- Performing UART decoding ..... 628


### 14.7.1 About the UART / RS-232 interface

The UART uses only one line, or two lines for transmitter and receiver.

**Data transfer**

The data is transmitted in words, also referred to as symbols or characters. Each word consists of a start bit, several data bits, an optional parity bit, and one or more stop bits. Several words can form a package, or frame. The end of a package is marked with a reserved word or by a pause between two words.

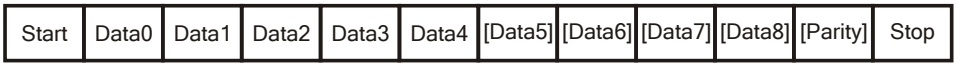



*Figure 14-20: Bit order in a UART word (symbol)*

- The start bit is a logic 0.
- The stop bits and the idle state are always logic 1.

The UART protocol has no clock for synchronization. The receiver synchronizes by the start and stop bits, and the bit rate that must be known to the receiver.

**Trigger**

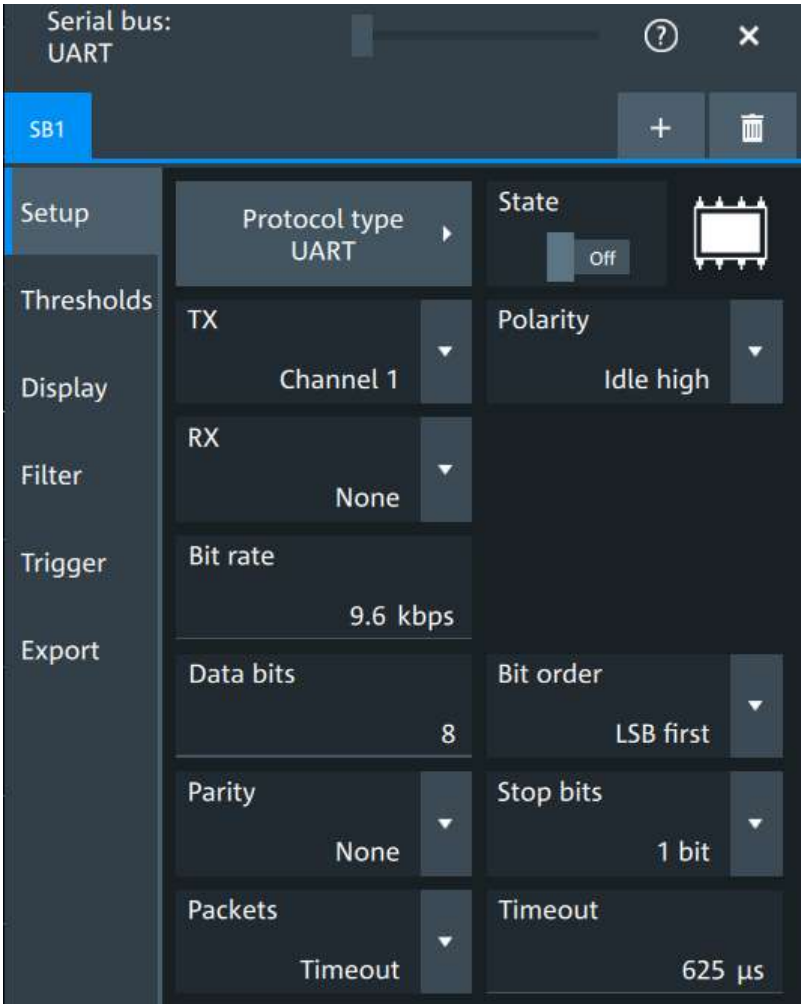
The MXO 3 can trigger on specified parts of UART serial signals:

- Start bit
- Packet start
- Parity errors, and breaks
- Stop errors
- A serial pattern at any or a specified position

## 14.7.2 UART configuration

### 14.7.2.1 UART configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "UART / RS232" > "Setup".




**Protocol type**
Selects the protocol type to be decoded.

Remote command:
SBUS<sb>:TYPE on page 1362

**State**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
SBUS<sb>[:STATe] on page 1362

**Source: Tx, Rx**
Select the input channels for the transmitter and receiver signals.

Do not combine a reference waveform with a channel or math waveform because the time correlation of these waveforms can differ.

Remote command:
SBUS<sb>:UART:TX:SOURce on page 1537
SBUS<sb>:UART:RX:SOURce on page 1535

**Bit order**
Selects the bit order, which determines if the data of the messages starts with MSB (most significant bit) or LSB (least significant bit).

Remote command:
SBUS<sb>:UART:BORDer on page 1534

**Polarity**
Defines the logic levels of the bus. The idle state corresponds to a logic 1. The start bit corresponds to a logic 0. "Idle high" (high=1) is used, for example, for control signals, while "Idle low" (low=1) is defined for data lines (RS-232).

Remote command:
SBUS<sb>:UART:POLarity on page 1535

**Bit rate**
Sets the number of transmitted bits per second.

Remote command:
SBUS<sb>:UART:BITRate on page 1533

**Stop bits**
Sets the number of stop bits: 1 or 1.5 or 2 stop bits are possible.

Remote command:
SBUS<sb>:UART:SBIT on page 1536

**Data bits**
Sets the number of data bits of a word in a range from 5 bits to 8 bits. If no parity bit is used, then 9 data bits are possible.

Remote command:
SBUS<sb>:UART:SSIZe on page 1536

**Parity**
Defines the optional parity bit that is used for error detection.

"None" No parity bit is used.
"Odd" The parity bit is set to "1" if the number of data bits set to "1" is even.
"Even" The parity bit is set to "1" if the number of data bits set to "1" is odd.
"Mark" The parity bit is always a logic 1.
"Space" The parity bit is always a logic 0.
"Don't care" The parity is ignored.

Remote command:
SBUS<sb>:UART:PARity on page 1534

**Packets**
Allows you to define packets of several words in the data stream.

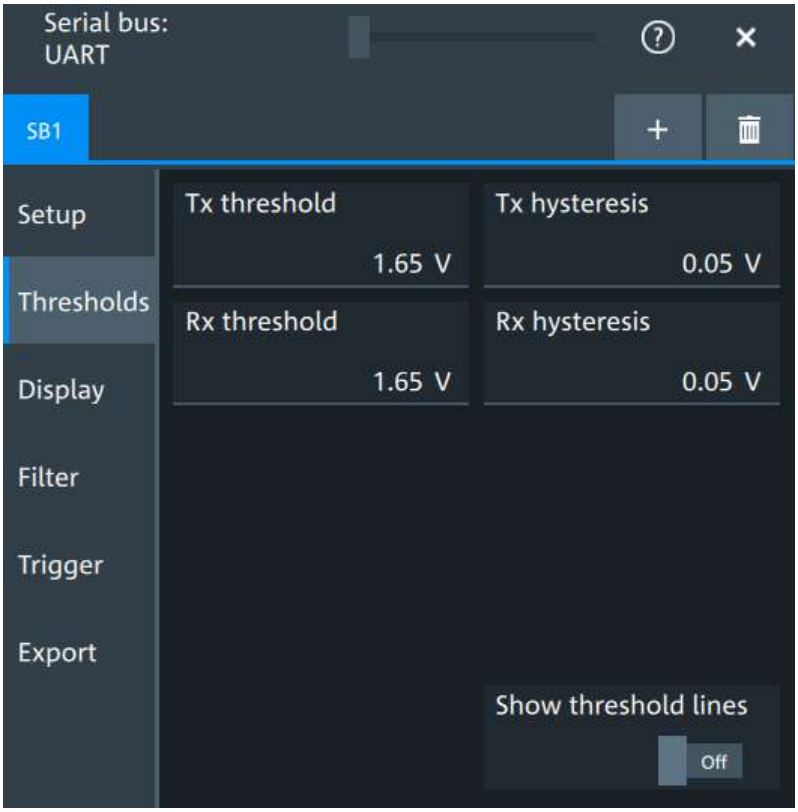
"None" Packets are not considered.

"Timeout" Defines a timeout between a stop bit and the next start bit. Enter the minimum time that marks the end of a packet.
A new packet starts with the first start bit after the timeout.

Remote command:
SBUS<sb>:UART:PACKets on page 1534
SBUS<sb>:UART:TOUT on page 1537

### 14.7.2.2 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "UART / RS232" > "Thresholds".




**Threshold**
Sets the threshold for the TX and RX channels. Enter the value directly in the fields.

Additional to the threshold, you can also set a hysteresis.

Remote command:
SBUS<sb>:UART:RX:HYSTeresis on page 1535
SBUS<sb>:UART:RX:THReshold on page 1536
SBUS<sb>:UART:TX:HYSTeresis on page 1537
SBUS<sb>:UART:TX:THReshold on page 1538

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

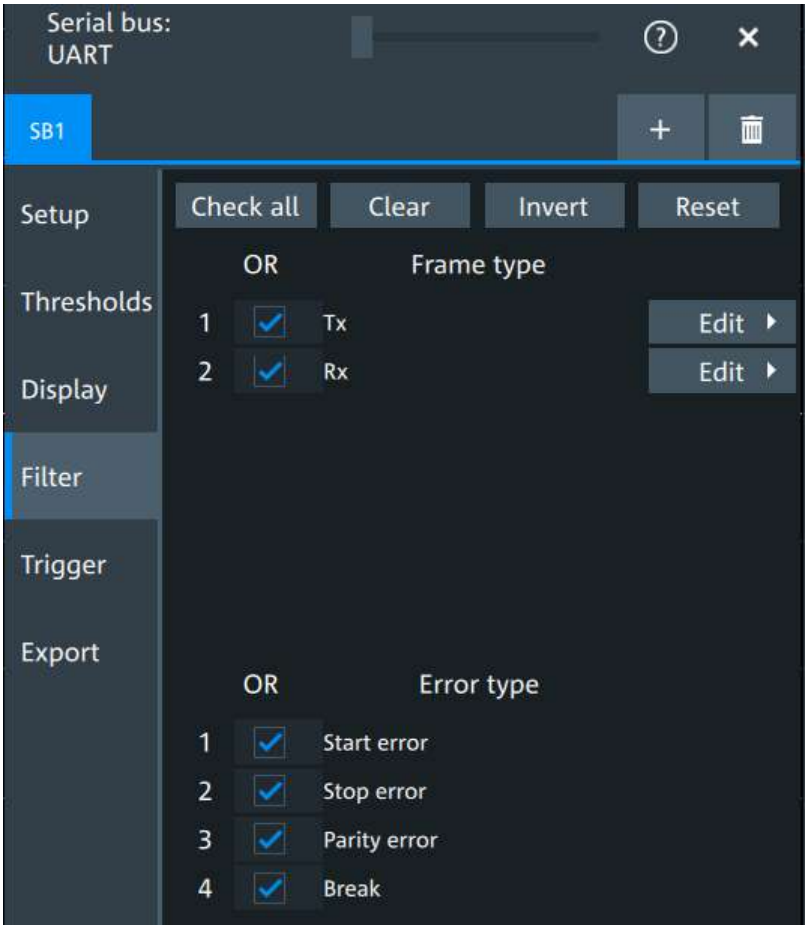
Remote command:
SBUS<sb>:THReshold on page 1364

#### 14.7.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

### 14.7.3 UART filter

Access:"Menu" > "Apps" > "Protocol" tab >"UART / RS232" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
`SBUS<sb>:UART:FILTer:CHKall` on page 1539
`SBUS<sb>:UART:FILTer:CLR` on page 1540
`SBUS<sb>:UART:FILTer:INVert` on page 1540
`SBUS<sb>:UART:FILTer:RST` on page 1540

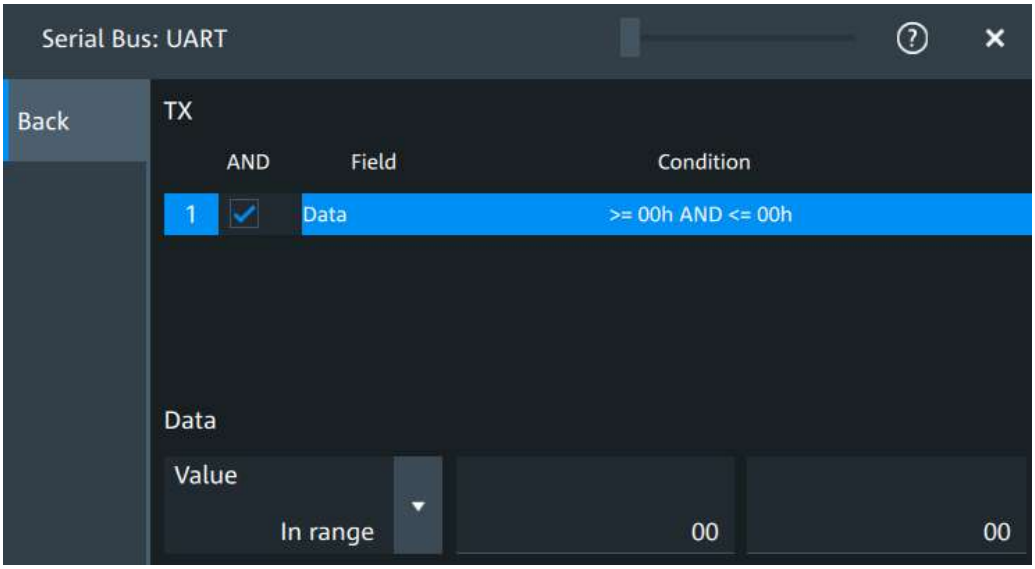
**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

The available frames are "Tx", "Rx".

Remote command:
`SBUS<sb>:UART:FILTer:FRAMe<fr>:ENABle` on page 1543
`SBUS<sb>:UART:FILTer:FRENable` on page 1543

**Edit**
Opens a dialog to define the details of the selected frame.




"Field" Enables the field type that you want to filter on for the selected frame. The available field is "Data".

Remote command:
`SBUS<sb>:UART:FILTer:FIENable` on page 1544
`SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:ENABle` on page 1544

"Condition" Displays the value condition for the selected field.

Remote command:
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:BIT on page 1543

"Data" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1543
SBUS<sb>:UART:FILTer:DMAX on page 1543
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1542
SBUS<sb>:UART:FILTer:DMIN on page 1542
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1542
SBUS<sb>:UART:FILTer:DOPerator on page 1542

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1541
SBUS<sb>:UART:FILTer:IMAX on page 1541
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1540
SBUS<sb>:UART:FILTer:IMIN on page 1540
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1541
SBUS<sb>:UART:FILTer:IOPerator on page 1541

**Error type**
Enables filtering on the selected error type.

Available error types are "Start error", "Stop error", "Parrity error", "Break".

Remote command:
SBUS<sb>:UART:FILTer:ERRor<n>:ENABle on page 1542

## 14.7.4 UART format configuration

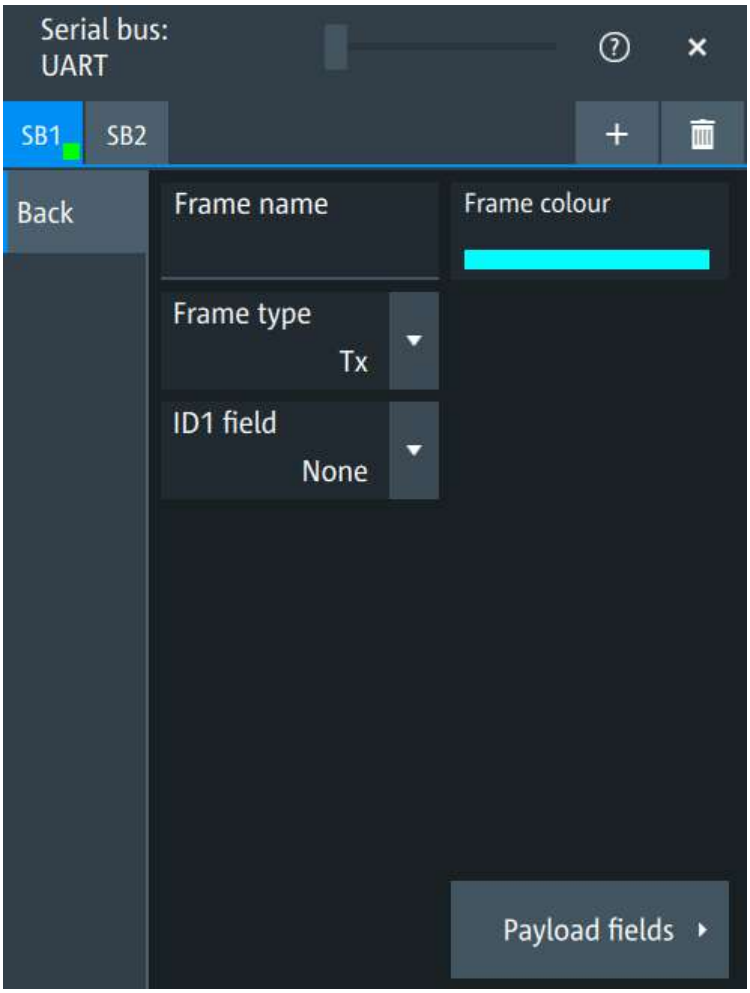
Access: "Menu" > "Apps" > "Protocol" tab > "UART / RS232" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
SBUS<sb>:UART:FORMat:SELect on page 1545

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:UART:FORMat:ADDFrame on page 1545

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:UART:FORMat:CLR on page 1546
SBUS<sb>:UART:FORMat:FCOunt? on page 1546

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:UART:FORMat:LOAD on page 1545

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:UART:FORMat:SAVE on page 1546

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:UART:FORMat:FRAMe<fr>:NAME on page 1547

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:UART:FORMat:FRAMe<fr>:COLor on page 1547

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:UART:FORMat:FRAMe<fr>:TYPE on page 1548

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID1Field on page 1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID2Field on page 1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID3Field on page 1548

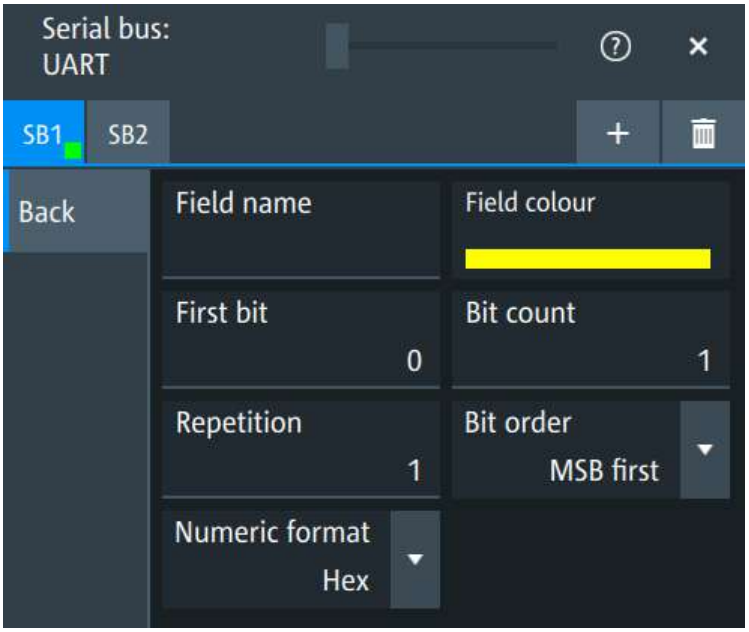
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID1Value on page 1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID2Value on page 1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID3Value on page 1548

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
SBUS<sb>:UART:FORMat:FRAMe<fr>:ADDField on page 1546

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1548

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1549

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1549

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.
- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  - Bit count = 2: truncated. Result: =00
  - Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  - Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  - Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1550

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
`SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:REPetition` on page 1550

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
`SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:BITorder` on page 1550

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:
- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.

- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  – "=1HL111000": valid (read as binary)
  – "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  – "=1010": valid (read as HEX, with a total of 16 bits)
  – "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)
  – "=5X12": invalid
  – "=1H33": invalid

Remote command:
`SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FORMat` on page 1551

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
`SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ADDValue` on page 1551

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
`SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue` on page 1552

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
`SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel` on page 1552

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
`SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:UNIT` on page 1552

**Parameter A,B**
Defines the parameters of the linear function.

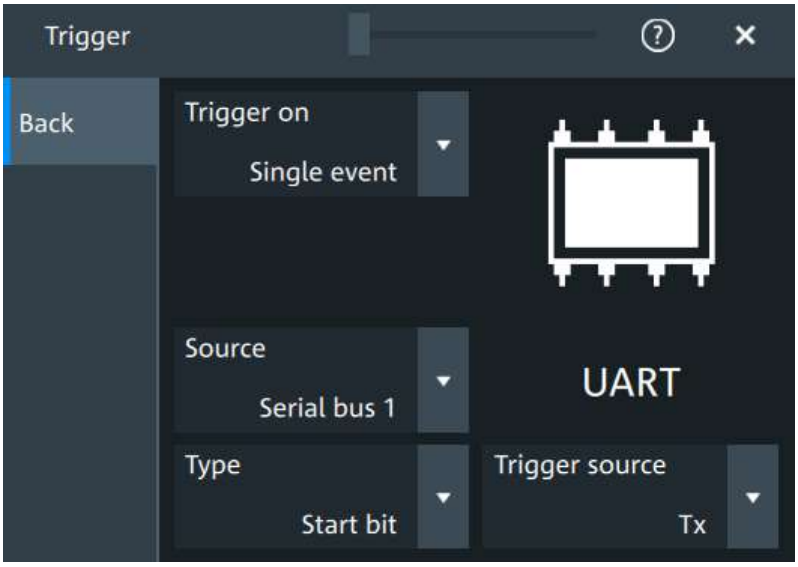
- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
`SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:PARA` on page 1553
`SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:PARB` on page 1553

### 14.7.5 UART hardware trigger

Access: "Menu" > "Apps" > "Protocol" tab > "UART / RS232" > "Trigger" tab > "Setup Hardware Trigger"




**Data conditions**
Specify the data conditions if the trigger type is set to "Data".

**Data ← Data conditions**
Selects the operator for the "Data" pattern.

Remote command:
`TRIGger:SBHW:UART:FCONdition` on page 1554

**Value ← Data conditions**
Specifies the data pattern to be found on the specified trigger source. Enter the words in MSB first bit order.

Remote command:
`TRIGger:SBHW:UART:DATA` on page 1553

**Position ← Data conditions**
Sets the number of words before the first word of interest. These offset words are ignored.

The setting is available if packet detection is enabled in the protocol configuration.

Remote command:
`TRIGger:SBHW:UART:DPOSition` on page 1554

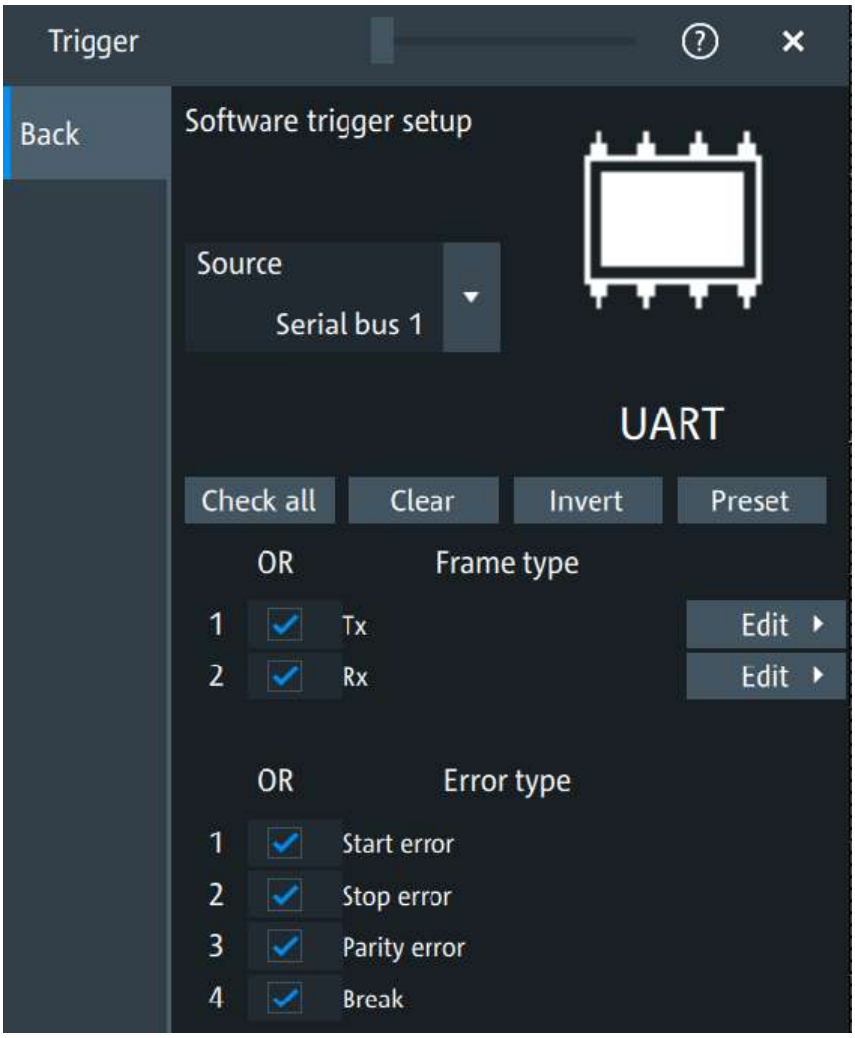
## 14.7.6 UART software trigger

### 14.7.6.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.7.6.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "UART / RS232" > "Trigger" tab > "Setup Software Trigger" > "Setup SB Trigger".




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

| | |
|---|---|
| "Check all" | Enables the software trigger for all available frames and error types. |
| "Clear" | Disables the software trigger for all available frames and error types. |
| "Invert" | Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa. |
| "Preset" | Presets the state of the selected frames and error types for the software trigger. |

Remote command:
`TRIGger:SBSW:UART:CHKall` on page 1556
`TRIGger:SBSW:UART:CLR` on page 1556
`TRIGger:SBSW:UART:INVert` on page 1556
`TRIGger:SBSW:UART:RST` on page 1556

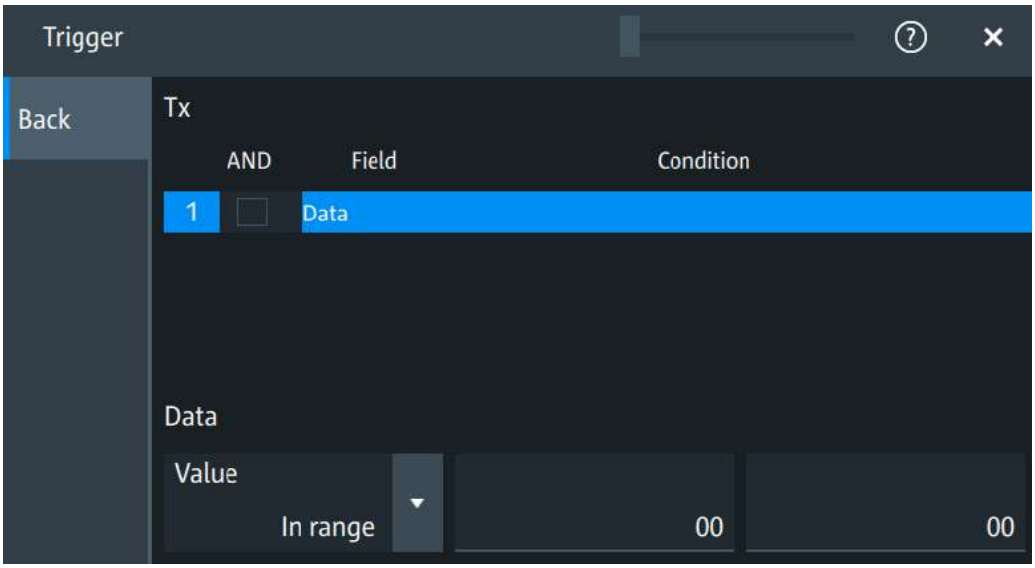
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: "Tx" and "Rx".

Remote command:
`TRIGger:SBSW:UART:FRAMe<fr>:ENABle` on page 1556
`TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:ENABle` on page 1558

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.



"Field" Enables the field type that you want to trigger on for the selected frame.
The available field is "Data".

Remote command:
TRIGger:SBSW:UART:FIENable on page 1558
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:ENABle on page 1558

"Condition" Displays the value condition for the selected field.

Remote command:
TRIGger:SBSW:UART:BIT on page 1557
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:BIT on page 1557

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
TRIGger:SBSW:UART:DMAX on page 1557
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DMAX on page 1557
TRIGger:SBSW:UART:DMIN on page 1557
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DMIN on page 1557
TRIGger:SBSW:UART:DOPerator on page 1558
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DOPerator on page 1558

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:UART:IMAX on page 1559
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IMAX on page 1559
TRIGger:SBSW:UART:IMIN on page 1559
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IMIN on page 1559
TRIGger:SBSW:UART:IOPerator on page 1559
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IOPerator on page 1559

**Error type**
Enables triggering on the selected error type.

The following error types are available: "Start error", "Stop error", "Parity error", "Break".

Remote command:
TRIGger:SBSW:UART:ERENable on page 1560
TRIGger:SBSW:UART:ERRor<m>:ENABle on page 1560

### 14.7.7 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.7.8 UART decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

For more information on decoding UART, you can also refer to the video, available on the Rohde & Schwarz YouTube channel: Decoding UART with MXO Series Oscilloscopes.




*Figure 14-21: Decoded UART signal*

The decode results table contains information about all decoded frames.

*Table 14-11: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Number of the decoded frames |
| State | Decoding state of the word.<br>"Insuffcient waveform length" indicates that the word is not completely contained in the acquisition. Change the horizontal scale, or move the reference point to the left to get a longer acquisition. |
| Start | Time of the word start (start bit) |
| Stop | Time of the word stop (stop bit) |
| Tx value | Value of the Tx word. Select the data format in the "Display" tab. |
| Rx value | Value of the Rx word. Select the data format in the "Display" tab. |
| Data rate | Value of the data rate |

**Data format**

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

**Remote commands**

Remote commands to retrieve decode results are described in Section 18.18.7.6, "Decode results", on page 1561.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

**Example UART export file**

```
Index,Start,Stop,Source,State,Tx value,Rx value,Bit rate
1,-0.0629961,-0.0629531,'TX','OK',33h,---,256000
2,-0.0629453,-0.0629023,'RX','OK',---,F5h,256000
3,-0.0628945,-0.0628516,'TX','STER',C8h,---,256000
4,-0.0628438,-0.0628008,'RX','SPER',---,28h,256000
5,-0.062793,-0.06275,'TX','PRER',A8h,---,256000
```

## 14.7.9 Performing UART decoding

This section explains step by step how to configure and decode the UART bus.

For more information on decoding UART, you can also refer to the video, available on the Rohde & Schwarz YouTube channel: Decoding UART with MXO Series Oscilloscopes.

### 14.7.9.1 Configuring UART signals

For configuration, assign the lines to the input channels, and define the active states and the logical thresholds.

For details on configuration settings, see UART configuration.

1. Tap the "+ Bus" activator in the bottom right of the screen.




2. Select the tab of the bus that you want to set up, for example "SB1".
3. Tap the "Protocol type" button. Select the protocol: "UART".
4. Tap on "State" to enable the decoding.

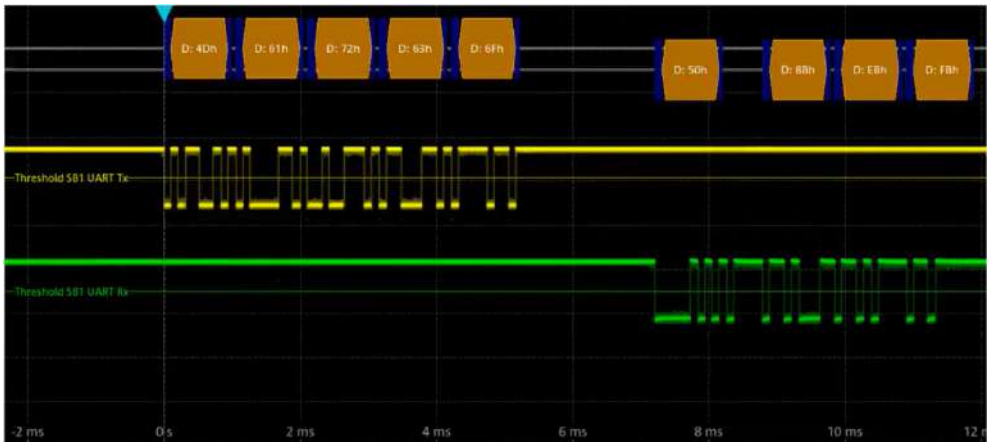
An SPI shortcut appears at the bottom left of the screen. Tap on the shortcut for a quick access to the "UART / RS232" dialog settings.




5. Select the correct channels for each active line: "Tx" and "Rx".
6. Set the "Bit rate".
7. Set the "Data bits". Typical value is 7 or 8.
8. Set the "Stop bits". Typical value is 1.
9. Set the "Parity" to "None", "Odd" or "Even".
10. Set "Bit order" to "MSB" or "LSB first".
11. Set the "Packets".
12. Set the "Timeout".
13. Set the "Polarity" to "Idle high" or "Idle low".
14. Check that the signals are on the screen.
    If not try adjusting the vertical and horizontal settings.




15. Set the logical thresholds:
    a) Tap the "Thresholds" tab.
    b) For each line, set the threshold value. A typical value is 2.5 V.
    c) If necessary, set the hysteresis value for the wires.
    d) Optionally, tap on "Show threshold lines".




### 14.7.9.2 Exporting decode results

Once a decode has been completed, you can generate an export of all decode results.

Prerequisites: A bus is configured for the UART and decoded.

1. Tap on the "UART / RS232" shortcut.
2. Tap on the "Export" tab.
3. If necessary, enable "Include details".
4. If necessary, enable "Include timing".
5. Select the "File type".

   An export file is saved in the selected directory.

The UART export files contain the following fields:

- Index
- Start
- Stop
- Source
- State
- TX value
- RX value
- Bit rate

**Example UART export file**

```
Index,Start,Stop,Source,State,Tx value,Rx value,Bit rate
1,-0.0629961,-0.0629531,'TX','OK',33h,---,256000
2,-0.0629453,-0.0629023,'RX','OK',---,F5h,256000
3,-0.0628945,-0.0628516,'TX','STER',C8h,---,256000
4,-0.0628438,-0.0628008,'RX','SPER',---,28h,256000
5,-0.062793,-0.06275,'TX','PRER',A8h,---,256000
```

# 14.8 NRZ clocked & unclocked (option R&S MXO3-K510)

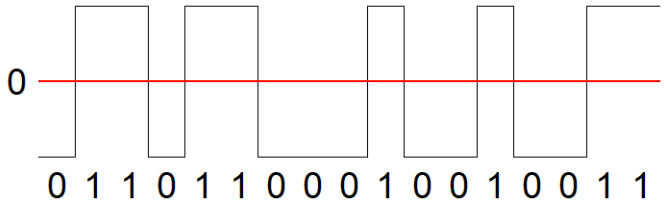
NRZ is a communication protocol for triggering at data rates up to 50 Mbit/s. It is suited for fast eye-diagram mask testing.

The NRZ protocol can either be **clocked** or **unclocked**. NRZ unclocked signals require a user-defined bit rate and gap time setting for triggering and decoding.


- NRZ basics ..... 631
- NRZ configuration ..... 631
- NRZ filter ..... 635
- NRZ format configuration ..... 639
- NRZ software trigger ..... 646
- Measure ..... 650
- NRZ decode results ..... 650


## 14.8.1 NRZ basics

In this line coding scheme, NRZ stands for "non-return-to-zero": The logical 1 is represented by a high voltage level, and the logical 0 is represented by a low voltage level:




The voltage remains at the same level for the entire bit period, without returning to a zero voltage level between bits. The voltage level changes only when the data bit changes.

Therefore, NRZ is an efficient protocol for high-speed serial data transmission. Its simplicity and low overhead make it a good choice for applications where data integrity and reliability are critical. It requires only half the bandwidth of the related Manchester code, which uses phase-shift keying.

## 14.8.2 NRZ configuration

### 14.8.2.1 NRZ configuration settings

Access:

- "Menu" > "Apps" > "Protocol" tab > "NRZ Clocked" > "Setup".
- "Menu" > "Apps" > "Protocol" tab > "NRZ Unclocked" > "Setup".

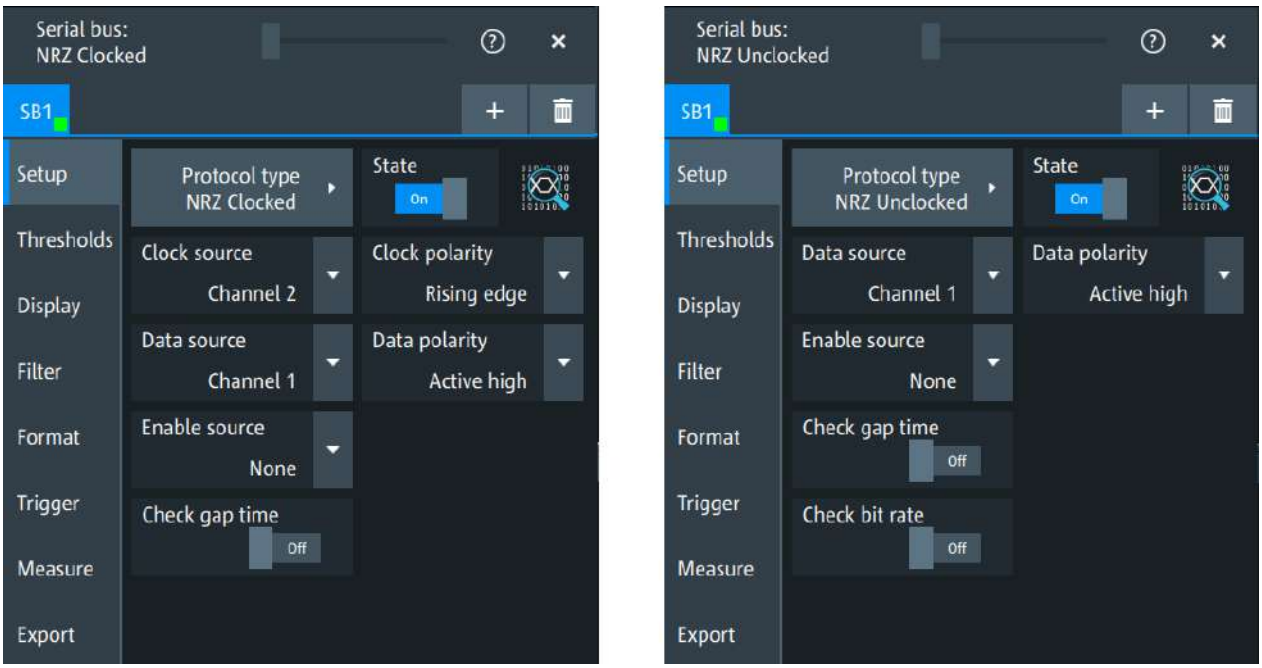



*Figure 14-22: Setup dialog, left: clocked version, right: unclocked version of NRZ*

**Data source**
Selects the source for the data signal.

Remote command:
`SBUS<sb>:NRZC:DATA:SOURce` on page 1567
`SBUS<sb>:NRZU:DATA:SOURce` on page 1596

**Clock source**
Available only for "NRZ Clocked".

Selects the source for the clock signal.

Remote command:
`SBUS<sb>:NRZC:CLK:SOURce` on page 1566

**Data polarity**
Selects the polarity for the data signal.

Remote command:
`SBUS<sb>:NRZC:DATA:POLarity` on page 1567
`SBUS<sb>:NRZU:DATA:POLarity` on page 1596

**Clock polarity**
Available only for "NRZ Clocked".

Selects the polarity for the clock signal.

Remote command:
`SBUS<sb>:NRZC:CLK:POLarity` on page 1566

**Enable source**
Selects the source of the enable signal.

If you select any source other than "None", the "Enable polarity" becomes available, which selects the polarity of the enable signal.

Remote command:
`SBUS<sb>:NRZC:ENABle:SOURce` on page 1568
`SBUS<sb>:NRZC:ENABle:POLarity` on page 1568
`SBUS<sb>:NRZU:ENABle:SOURce` on page 1597
`SBUS<sb>:NRZU:ENABle:POLarity` on page 1597

**Check gap time**
Enables the detection of the minimum idle time between two frames during decoding. Also enables setting the "Min gap time". If you set it, the decoder considers any longer idle time a gap.

Remote command:
`SBUS<sb>:NRZC:MINGap:SELect` on page 1568
`SBUS<sb>:NRZU:MINGap:SELect` on page 1597
`SBUS<sb>:NRZC:MINGap:WIDTh` on page 1569
`SBUS<sb>:NRZU:MINGap:WIDTh` on page 1598

**Check bit rate**
Available only for "NRZ Unclocked".

Enables setting the "Bit rate".
- If you define a bit rate, it is used for triggering and decoding.
- If you disable this option, the decoder calculates the bit rate.

Remote command:
`SBUS<sb>:NRZU:BITRate:SELect` on page 1598
`SBUS<sb>:NRZU:BITRate:WIDTh` on page 1598

### 14.8.2.2 Thresholds

Access:
- "Menu" > "Apps" > "Protocol" tab > "NRZ Clocked" > "Thresholds"
- "Menu" > "Apps" > "Protocol" tab > "NRZ Unclocked" > "Thresholds"

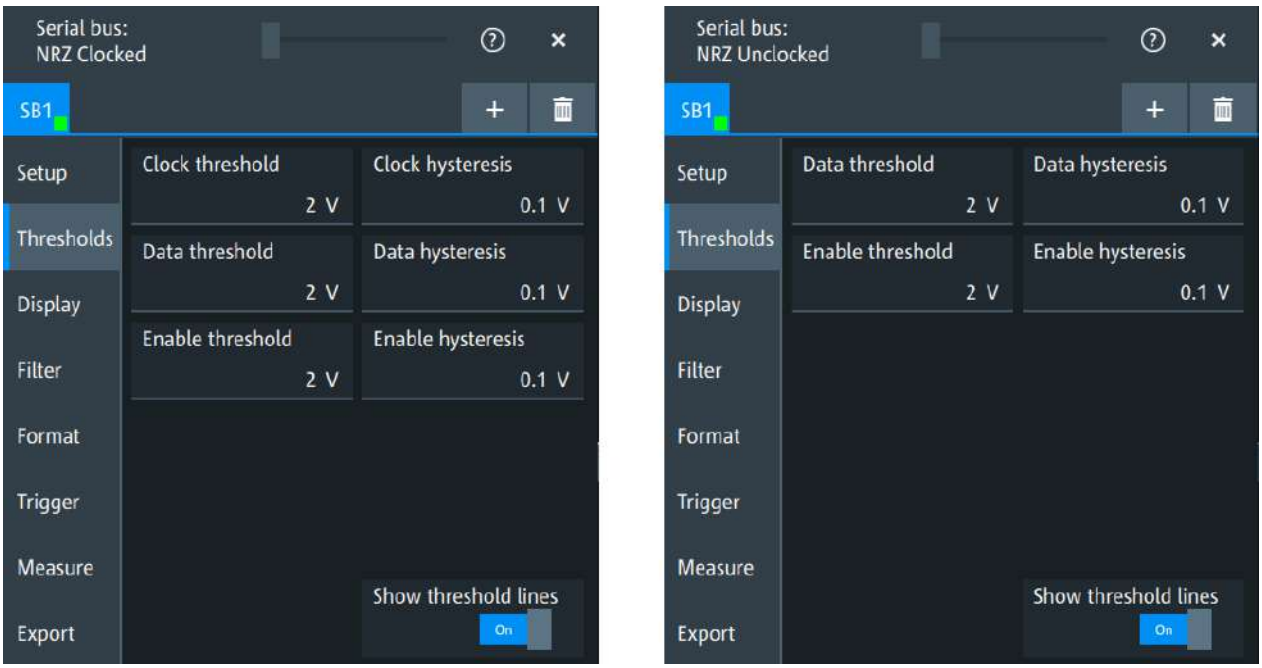



*Figure 14-23: Thresholds dialog, left: clocked version, right: unclocked version of NRZ*

Enter the values directly in the fields.

**Data threshold, Data hysteresis**
Sets the middle threshold and the hysteresis for the data source channel.

Remote command:
`SBUS<sb>:NRZU:DATA:THReshold` on page 1599
`SBUS<sb>:NRZU:DATA:HYSTeresis` on page 1599
`SBUS<sb>:NRZC:DATA:THReshold` on page 1569
`SBUS<sb>:NRZC:DATA:HYSTeresis` on page 1570

**Clock threshold, Clock hysteresis**
Available only for "NRZ Clocked".

Sets the middle threshold and the hysteresis for the clock source channel.

Remote command:
`SBUS<sb>:NRZC:CLK:THReshold` on page 1569
`SBUS<sb>:NRZC:CLK:HYSTeresis` on page 1569

**Enable threshold, Enable hysteresis**
Available only, if you have selected a source for the enable signal. See "Enable source" on page 633.

Sets the middle threshold and the hysteresis for the enable source channel.

Remote command:
`SBUS<sb>:NRZC:ENABle:THReshold` on page 1570
`SBUS<sb>:NRZC:ENABle:HYSTeresis` on page 1570

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
`SBUS<sb>:THReshold` on page 1364

#### 14.8.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

| | |
|---|---|
| "Off" | Disables the display of the decode layer. |
| "Edges" | Enables the display of all edges. |
| "Bits" | Enables the display of all encoded bits. |

**Data format**
See "Data format" on page 493.

#### 14.8.2.4 NRZ symbols

There is no symbolic translation for NRZ protocols, because of the custom frame definitions.

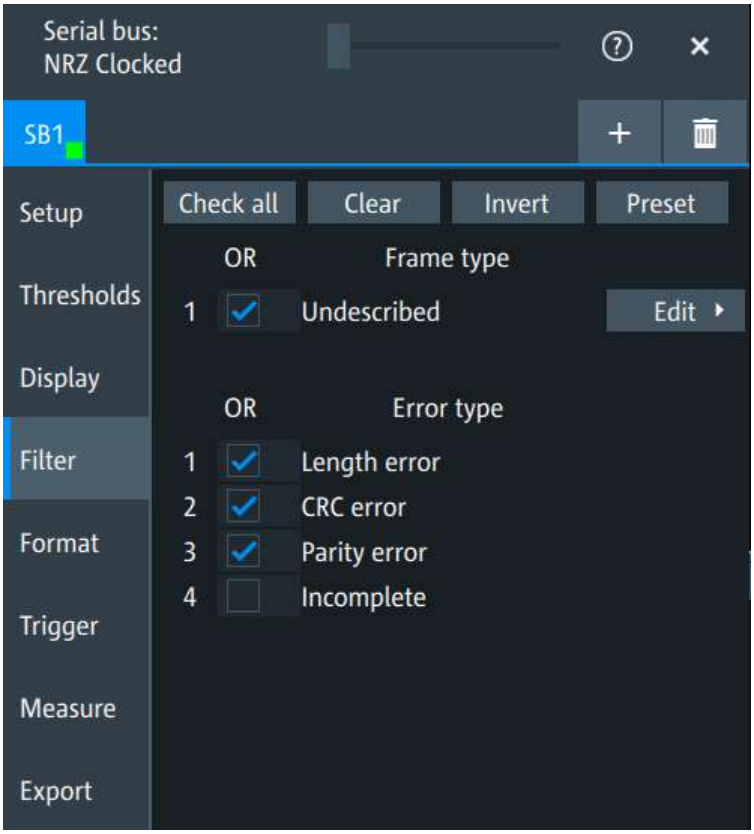
### 14.8.3 NRZ filter

Access:

- "Menu" > "Apps" > "Protocol" tab > "NRZ Clocked" > "Filter" tab
- "Menu" > "Apps" > "Protocol" tab > "NRZ Unclocked" > "Filter" tab

Although NRZ has a free format description according to section Section 14.8.4, "NRZ format configuration", on page 639, we provide a dynamic filter solution that can adapt to custom frames and fields in the decoded events.

In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.




**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
`SBUS<sb>:NRZC:FILTer:CHKall` on page 1572
`SBUS<sb>:NRZU:FILTer:CHKall` on page 1601
`SBUS<sb>:NRZC:FILTer:CLR` on page 1572
`SBUS<sb>:NRZU:FILTer:CLR` on page 1601
`SBUS<sb>:NRZC:FILTer:INVert` on page 1572
`SBUS<sb>:NRZU:FILTer:INVert` on page 1601
`SBUS<sb>:NRZC:FILTer:RST` on page 1572
`SBUS<sb>:NRZU:FILTer:RST` on page 1602

**Enable**
Enables the filtering on NRZ frames. Only the frames that match the selected filter conditions are displayed.

Remote command:
`SBUS<sb>:NRZC:FILTer:FRENable` on page 1575
`SBUS<sb>:NRZC:FILTer:FRAMe<fr>:ENABle` on page 1575
`SBUS<sb>:NRZU:FILTer:FRENable` on page 1604
`SBUS<sb>:NRZU:FILTer:FRAMe<fr>:ENABle` on page 1604

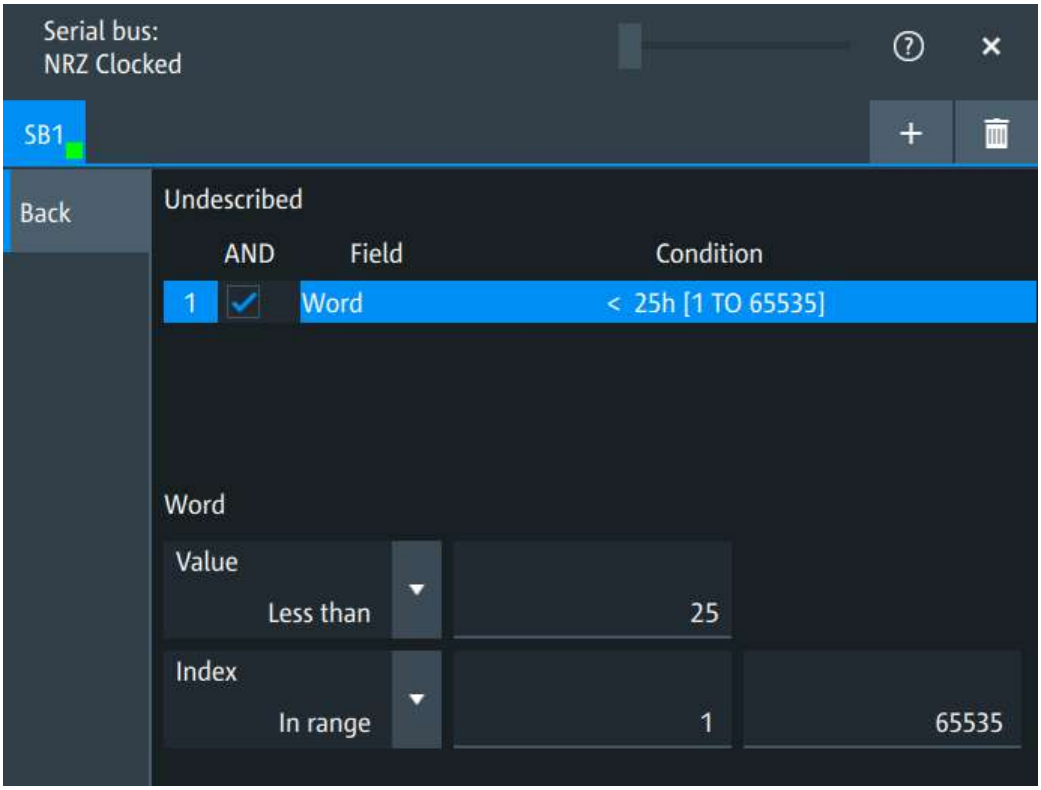
**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

The only frame type available by default is "Undescribed", because the NRZ protocols support custom coding.

However, the frame formats that you define appear in this list, too. See Section 14.8.4, "NRZ format configuration", on page 639.

**Edit**
Opens a dialog to define the details of the selected frame.



"Field" Enables the field type that you want to filter on for the selected frame. For the default "Undescribed" frame, the only available field type is "Word", which is the payload component of an undescribed frame. See "Frame type" on page 637.
However, the field formats that you define appear in this list, too. See Section 14.8.4, "NRZ format configuration", on page 639.

Remote command:
`SBUS<sb>:NRZC:FILTer:FIENable` on page 1574
`SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:ENABle` on page 1574
`SBUS<sb>:NRZU:FILTer:FIENable` on page 1604
`SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:ENABle` on page 1604

"Condition" Displays the value condition for the selected field.

Remote command:
`SBUS<sb>:NRZC:FILTer:BIT` on page 1572
`SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:BIT` on page 1572
`SBUS<sb>:NRZU:FILTer:BIT` on page 1602
`SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:BIT` on page 1602

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`SBUS<sb>:NRZC:FILTer:DMAX` on page 1573
`SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DMAX` on page 1573
`SBUS<sb>:NRZU:FILTer:DMAX` on page 1603
`SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DMAX` on page 1603
`SBUS<sb>:NRZC:FILTer:DMIN` on page 1574
`SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DMIN` on page 1574
`SBUS<sb>:NRZU:FILTer:DMIN` on page 1603
`SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DMIN` on page 1603
`SBUS<sb>:NRZC:FILTer:DOPerator` on page 1574
`SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DOPerator` on page 1574
`SBUS<sb>:NRZU:FILTer:DOPerator` on page 1603
`SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DOPerator` on page 1603

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:NRZC:FILTer:IMAX on page 1575
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1575
SBUS<sb>:NRZU:FILTer:IMAX on page 1604
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1604
SBUS<sb>:NRZC:FILTer:IMIN on page 1576
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1576
SBUS<sb>:NRZU:FILTer:IMIN on page 1605
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1605
SBUS<sb>:NRZC:FILTer:IOPerator on page 1576
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1576
SBUS<sb>:NRZU:FILTer:IOPerator on page 1605
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1605

**Error type**
Enables filtering on the selected error type.

The available error types are "Length error", "CRC error", "Parity error", "Incomplete".

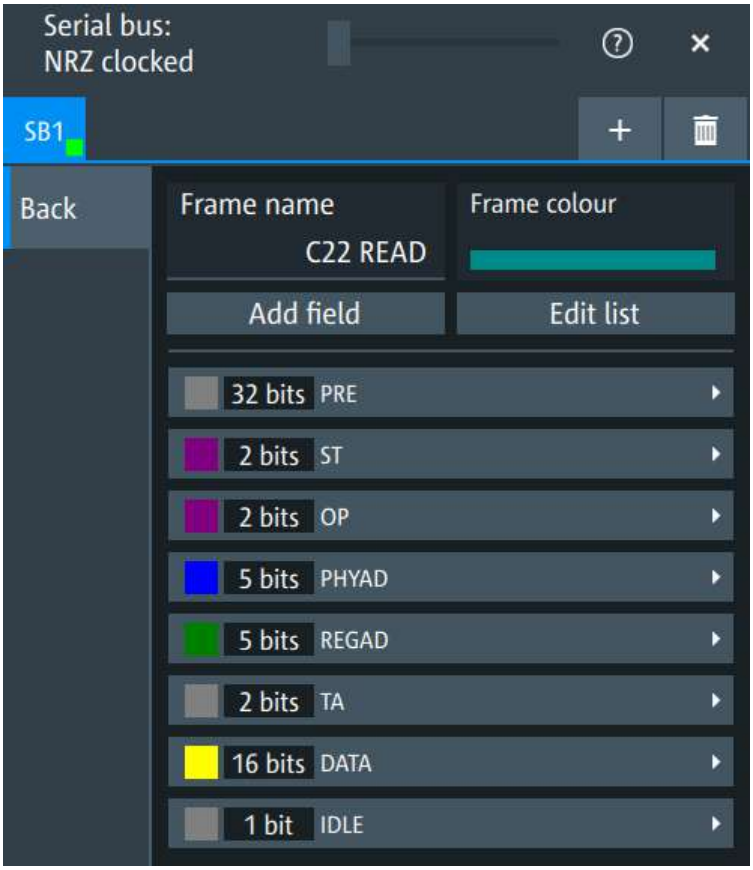
Remote command:
SBUS<sb>:NRZC:FILTer:ERENable on page 1573
SBUS<sb>:NRZC:FILTer:ERRor<n>:ENABle on page 1573
SBUS<sb>:NRZU:FILTer:ERENable on page 1602
SBUS<sb>:NRZU:FILTer:ERRor<n>:ENABle on page 1602

### 14.8.4 NRZ format configuration

Access: "Menu" > "Apps" > "Protocol" tab > "NRZ Clocked"or "NRZ Unclocked" > "Format".

See Section 14.1.5, "Format", on page 497.

**Add Frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:NRZC:FORMat:ADDFrame on page 1577
SBUS<sb>:NRZU:FORMat:ADDFrame on page 1606

**Edit List**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:NRZC:FORMat:CLR on page 1578
SBUS<sb>:NRZU:FORMat:CLR on page 1607
SBUS<sb>:NRZC:FORMat:FCOunt? on page 1578
SBUS<sb>:NRZU:FORMat:FCOunt? on page 1607

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:NRZC:FORMat:LOAD on page 1577
SBUS<sb>:NRZU:FORMat:LOAD on page 1606

**Save as**
Opens a dialog for saving your current list of frame formats to an `XML` file. Saved frame formats can support efficient and convenient working.

Remote command:
`SBUS<sb>:NRZC:FORMat:SAVE` on page 1577
`SBUS<sb>:NRZU:FORMat:SAVE` on page 1607

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

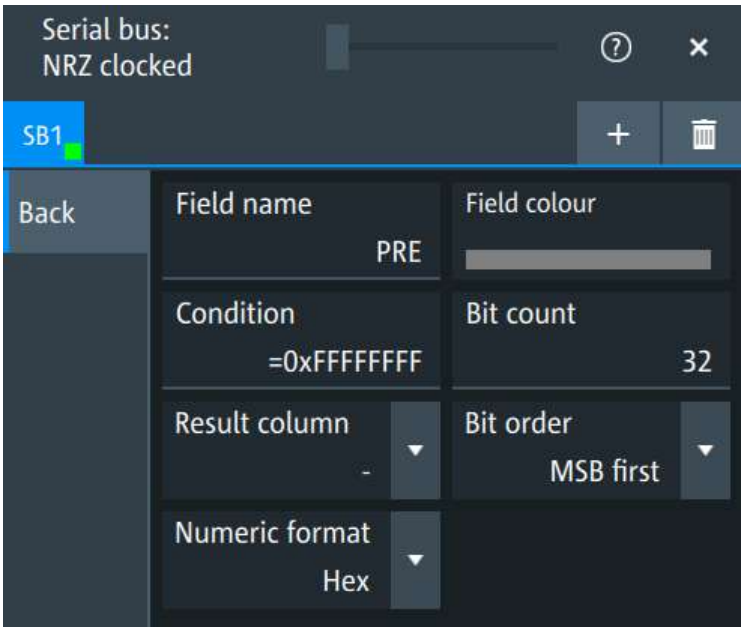
Remote command:
`SBUS<sb>:NRZC:FORMat:FRAMe<fr>:NAME` on page 1579
`SBUS<sb>:NRZU:FORMat:FRAMe<fr>:NAME` on page 1608

**Frame Color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
`SBUS<sb>:NRZC:FORMat:FRAMe<fr>:COLor` on page 1579
`SBUS<sb>:NRZU:FORMat:FRAMe<fr>:COLor` on page 1608

**Add Field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.



Remote command:
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:ADDField on page 1578
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:ADDField on page 1608

**Edit List**
Enables editing the field list. If enabled, you can delete a field description by clicking its "Delete" icon. In a list with several entries, you can move field descriptions up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLDCount? on page 1578
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLDCount? on page 1608

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1579
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1609

**Field Color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1580
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1609

**Condition**
Defines a rule or pattern that a field must match to help identify a frame and/or validate mandatory values, e.g., CRC, ID. The condition must match the field's "Numeric format", "Bit order", and "Bit length".

The following conditions are implemented:

"=" (equal / pattern match)

Defines a bit pattern that the field must match. Allowed characters depend on the field's "Format", "Bit order", and "Bits". In binary format, for example, valid characters are "1", "0", "H" (high), and "L" (low).

The behavior depends on whether the frame contains a variable-length array [ ]:

- **No [ ] in the frame**: All fields using = act as keys to identify the frame type. The frame matches only if all such fields match.
- **Frame contains one [ ]**: If there is a "Variable Length Array" field, then the equal operator has two different functionalities, depending on the position of the equal-operator field within the frame format configuration:
  - **Field is before the [ ]**: The condition acts as a key to identify a frame type and matches starting bits.
  - **Field is immediately after the [ ]**: The condition acts as an array delimiter and matches the bits right after the variable-length data to terminate the array.
    Note: If a = field is after the [ ] but not immediately after it, decode results are unpredictable.

"[ ]" (array / repetition)

Defines the number of permissible repetitions of the field.

- **Fixed-length array [n] (n > 0)**: Repeat the field n times. Effective length is n × "Bit count". Elements appear individually in results/ honeycomb under the same name and color.
- **Variable-length array [ ]**: Repeat the field 0 or more times, so the frame length is unspecified and ends via an end-of-frame condition, which can be an operator or gap.
  Constraints:
  - Only one variable-length array is supported per frame (processing limitation).
  - A delimiter field must follow immediately after [ ] (unless [ ] is the last field) so the decoder can terminate the array.
  - Variable-length array elements appear in details as Name: 1, Name: 2, etc. The selected field shows as Name[n] where n is the detected size.
  - Exception: Consecutive variable arrays can form a repeated structure, e.g. A[] + B[] → ABABAB... until the end condition has been detected.
  - Check functions can be combined in dynamic arrays, e.g., B[] can check parity relative to A[] by index.

"crc5usb(n-m)" (CRC check, 5-bit, USB polynomial)

Verifies a 5-bit CRC over fields in the index range n to m. For example, use "crc5usb(1-5)" to check fields 1 through 4. If the range includes an array, all elements in the array are included.
If the CRC check fails, the CRC field is marked "CRC error" (red), and the containing frame marked similarly unless a higher-priority error exists.

| | |
|---|---|
| "odd(n-m), even(n-m)" (parity check) | Checks odd/even parity over the index range n to m including the parity bit.<br>If the parity check fails, the parity field is marked "CRC error" (red), and the containing frame marked similarly unless a higher-priority error exists. |

Remote command:
`SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:CONDition` on page 1580
`SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:CONDition` on page 1610

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.
- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  - Bit count = 2: truncated. Result: =00
  - Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  - Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  - Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1580
`SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1609

**Result column**
Determines which field is displayed in which result column of the decode table.

- Selecting "-" (no index) means that the result is not displayed.
- Selecting "1", "2" or "3" means that the result is displayed in the 1st, 2nd or 3rd result column, that the decode table supports. Each of the three result columns has to be unique for each frame type. But to display unrelated information for different frame types, you can define different result columns.

Remote command:
`SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:CLMN` on page 1582
`SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:CLMN` on page 1611

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:BITorder on page 1582
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:BITorder on page 1611

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:

- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  – "=1HL111000": valid (read as binary)
  – "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  – "=1010": valid (read as HEX, with a total of 16 bits)
  – "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)
  – "=5X12": invalid
  – "=1H33": invalid

Remote command:
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1581
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1610

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1583
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1612

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1583
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1612

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1583
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1613

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1584
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1613

**Parameter A,B**
Defines the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1584
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1584
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1613
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1613

## 14.8.5 NRZ software trigger

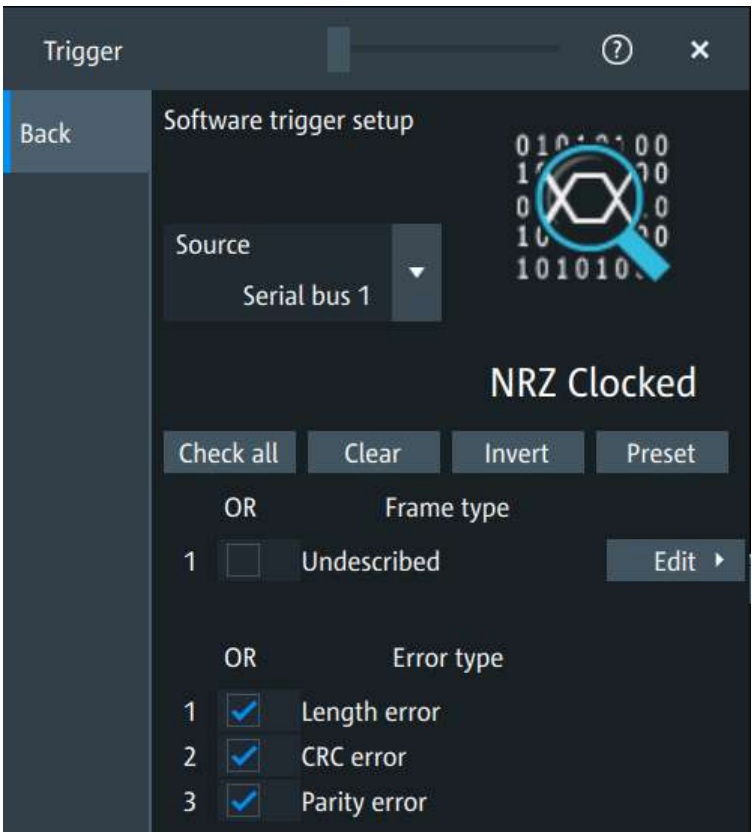
### 14.8.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.8.5.2 Software trigger settings

Access:

- "Menu" > "Apps" > "Protocol" tab > "NRZ Clocked" > "Trigger" tab > "Setup Software Trigger"
- "Menu" > "Apps" > "Protocol" tab > "NRZ Unclocked" > "Trigger" tab > "Setup Software Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:NRZC:CHKall` on page 1585
`TRIGger:SBSW:NRZU:CHKall` on page 1615
`TRIGger:SBSW:NRZC:CLR` on page 1586
`TRIGger:SBSW:NRZU:CLR` on page 1615
`TRIGger:SBSW:NRZC:INVert` on page 1586
`TRIGger:SBSW:NRZU:INVert` on page 1615
`TRIGger:SBSW:NRZC:RST` on page 1586
`TRIGger:SBSW:NRZU:RST` on page 1615

**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

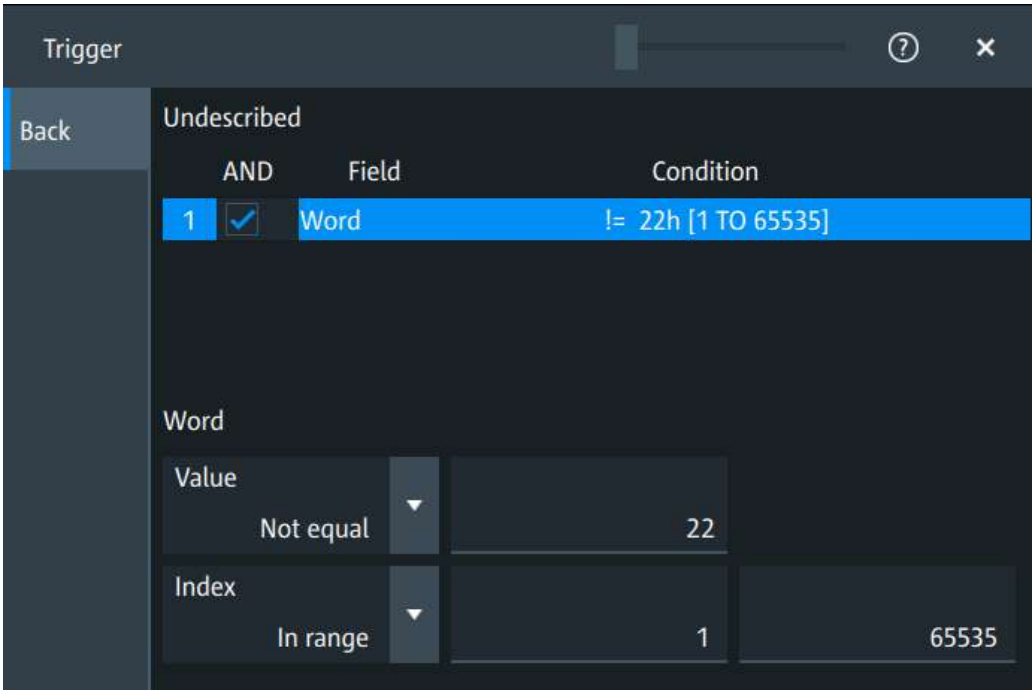
For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The only frame type available by default is "Undescribed", because the NRZ protocols support custom coding.

However, the frame formats that you define appear in this list, too. See Section 14.8.4, "NRZ format configuration", on page 639.

Remote command:
`TRIGger:SBSW:NRZC:FRENable` on page 1586
`TRIGger:SBSW:NRZU:FRENable` on page 1615
`TRIGger:SBSW:NRZC:FRAMe<fr>:ENABle` on page 1586
`TRIGger:SBSW:NRZU:FRAMe<fr>:ENABle` on page 1615

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.



"Field" Enables the field type that you want to trigger on for the selected frame.
For the default "Undescribed" frame, the only available field type is "Word". See "Frame type" on page 648.
However, the field formats that you define appear in this list, too. See Section 14.8.4, "NRZ format configuration", on page 639.

Remote command:
`TRIGger:SBSW:NRZC:FIENable` on page 1588
`TRIGger:SBSW:NRZU:FIENable` on page 1617
`TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:ENABle` on page 1588
`TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:ENABle` on page 1617

"Condition" Displays the value condition for the selected field.

Remote command:
`TRIGger:SBSW:NRZC:BIT` on page 1586
`TRIGger:SBSW:NRZU:BIT` on page 1616
`TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:BIT` on page 1586
`TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:BIT` on page 1616

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`TRIGger:SBSW:NRZC:DMAX` on page 1587
`TRIGger:SBSW:NRZU:DMAX` on page 1616
`TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DMAX` on page 1587
`TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DMAX` on page 1616
`TRIGger:SBSW:NRZC:DMIN` on page 1587
`TRIGger:SBSW:NRZU:DMIN` on page 1616
`TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DMIN` on page 1587
`TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DMIN` on page 1616
`TRIGger:SBSW:NRZC:DOPerator` on page 1588
`TRIGger:SBSW:NRZU:DOPerator` on page 1617
`TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DOPerator` on page 1588
`TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DOPerator` on page 1617

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:NRZC:IMAX on page 1588
TRIGger:SBSW:NRZU:IMAX on page 1618
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IMAX on page 1588
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IMAX on page 1618
TRIGger:SBSW:NRZC:IMIN on page 1589
TRIGger:SBSW:NRZU:IMIN on page 1618
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IMIN on page 1589
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IMIN on page 1618
TRIGger:SBSW:NRZC:IOPerator on page 1589
TRIGger:SBSW:NRZU:IOPerator on page 1618
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IOPerator on page 1589
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IOPerator on page 1618

**Error type**
Enables triggering on the selected error type.

The available error types are "Length error", "CRC error", "Parity error".

Remote command:
TRIGger:SBSW:NRZC:ERENable on page 1589
TRIGger:SBSW:NRZU:ERENable on page 1619
TRIGger:SBSW:NRZC:ERRor<m>:ENABle on page 1589
TRIGger:SBSW:NRZU:ERRor<m>:ENABle on page 1619

### 14.8.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.8.7 NRZ decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

**Decode results table**

The "Decode results" table contains information about all decoded frames.

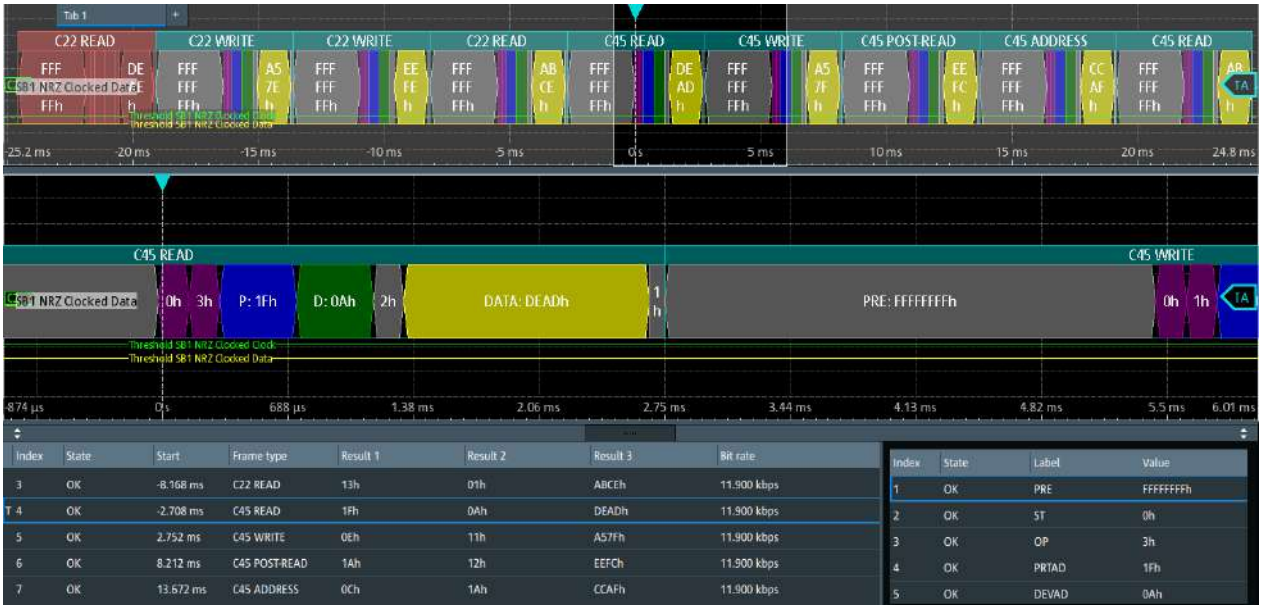



*Figure 14-24: Decoded NRZ clocked signal*

The decode results table contains information about all decoded frames.

*Table 14-12: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Frame count |
| State | Overall state of the frame |
| Start | Start time of the frame |
| Frame type | The type of the frame |
| Result | Data displayed in the 1st, 2nd and 3rd results column. For "undescribed" frames (see Frame type), you can select the data format in the "Display" tab. For other frames, the "Format" table controls the format of each field. See Section 14.8.4, "NRZ format configuration", on page 639. |
| Bit rate | Value of the bit rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

*Table 14-13: Content of the frame details table*

| Column | Description |
|---|---|
| Index | Number of the decoded field |
| Label | Name of the field |
| State | State of the field |
| Value | Value of the field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.8.5, "Decode results", on page 1590 (NRZ clocked) and Section 18.18.9.5, "Decode results", on page 1620 (NRZ unclocked).

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example NRZC export file

```
Index,Start,Stop,State,Type,Result 1,Result 2,Result 3,Bit-rate
1,-0.019539,-0.014079,'OK','C22 WRITE',0Eh,11h,A57Eh,11900
2,-0.014079,-0.008619,'OK','C22 WRITE',1Ah,12h,EEFEh,11900
3,-0.008619,-0.00315997,'OK','C22 READ',13h,01h,ABCEh,11910
4,-0.00315997,0.0023,'OK','C45 READ',1Fh,0Ah,DEADh,11900
5,0.0023,0.0077608,'OK','C45 WRITE',0Eh,11h,A57Fh,11900
6,0.0077608,0.013221,'OK','C45 POST-READ',1Ah,12h,EEFCh,11900
7,0.013221,0.018681,'OK','C45 ADDRESS',0Ch,1Ah,CCAFh,11900
8,0.018681,0.0241433,'OK','C45 READ',13h,01h,ABCDh,11900

Details frame 1
Index,Label,State,Value
1,'PRE','OK',FFFFFFFFh
2,'ST','OK',1h
3,'OP','OK',1h
4,'PHYAD','OK',0Eh
5,'REGAD','OK',11h
6,'TA','OK',2h
7,'DATA','OK',A57Eh
8,'IDLE','OK',1h

Details frame 2
Index,Label,State,Value
1,'PRE','OK',FFFFFFFFh
2,'ST','OK',1h
3,'OP','OK',1h
4,'PHYAD','OK',1Ah
5,'REGAD','OK',12h
6,'TA','OK',2h
7,'DATA','OK',EEFEh
8,'IDLE','OK',1h
```

(...)

## 14.9 Manchester (option R&S MXO3-K510)

Manchester is a simple protocol for enhanced data integrity and reliability in serial bus communication. It is based on a self-clocked coding scheme also known as binary phase-shift keying (BPSK, or phase encoding, PE). Manchester is used in protocols such as ProfiBus (IEC 61158), DALI (Digital Addressable Lighting Interface, IEC 60929 and IEC 62386), MVB (Multifunction Vehicle Bus, part of IEC 61375 for Train Communication Networks, TCN), and Ethernet 10BASE-T (10 Mbit/s, IEEE 802.3i).

- Manchester basics ..... 653
- Manchester configuration ..... 655
- Manchester filter ..... 660
- Manchester format configuration ..... 663
- Manchester software trigger ..... 668
- Measure ..... 672
- Manchester decode results ..... 672

### 14.9.1 Manchester basics

There are two opposing data polarity conventions:

- In Manchester coding (as per G. E. Thomas), a logical 0 is expressed by a low-to-high transition, while a logical 1 is represented by a high-to-low transition. These line voltage transitions occur at the middle of each bit period.
- In "Manchester II" coding (as per IEEE 802.3), the value of each bit is the Boolean exclusive disjunction (XOR) of the original data value and the clock value. Hence, it is equivalent to inverted Manchester coding: a logical 0 is represented by a high-to-low transition, a logical 1 is represented by a low-to-high transition.

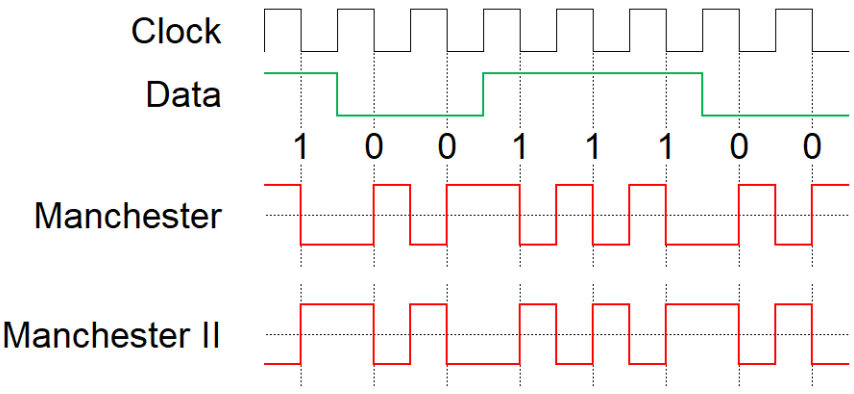



*Figure 14-25: Data polarity conventions for Manchester and Manchester II*

Transitions at the start of a bit period are overhead and do not represent data. Because of this overhead, Manchester coding requires twice the bandwidth of the related NRZ code.

However, the frequent transitions at the beginning and middle of bit periods support clock recovery. All transitions are directly proportional to the clock rate, allowing the signal to be self-clocked. Unless idle, the voltage level is either high or low: it does not return to a zero voltage level between bits.

#### 14.9.1.1 Special features of Manchester coding

In practical protocols, Manchester coding appears in many variations, often employing deliberate coding violations to encode special waveform features, such as unambiguous synchronization and termination patterns. To adapt to these specific Manchester implementations and handle ambiguous signals, the option R&S MXO3-K510 uses a combination of automatic algorithms and user configurable parameters.

**Quaternary Symbols**

The software supports not just traditional binary symbols "0" and "1", but also arbitrary violation waveforms that use two additional symbols, yielding a total of four valid "quaternary bit" values. The two additional violation symbols are "H" (high) and "L" (low). Values of "H" correspond to a waveform lacking a transition in the center of the bit, with a physical high voltage state. Similarly, "L" violations also lack a center transition, but have a physical low voltage state. Most Manchester synchronization and termination conventions, even those containing violations, may be expressed as sequences of these four symbols. R&S MXO3-K510 uses the quaternary notation to support Manchester patterns in the honeycomb display and to describe synchronization and termination patterns in the frame description table.

**Idle Conditions**

The state of the signal line in between messages is the idle condition. Manchester appears in practical standards with varying idle conditions: it can idle at the high, low, or middle voltage state. High and low idle states correspond to "biphase" Manchester, while the middle voltage (often ground) adds a third state to become "ternary" Manchester. Using ternary Manchester, option R&S MXO3-K510 can usually establish the gaps between messages automatically. Using binary Manchester, the software has no way to automatically discriminate an idling bus from monotonic sequences of "H" or "L" violations. For these biphase situations, R&S MXO3-K510 offers a "Gap Time" detection feature, which allows to distinguish long intervals of non-transitions between bus idling and sequences of violations. Other differences between biphase and ternary Manchester are managed automatically by the software, with no user input required.

**Edge Conventions**

Most Manchester encodings establish the beginning of the first bit by a first transition, hence an "overhead" edge. The center of the bit is then marked by a second transition, which is a "sampling" edge. Some Manchester implementations, however, sample the first bit on the first edge. The option R&S MXO3-K510 attempts to automatically detect this situation. Unfortunately, it is possible to trick the algorithm with waveforms that

contain many (legitimate) violations. In these situations, the user can force a "First Edge" or "Second Edge" convention for handling edges. Edge sampling according to the "First Edge" convention is more likely to appear in biphase Manchester, but the software also supports this setting for ternary Manchester situations.

**Bit Rate**

Typically, a single bit rate is clearly specified in Manchester protocols; however, some implementations use a variable bit rate. If you do not specify it in the "Setup" tab, R&S MXO3-K510 by default automatically estimates it. However, with an estimated (hence, undefined) bit rate, fundamental ambiguities are possible in Manchester. In particular, sequences like "0000", "1111", "0101", "1010", and many situations involving "H" and "L" violations, cannot be decoded without it. The situation becomes even less defined with eventual Manchester coding violations. In these situations, you must provide a fixed ""Bit rate"" setting to bypass the software's estimation algorithm.

## 14.9.2 Manchester configuration

### 14.9.2.1 Manchester configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "Manchester" > "Setup".

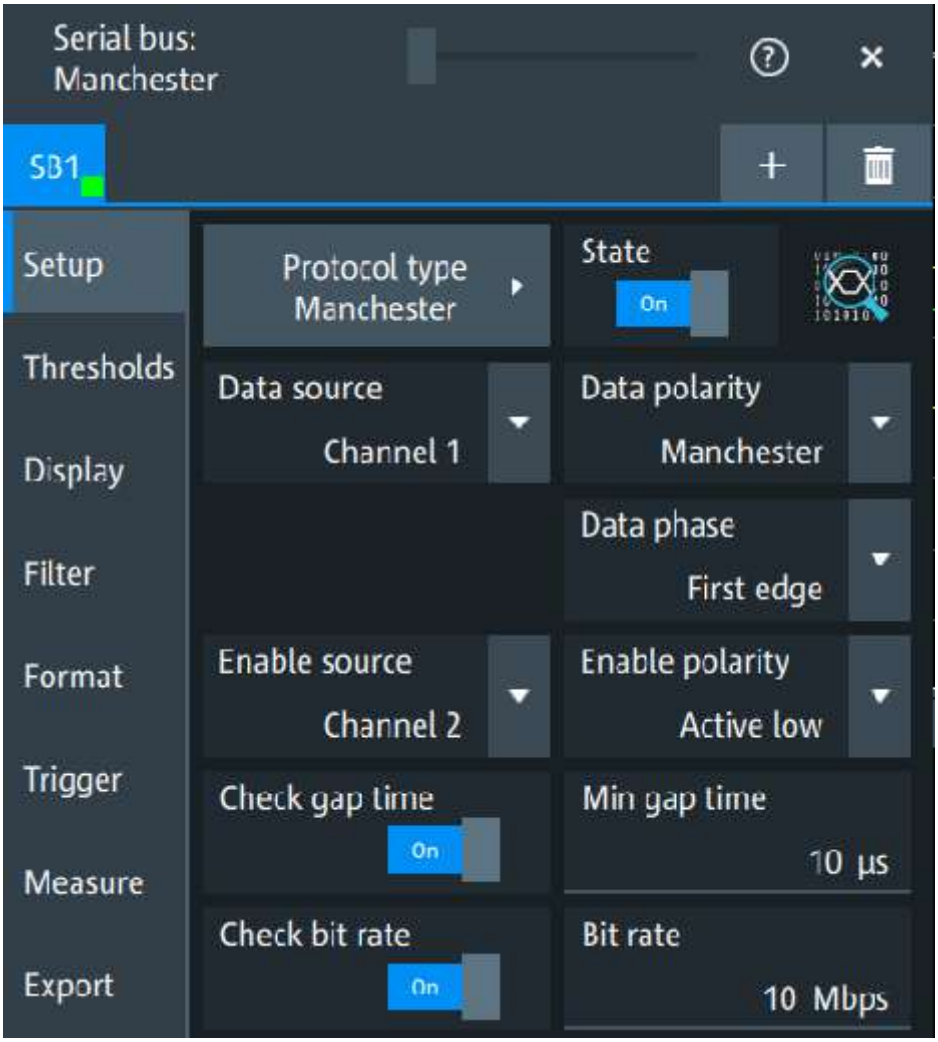



*Figure 14-26: Setup dialog, Data polarity selects "Manchester" or "Manchester II"*

Make sure that the tab of the correct serial bus is selected.

**Protocol type**
Selects the protocol type to be decoded.

Remote command:
`SBUS<sb>:TYPE` on page 1362

**State**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
`SBUS<sb>[:STATe]` on page 1362

**Data source**
Selects the source for the data signal.

Remote command:
SBUS<sb>:MANCh:DATA:SOURce on page 1625

**Data polarity**
Selects the polarity for the data signal, and hence the data representation convention, see Figure 14-25.

- "Manchester" (as per G. E. Thomas):
  – Low-to-high transition for logical 0
  – High-to-low transition for logical 1
- "Manchester II" (as per IEEE 802.3):
  – High-to-low transition for logical 0
  – Low-to-high transition for logical 1

Remote command:
SBUS<sb>:MANCh:DATA:POLarity on page 1626

**Data phase**
Defines the phase for the Manchester data signal. The available settings are:

- "First edge": the decoder captures data bits on the clock's rising edge (low-to-high transition).
  Select this option, if the 1st transition is no overhead and needs to be sampled.
- "Second edge": the decoder captures data bits on the clock's falling edge (high-to-low transition).
  Select this option, if the 1st edge is only an overhead transition from the idle state.

**Note:** The requirement to specify "First edge" or "Second edge" has the following background: An edge is always a transition from high to low or from low to high. In Manchester coding, however, if the signal comes from the idle state, there is right before the first valid edge always an overhead transition from idle to high or from idle to low. Some standards can regard this as a valid transition. Defining the phase avoids this potential ambiguity.
See also Special features of Manchester coding.

Remote command:
SBUS<sb>:MANCh:DATA:PHASe on page 1626

**Enable source**
Selects the source of the enable signal.

If you select any source other than "None", the "Enable polarity" becomes available, which selects the polarity of the enable signal.

Remote command:
SBUS<sb>:MANCh:ENABle:SOURce on page 1627
SBUS<sb>:MANCh:ENABle:POLarity on page 1627

**Check gap time**
Enables the detection of the minimum idle time between two frames during decoding. Also enables setting the "Min gap time". Optionally, if you set it, the decoder considers any longer idle time a gap. A known minimum gap time helps the decoder with frame synchronization.

Remote command:
`SBUS<sb>:MANCh:MINGap:SELect` on page 1627
`SBUS<sb>:MANCh:MINGap:WIDTh` on page 1628

**Check bit rate**
Enables setting the "Bit rate".
- If you define a bit rate, it is used for triggering and decoding.
- If you disable this option, the decoder calculates the bit rate.

Remote command:
`SBUS<sb>:MANCh:BITRate:SELect` on page 1628
`SBUS<sb>:MANCh:BITRate:WIDTh` on page 1628

### 14.9.2.2 Thresholds

Access: "Menu" > "Apps" > "Protocol" tab > "Manchester" > "Thresholds"

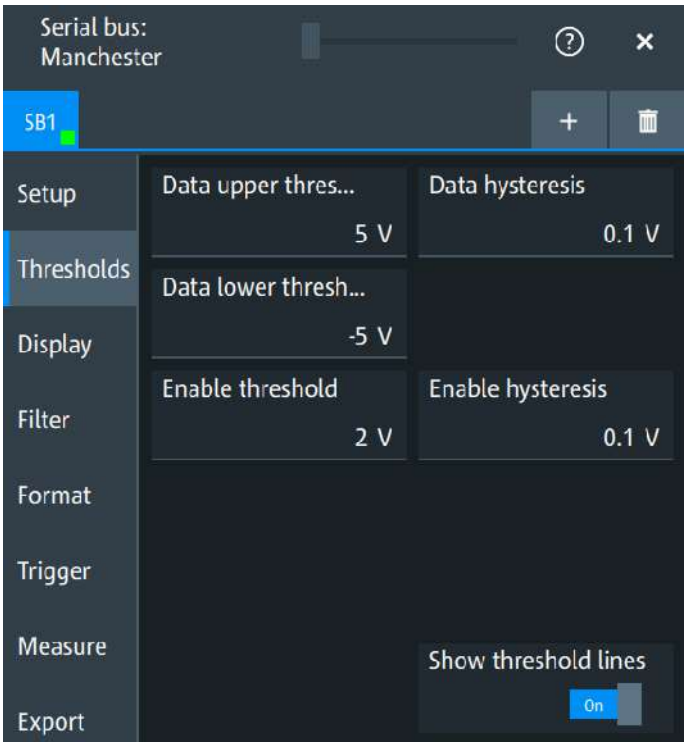



*Figure 14-27: Thresholds dialog*

Enter the values directly in the fields.

**Upper threshold, Lower threshold, Hysteresis**
Sets the upper and lower threshold and the hysteresis for the data source channel.

Remote command:
SBUS<sb>:MANCh:DATA:THUPper on page 1628
SBUS<sb>:MANCh:DATA:THLower on page 1629
SBUS<sb>:MANCh:DATA:HYSTeresis on page 1629

**Enable threshold, Enable hysteresis**
Available only, if you have selected a source for the enable signal. See "Enable source" on page 657.

Sets the middle threshold and the hysteresis for the enable source channel.

Remote command:
SBUS<sb>:MANCh:ENABle:THReshold on page 1629
SBUS<sb>:MANCh:ENABle:HYSTeresis on page 1629

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
SBUS<sb>:THReshold on page 1364

#### 14.9.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

| | |
|---|---|
| "Off" | Disables the display of the decode layer. |
| "Edges" | Enables the display of all edges. |
| "Bits" | Enables the display of all encoded bits. |

**Data format**
See "Data format" on page 493.

#### 14.9.2.4 Manchester symbols

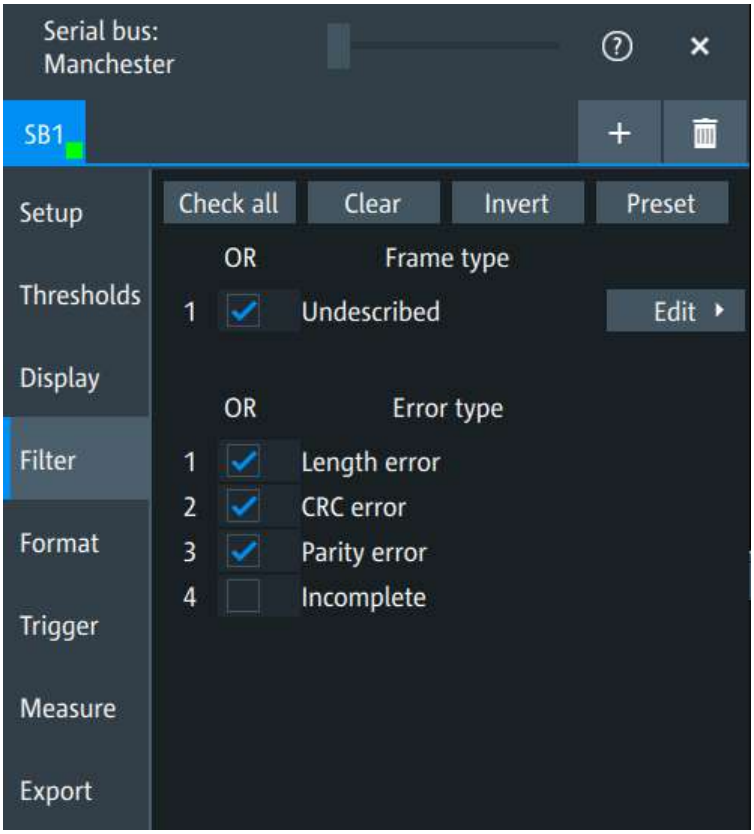
There is no symbolic translation for the Manchester protocol, because of the custom frame definitions. See Section 14.9.4, "Manchester format configuration", on page 663.

### 14.9.3 Manchester filter

Access: "Menu" > "Apps" > "Protocol" tab > "Manchester" > "Filter" tab

Although Manchester has a free format description according to section Section 14.9.4, "Manchester format configuration", on page 663, we provide a dynamic filter solution that can adapt to custom frames and fields in the decoded events.

In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.




**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
`SBUS<sb>:MANCh:FILTer:CHKall` on page 1631
`SBUS<sb>:MANCh:FILTer:CLR` on page 1631
`SBUS<sb>:MANCh:FILTer:INVert` on page 1631
`SBUS<sb>:MANCh:FILTer:RST` on page 1632

**Enable**
Enables the filtering on Manchester frames. Only the frames that match the selected filter conditions are displayed.

Remote command:
`SBUS<sb>:MANCh:FILTer:FRENable` on page 1634
`SBUS<sb>:MANCh:FILTer:FRAMe<fr>:ENABle` on page 1634

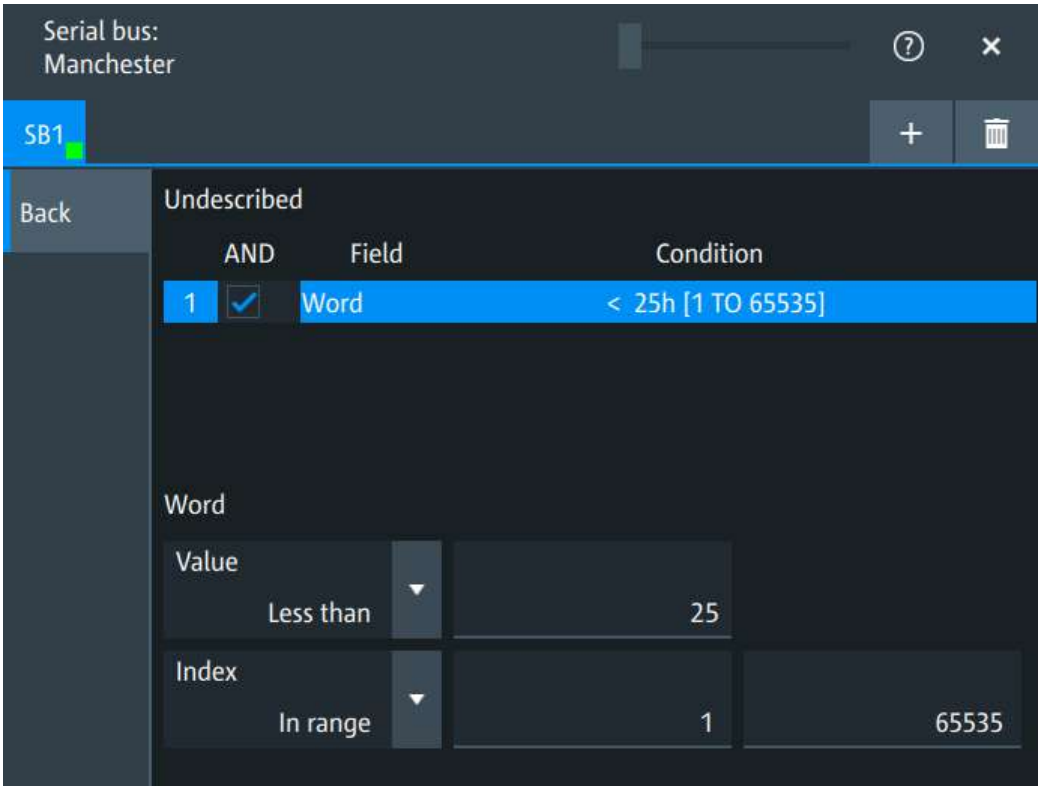
**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

The only frame type available by default is "Undescribed", because the Manchester protocol supports custom coding.

However, the frame formats that you define appear in this list, too. See Section 14.9.4, "Manchester format configuration", on page 663.

**Edit**
Opens a dialog to define the details of the selected frame.



"Field" Enables the field type that you want to filter on for the selected frame. For the default "Undescribed" frame, the only available field type is "Word", which is the payload component of an undescribed frame. See "Frame type" on page 661.
However, the field formats that you define appear in this list, too. See Section 14.9.4, "Manchester format configuration", on page 663.

Remote command:
`SBUS<sb>:MANCh:FILTer:FIENable` on page 1634
`SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:ENABle` on page 1634

"Condition" Displays the value condition for the selected field.

Remote command:
`SBUS<sb>:MANCh:FILTer:BIT` on page 1632
`SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:BIT` on page 1632

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`SBUS<sb>:MANCh:FILTer:DMAX` on page 1633
`SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DMAX` on page 1633
`SBUS<sb>:MANCh:FILTer:DMIN` on page 1633
`SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DMIN` on page 1633
`SBUS<sb>:MANCh:FILTer:DOPerator` on page 1633
`SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DOPerator` on page 1633

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
`SBUS<sb>:MANCh:FILTer:IMAX` on page 1634
`SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IMAX` on page 1634
`SBUS<sb>:MANCh:FILTer:IMIN` on page 1635
`SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IMIN` on page 1635
`SBUS<sb>:MANCh:FILTer:IOPerator` on page 1635
`SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IOPerator` on page 1635

**Error type**
Enables filtering on the selected error type.

The available error types are "Length error", "CRC error", "Parity error", "Incomplete".

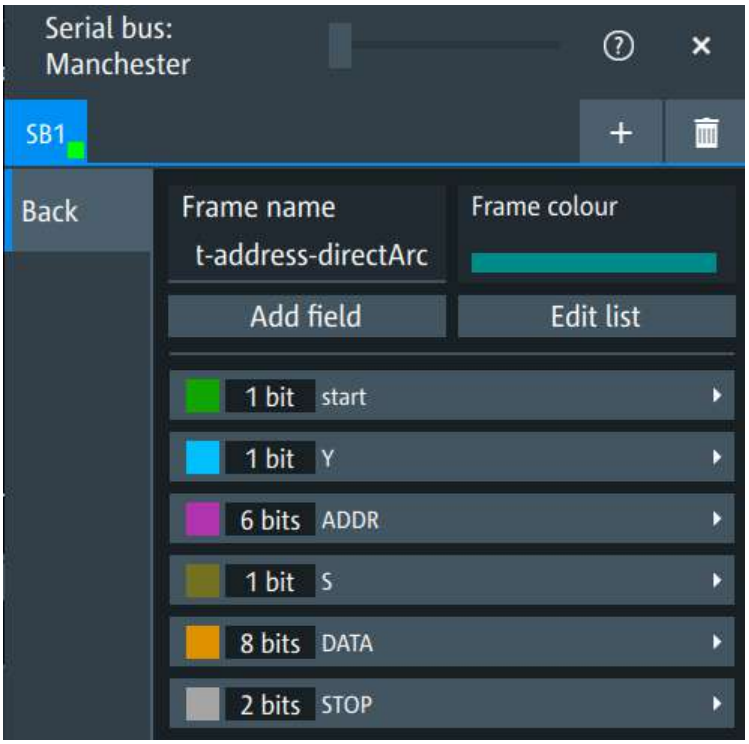
Remote command:
`SBUS<sb>:MANCh:FILTer:ERENable` on page 1632
`SBUS<sb>:MANCh:FILTer:ERRor<n>:ENABle` on page 1632

### 14.9.4 Manchester format configuration

Access: "Menu" > "Apps" > "Protocol" tab > "Manchester" > "Format".

See Section 14.1.5, "Format", on page 497.

**Add Frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:MANCh:FORMat:ADDFrame on page 1636

**Edit List**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:MANCh:FORMat:CLR on page 1637
SBUS<sb>:MANCh:FORMat:FCOunt? on page 1637

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:MANCh:FORMat:LOAD on page 1636

**Save as**
Opens a dialog for saving your current list of frame formats to an `XML` file. Saved frame formats can support efficient and convenient working.

Remote command:
`SBUS<sb>:MANCh:FORMat:SAVE` on page 1637

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

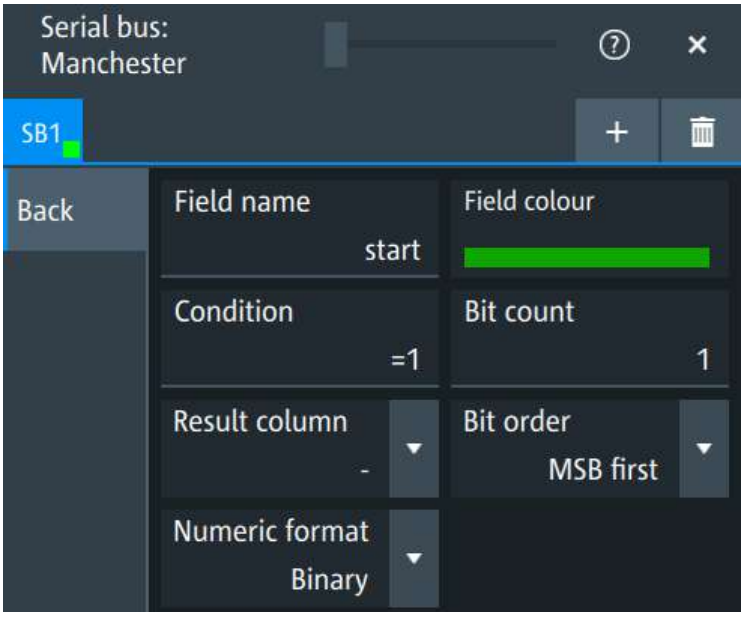
Remote command:
`SBUS<sb>:MANCh:FORMat:FRAMe<fr>:NAME` on page 1638

**Frame Color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
`SBUS<sb>:MANCh:FORMat:FRAMe<fr>:COLor` on page 1638

**Add Field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
`SBUS<sb>:MANCh:FORMat:FRAMe<fr>:ADDField` on page 1638

**Edit List**
Enables editing the field list. If enabled, you can delete a field description by clicking its "Delete" icon. In a list with several entries, you can move field descriptions up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
`SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLDCount?` on page 1638

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
`SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:NAME` on page 1639

**Field Color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
`SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:COLor` on page 1639

**Condition**
Defines a rule or pattern that a field must match to help identify a frame and/or validate mandatory values, e.g., CRC, ID. The condition must match the field's "Numeric format", "Bit order", and "Bit length".

The following conditions are implemented:

"=" (equal / pattern match) — Defines a bit pattern that the field must match. Allowed characters depend on the field's "Format", "Bit order", and "Bits". In binary format, for example, valid characters are "1", "0", "H" (high), and "L" (low).

The behavior depends on whether the frame contains a variable-length array [ ]:

- **No [ ] in the frame**: All fields using = act as keys to identify the frame type. The frame matches only if all such fields match.
- **Frame contains one [ ]**: If there is a "Variable Length Array" field, then the equal operator has two different functionalities, depending on the position of the equal-operator field within the frame format configuration:
  - **Field is before the [ ]**: The condition acts as a key to identify a frame type and matches starting bits.
  - **Field is immediately after the [ ]**: The condition acts as an array delimiter and matches the bits right after the variable-length data to terminate the array.
    Note: If a = field is after the [ ] but not immediately after it, decode results are unpredictable.

"[ ]" (array / repetition)
Defines the number of permissible repetitions of the field.

- **Fixed-length array [n] (n > 0)**: Repeat the field n times. Effective length is n × "Bit count". Elements appear individually in results/ honeycomb under the same name and color.
- **Variable-length array [ ]**: Repeat the field 0 or more times, so the frame length is unspecified and ends via an end-of-frame condition, which can be an operator or gap.
  Constraints:
  – Only one variable-length array is supported per frame (processing limitation).
  – A delimiter field must follow immediately after [ ] (unless [ ] is the last field) so the decoder can terminate the array.
  – Variable-length array elements appear in details as Name: 1, Name: 2, etc. The selected field shows as Name[n] where n is the detected size.
  – Exception: Consecutive variable arrays can form a repeated structure, e.g. A[] + B[] → ABABAB... until the end condition has been detected.
  – Check functions can be combined in dynamic arrays, e.g., B[] can check parity relative to A[] by index.

"crc5usb(n-m)" (CRC check, 5-bit, USB polynomial)
Verifies a 5-bit CRC over fields in the index range n to m. For example, use "crc5usb(1-5)" to check fields 1 through 4. If the range includes an array, all elements in the array are included.
If the CRC check fails, the CRC field is marked "CRC error" (red), and the containing frame marked similarly unless a higher-priority error exists.

"odd(n-m), even(n-m)" (parity check)
Checks odd/even parity over the index range n to m including the parity bit.
If the parity check fails, the parity field is marked "CRC error" (red), and the containing frame marked similarly unless a higher-priority error exists.

Remote command:
`SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:CONDition` on page 1640

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.
- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  – Bit count = 2: truncated. Result: =00
  – Bit count = 4: padded. Result: =0100

- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  - Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  - Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1639

**Result column**
Determines which field is displayed in which result column of the decode table.

- Selecting "-" (no index) means that the result is not displayed.
- Selecting "1", "2" or "3" means that the result is displayed in the 1st, 2nd or 3rd result column, that the decode table supports. Each of the three result columns has to be unique for each frame type. But to display unrelated information for different frame types, you can define different result columns.

Remote command:
`SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:CLMN` on page 1641

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
`SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:BITorder` on page 1641

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:

- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  - "=1HL111000": valid (read as binary)
  - "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  - "=1010": valid (read as HEX, with a total of 16 bits)
  - "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)
  - "=5X12": invalid
  - "=1H33": invalid

Remote command:
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1640

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1642

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1642

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1643

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1643

**Parameter A,B**
Defines the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1643
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1643

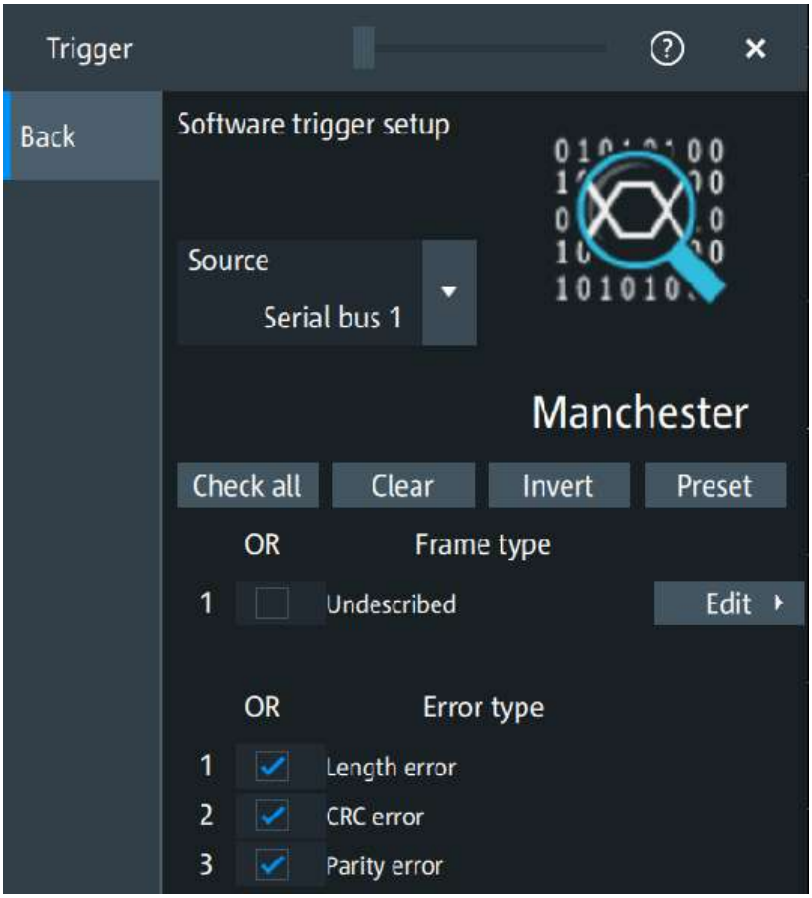
## 14.9.5 Manchester software trigger

### 14.9.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

#### 14.9.5.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "Manchester" > "Trigger" tab > "Setup Software Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:MANCh:CHKall` on page 1645
`TRIGger:SBSW:MANCh:CLR` on page 1645

TRIGger:SBSW:MANCh:INVert on page 1645
TRIGger:SBSW:MANCh:RST on page 1645

**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

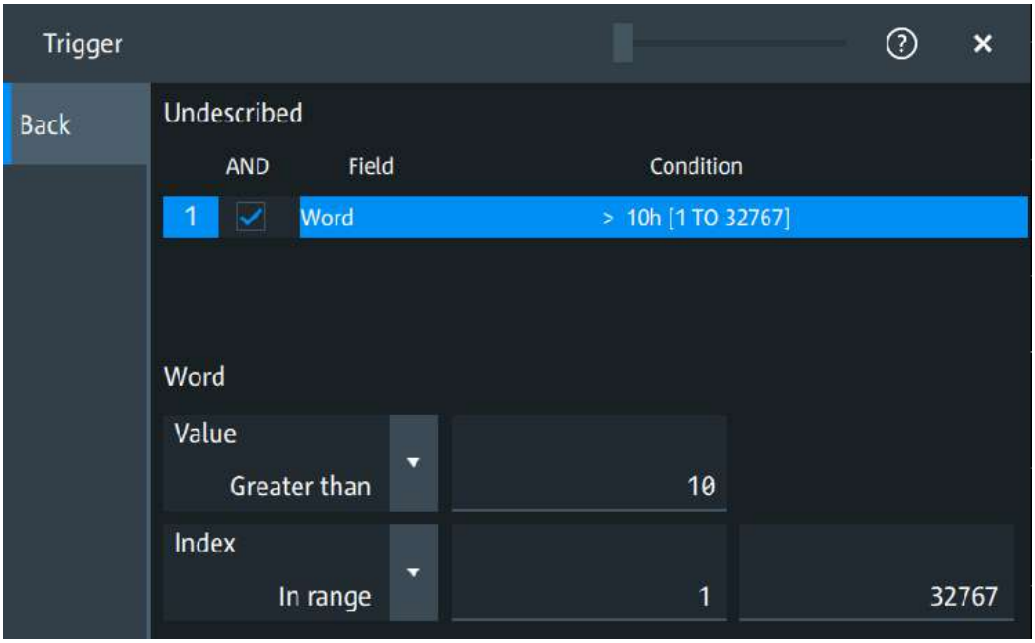
For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The only frame type available by default is "Undescribed", because the Manchester protocol supports custom coding.

However, the frame formats that you define appear in this list, too. See Section 14.9.4, "Manchester format configuration", on page 663.

Remote command:
TRIGger:SBSW:MANCh:FRENable on page 1645
TRIGger:SBSW:MANCh:FRAMe<fr>:ENABle on page 1645

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.



"Field" Enables the field type that you want to trigger on for the selected frame.
For the default "Undescribed" frame, the only available field type is "Word". See "Frame type" on page 670.
However, the field formats that you define appear in this list, too. See Section 14.9.4, "Manchester format configuration", on page 663.

Remote command:
`TRIGger:SBSW:MANCh:FIENable` on page 1647
`TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:ENABle` on page 1647

"Condition" Displays the value condition for the selected field.

Remote command:
`TRIGger:SBSW:MANCh:BIT` on page 1646
`TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:BIT` on page 1646

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`TRIGger:SBSW:MANCh:DMAX` on page 1646
`TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DMAX` on page 1646
`TRIGger:SBSW:MANCh:DMIN` on page 1646
`TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DMIN` on page 1646
`TRIGger:SBSW:MANCh:DOPerator` on page 1647
`TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DOPerator` on page 1647

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
`TRIGger:SBSW:MANCh:IMAX` on page 1648
`TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IMAX` on page 1648
`TRIGger:SBSW:MANCh:IMIN` on page 1648
`TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IMIN` on page 1648
`TRIGger:SBSW:MANCh:IOPerator` on page 1648
`TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IOPerator` on page 1648

**Error type**
Enables triggering on the selected error type.

The available error types are "Length error", "CRC error", "Parity error".

Remote command:
`TRIGger:SBSW:MANCh:ERENable` on page 1649
`TRIGger:SBSW:MANCh:ERRor<m>:ENABle` on page 1649

### 14.9.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.9.7 Manchester decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

#### Decode results table

The "Decode results" table contains information about all decoded frames.




*Figure 14-28: Decoded Manchester signal*

The decode results table contains information about all decoded frames.

*Table 14-14: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Frame count |
| State | Overall state of the frame |
| Start | Start time of the frame |
| Frame type | The type of the frame |
| Result | Data displayed in the 1st, 2nd and 3rd results column. For "undescribed" frames (see Frame type), you can select the data format in the "Display" tab. For other frames, the "Format" table controls the format of each field. See Section 14.9.4, "Manchester format configuration", on page 663. |
| Bit rate | Value of the bit rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

*Table 14-15: Content of the frame details table*

| Column | Description |
|---|---|
| Index | Number of the decoded field |
| Label | Name of the field |
| State | State of the field |
| Value | Value of the field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.10.5, "Decode results", on page 1650.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example Manchester export file

```
Index,State,Type,Result 1,Result 2,Result 3,Bit-rate
1,'OK','Cmd/Stat',3h,049h,1h,14280
2,'OK','Data',4h,1234h,0h,14290
3,'OK','Data',4h,5678h,1h,14280
4,'OK','Data',4h,9ABCh,0h,14280
5,'OK','Data',4h,DEF0h,1h,14280
6,'OK','Cmd/Stat',3h,000h,0h,14270
7,'OK','Cmd/Stat',3h,083h,1h,14280
8,'PAR','Cmd/Stat',3h,083h,0h,14280

Details frame 1
Index,Label,State,Value
1,'Sync','OK',3h
2,'RTA','OK',01h
3,'Info','OK',049h
4,'P','OK',1h

Details frame 2
Index,Label,State,Value
1,'Sync','OK',4h
2,'Data','OK',1234h
3,'P','OK',0h
```

(...)

# 14.10 CAN (automotive electronics, option R&S MXO3-K520)

CAN is the Controller Area Network, a bus system designed by Bosch for use within automotive network architecture, for example, for brake, power train and engine management. Today, it is also used in many other systems, for example, in industrial machines, aerospace, subsea, merchant marine.

With the MXO 3 you can decode frames from some of the most common CAN protocols standards:

- Classical CAN (CAN 2.0): includes the standard frame format (11-bit identifier) and the extended frame format (29-bit identifier).
- CAN FD: CAN with flexible data rate. It is an extension of the classical CAN protocol and allows for increased data rates and payloads. It is backward compatible with CAN 2.0.
- CAN XL: an advanced version of the CAN protocol designed to handle higher data rates and larger payloads compared to classical CAN and CAN FD.

**Requirements**

For performing CAN decode measurements, you need the following:

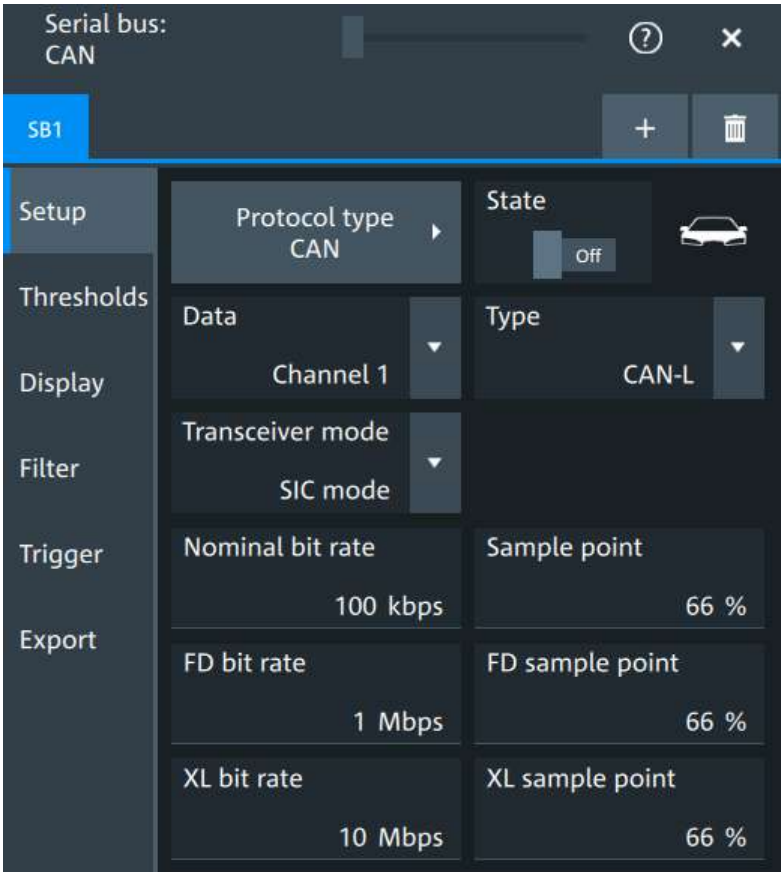
- MXO 3 with 1 available channel. The channel can be:
  - Analog channels (C1-C4)
  - Logic channels (D0-D15) (requires option R&S MXO3-B1)
- One probe
- Option R&S MXO3-K520

- CAN configuration....674
- CAN filter....680
- CAN format configuration....683
- CAN hardware trigger....688
- CAN software trigger....697
- Measure....701
- CAN decode results....701

## 14.10.1 CAN configuration

### 14.10.1.1 CAN configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "CAN" > "Setup".




Make sure that the tab of the correct serial bus is selected.

**Protocol type**
Selects the protocol type to be decoded.

Remote command:
`SBUS<sb>:TYPE` on page 1362

**State**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
`SBUS<sb>[:STATe]` on page 1362

**Data**
Sets the source of the selected data line.

Remote command:
`SBUS<sb>:CAN:SOURce` on page 1656

**Type**
Selects the CAN-High or CAN-Low line. CAN uses both lines for differential signal transmission.

If you measure with a differential probe, connect the probe to both CAN-H and CAN-L lines, and select the data "Type" = *CAN-H*.

If you use a single-ended probe, connect the probe to either CAN_L or CAN_H, and select the data type accordingly.

Remote command:
`SBUS<sb>:CAN:TYPE` on page 1655

**Transceiver mode**
Selects the transceiver mode for the CAN decoding.

"SIC mode" The signal improvement capability (SIC) mode supports bit rates up to 8 Mbit/s.

"FAST mode" The fast mode supports bit rates up to 20 Mbit/s.

Remote command:
`SBUS<sb>:CAN:TRCVmode` on page 1659

**Nominal bit rate**
Sets the number of transmitted bits per second.

Remote command:
`SBUS<sb>:CAN:BITRate` on page 1656

**Sample point**
Sets the position of the sample point within the bit in percent of the nominal bit time.

Remote command:
`SBUS<sb>:CAN:SAMPlepoint` on page 1656

**FD bit rate**
Sets the number of transmitted bits per second for the CAN FD frame.

Remote command:
`SBUS<sb>:CAN:FDATa:DBITrate` on page 1657

**FD sample point**
Sets the position of the sample point within the bit in percent of the nominal bit time for the CAN FD frame.

Remote command:
`SBUS<sb>:CAN:FDATa:SAMPlepoint` on page 1657

**XL bit rate**
Sets the number of transmitted bits per second for the CAN XL frame.

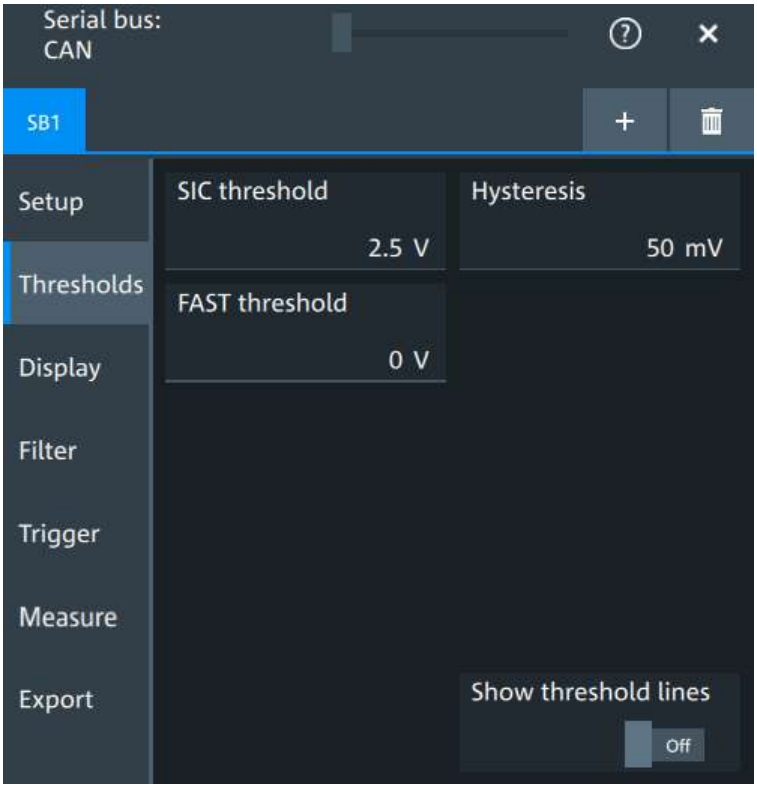
Remote command:
`SBUS<sb>:CAN:XDATa:DBITrate` on page 1658

**XL sample point**
Sets the position of the sample point within the bit in percent of the nominal bit time for the CAN XL frame.

Remote command:
SBUS<sb>:CAN:XDATa:SAMPlepoint on page 1659

### 14.10.1.2 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "CAN" > "Thresholds".




**Threshold**
Sets the threshold for the SIC and Fast channels. Enter the value directly in the fields.

Additional to the threshold, you can also set a hysteresis.

Remote command:
SBUS<sb>:CAN:FAST:HYSTeresis on page 1658
SBUS<sb>:CAN:FAST:THReshold on page 1658
SBUS<sb>:CAN:SIC:HYSTeresis on page 1657
SBUS<sb>:CAN:SIC:THReshold on page 1658

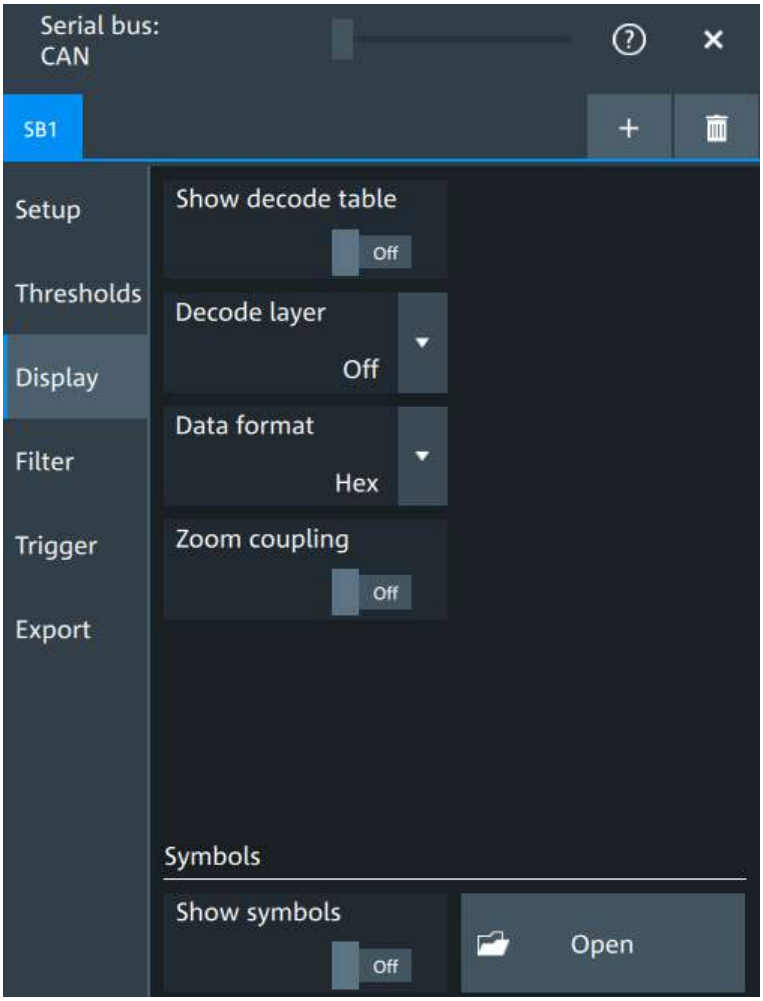
**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
SBUS<sb>:THReshold on page 1364

#### 14.10.1.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.




**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display. See "Decode layer" on page 493.

**Show symbols**
You can load symbol lists, and activate its usage for decoding. As a result, an additional "Symbol" column appears in the "Decode results" table, containing the symbolic label.

Symbol lists for CAN are available in DBC and CSV format, see CAN symbols.

Remote command:
SBUS<sb>:CAN:SYMBols on page 1659
SBUS<sb>:CAN:NEWList on page 1660

#### 14.10.1.4 CAN symbols

For the CAN protocol, you can upload symbol lists containing IDs and a symbolic name for each node. You can load a file in one of the supported formats: DBC and CSV.

**DBC files for CAN and CAN FD**

Industry standard DBC files contain more information than CSV files and translate the abstract decode results to human language. For each frame, the frame ID and the symbolic name of the ID are given. The frames are also called messages in CAN. The data of a CAN message can consist of several "signals". The DBC file provides the label, unit, start bit, length and other indicators for each signal. For state-encoded signals, the meaning of the states is given.

In the following demo example, the message "EngineData" has the decimal ID 2,166,573,756 and consists of 8 data bytes. These 8 bytes are defined as 6 signals. The first one, "PetrolLevel", starts at bit #24, has a length of 8 bit, and the unit is liter. The signal "IdleRunning" is state-encoded. It has only one bit. The binary value 0 means "Running", and the binary value 1 means "Idle".

**Example: CAN DBC file section**

```
BO_ 2166573756 EngineData: 8 Engine
 SG_ PetrolLevel : 24|8@1+ (1,0) [0|255] "l" ...
 SG_ EngPower : 48|16@1+ (0.01,0) [0|350] "kW" ...
 SG_ EngForce : 32|10@1+ (1,0) [0|1000] "N" ...
 SG_ IdleRunning : 23|1@1+ (1,0) [0|1] "" ...
 SG_ EngTemp : 16|7@1+ (2,-50) [-50|150] "degC" ....
 SG_ EngSpeed : 0|13@1+ (1,0) [0|8000] "rpm" ...
....
 VAL_ 2166573756 IdleRunning 0 "Running" 1 "Idle" ;
```

**CSV files for CAN / CAN FD**

Symbol list files are protocol-specific. A CSV label file for CAN protocols contains three values for each identifier:

- Identifier type, 11-bit or 29-bit long
- Identifier value
- Symbolic name of the identifier, specifying its function in the bus network.

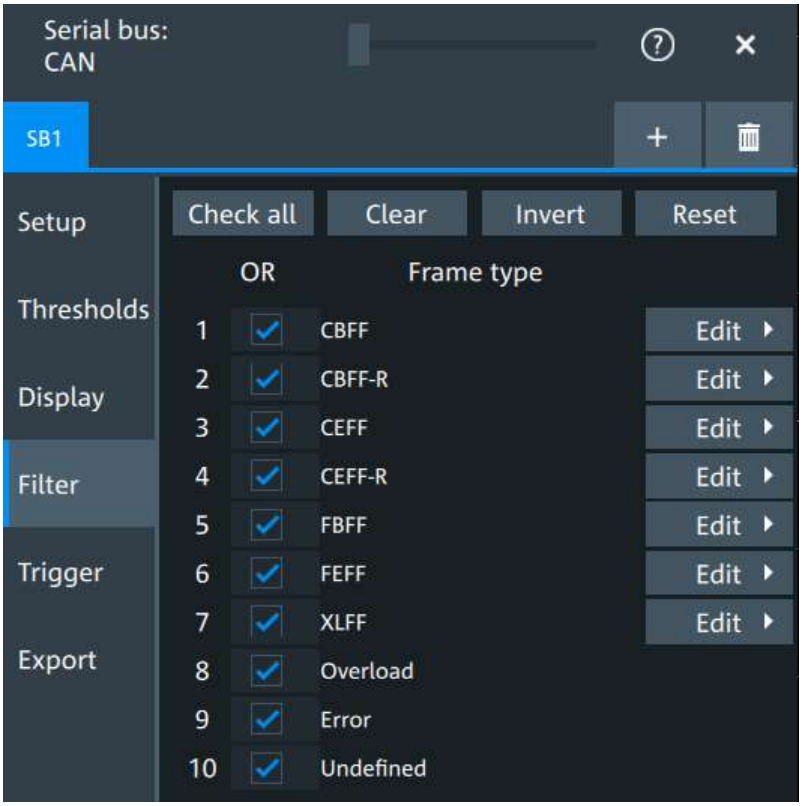
**Example: CAN CSV file**

```
# ------------------------------------------------------------------------
@FILE_VERSION  = 1.00
@PROTOCOL_NAME = can
# ------------------------------------------------------------------------
# Labels for CAN protocol
#   Column order: Identifier type, Identifier value, Label
```

```
# ------------------------------------------------------------------------
11,064h,Diag_Response
11,1E5h,EngineData
11,0A2h,Ignition_Info
11,1BCh,TP_Console
11,333h,ABSdata
11,313h,Door_Left
11,314h,Door_Right
29,01A54321h,Throttle
29,13A00FA2h,LightState
29,0630ABCDh,Engine_Status
29,03B1C002h,Airbag_Status
29,01234ABCh,NM_Gateway
# ------------------------------------------------------------------------
```

### 14.10.2 CAN filter

Access: "Menu" > "Apps" > "Protocol" tab > "CAN" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

| | |
|---|---|
| "Check all" | Enables the filter for all available frames and error types. |
| "Clear" | Disables the filter for all available frames and error types. |
| "Invert" | Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa. |
| "Preset" | Presets the state of the selected frames and error types. |

Remote command:
`SBUS<sb>:CAN:FILTer:CHKall` on page 1661
`SBUS<sb>:CAN:FILTer:CLR` on page 1661
`SBUS<sb>:CAN:FILTer:INVert` on page 1662
`SBUS<sb>:CAN:FILTer:RST` on page 1662

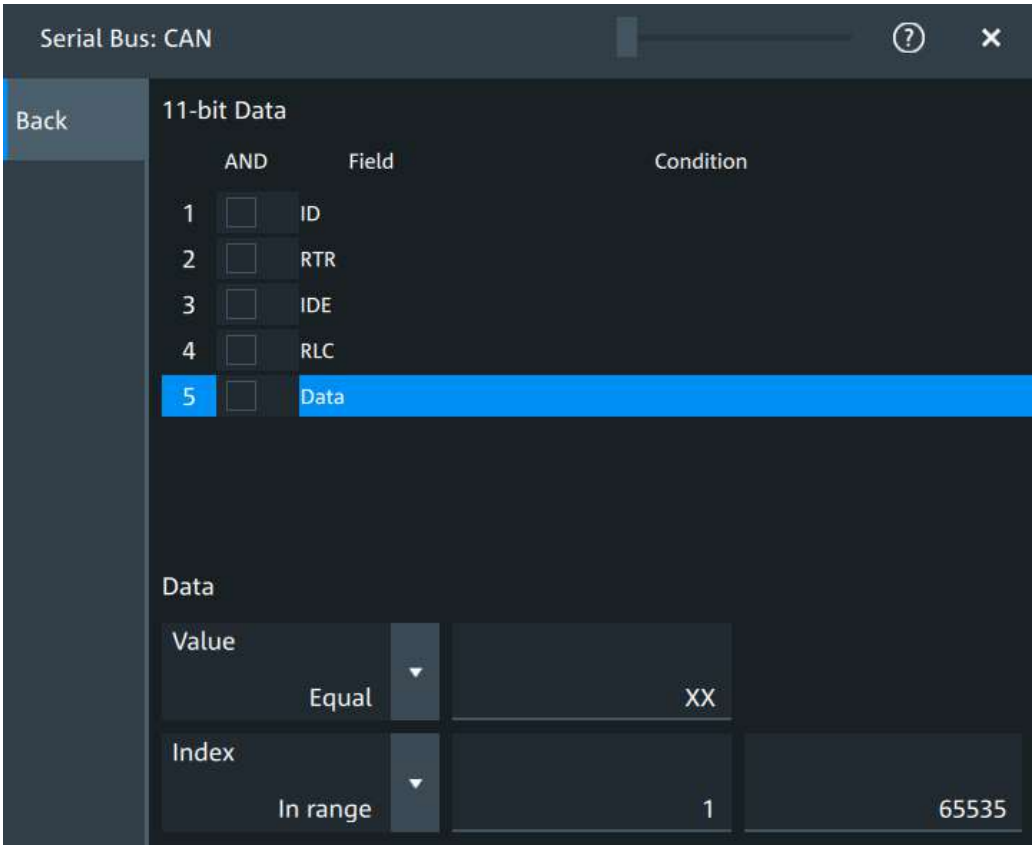
**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

Available frames are "CBFF", "CBFF-R", "CEFF", "CEFF-R", "FBFF", "FEFF", "XLFF", "Overload" and "Error".

Remote command:
`SBUS<sb>:CAN:FILTer:FRENable` on page 1664
`SBUS<sb>:CAN:FILTer:FRAMe<fr>:ENABle` on page 1664

**Edit**
Opens a dialog to define the details of the selected frame.




"Field" Enables the field type that you want to filter on for the selected frame. The available fields are "ID", "RTR", "ITE", "RLC" and "Data".

Remote command:
`SBUS<sb>:CAN:FILTer:FIENable` on page 1664
`SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:ENABle` on page 1664

"Condition" Displays the value condition for the selected field.

Remote command:
`SBUS<sb>:CAN:FILTer:BIT` on page 1662
`SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:BIT` on page 1662

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:CAN:FILTer:DMAX on page 1662
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1662
SBUS<sb>:CAN:FILTer:DMIN on page 1663
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1663
SBUS<sb>:CAN:FILTer:DOPerator on page 1663
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1663

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:CAN:FILTer:IMAX on page 1665
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1665
SBUS<sb>:CAN:FILTer:IMIN on page 1665
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1665
SBUS<sb>:CAN:FILTer:IOPerator on page 1665
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1665

**Error type**
Enables filtering on the selected error type.

Remote command:
SBUS<sb>:CAN:FILTer:ERENable on page 1664
SBUS<sb>:CAN:FILTer:ERRor<n>:ENABle on page 1664

### 14.10.3 CAN format configuration

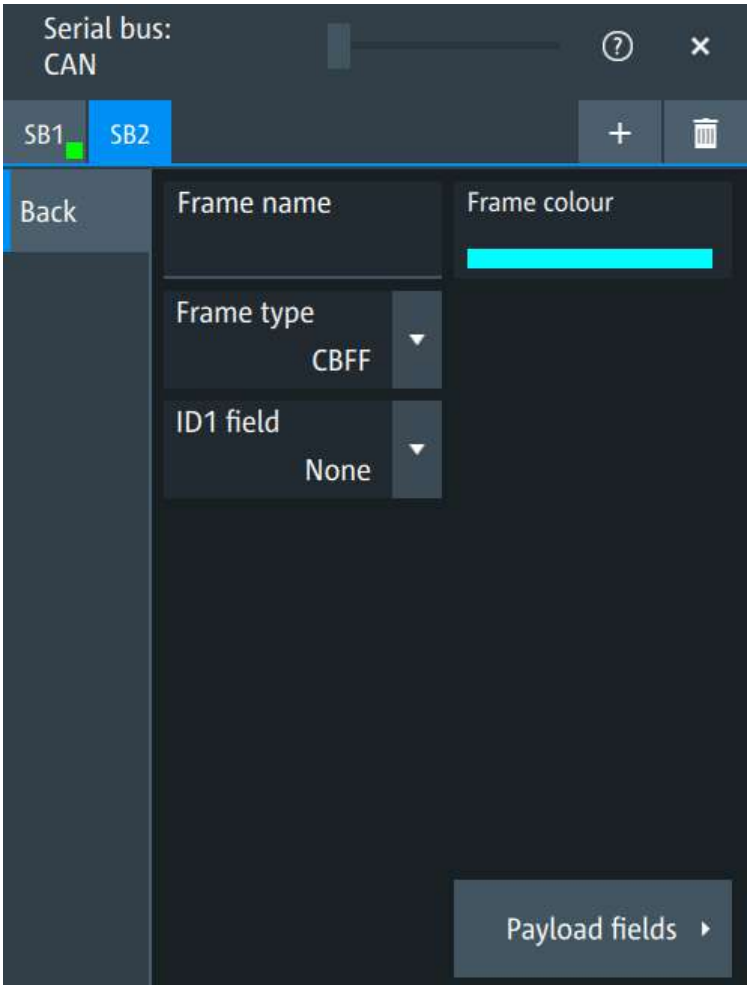
Access: "Menu" > "Apps" > "Protocol" tab > "CAN" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
SBUS<sb>:CAN:FORMat:SELect on page 1666

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:CAN:FORMat:ADDFrame on page 1667

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:CAN:FORMat:CLR on page 1668
SBUS<sb>:CAN:FORMat:FCOunt? on page 1668

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:CAN:FORMat:LOAD on page 1667

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:CAN:FORMat:SAVE on page 1667

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:NAME on page 1669

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:COLor on page 1669

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:TYPE on page 1669

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID1Field on page 1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID2Field on page 1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID3Field on page 1669

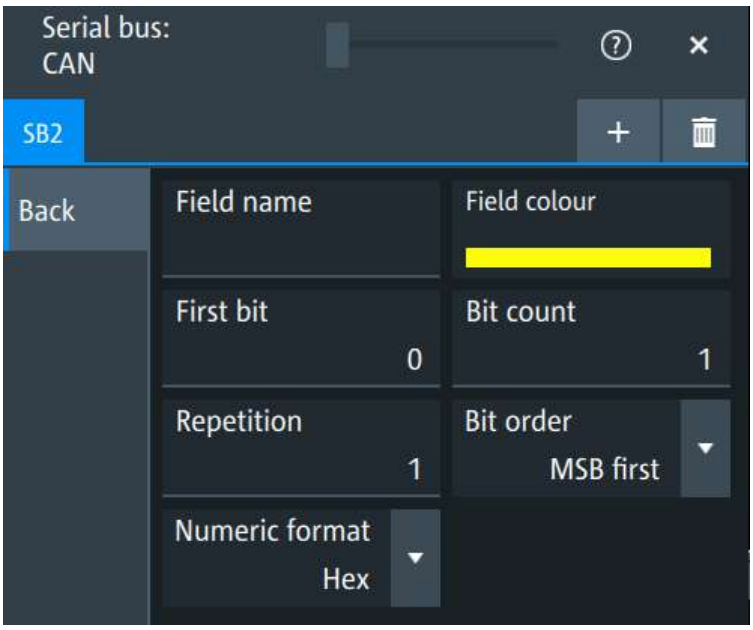
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID1Value on page 1670
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID2Value on page 1670
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID3Value on page 1670

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ADDField on page 1668

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1670

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1670

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1671

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.

- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  – Bit count = 2: truncated. Result: =00
  – Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  – Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  – Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1671

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
`SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:REPetition` on page 1671

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
`SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:BITorder` on page 1672

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:
- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  – "=1HL111000": valid (read as binary)
  – "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  – "=1010": valid (read as HEX, with a total of 16 bits)
  – "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)

– "=5X12": invalid
– "=1H33": invalid

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1672

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1673

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1673

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1674

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1674

**Parameter A,B**
Defines the parameters of the linear function.

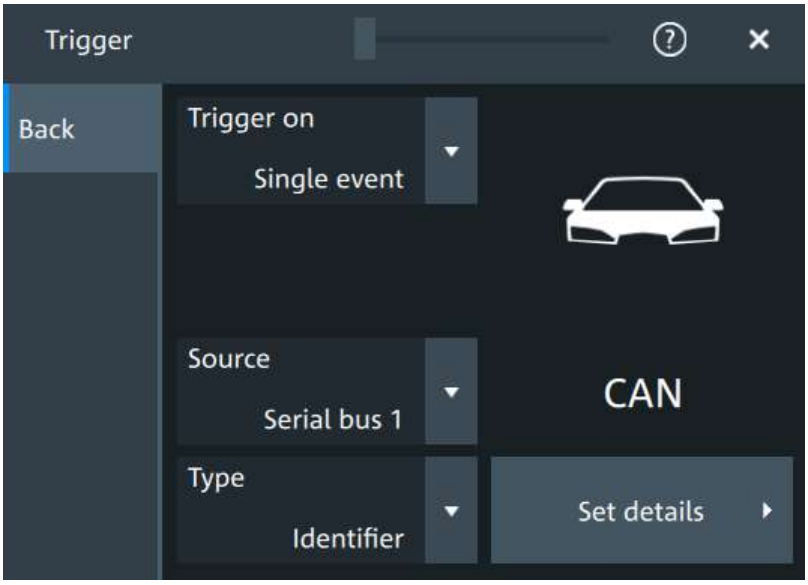
- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1675
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1675

### 14.10.4 CAN hardware trigger

Access: "Menu" > "Apps" > "Protocol" tab > "CAN" > "Trigger" tab > "Setup Hardware Trigger"



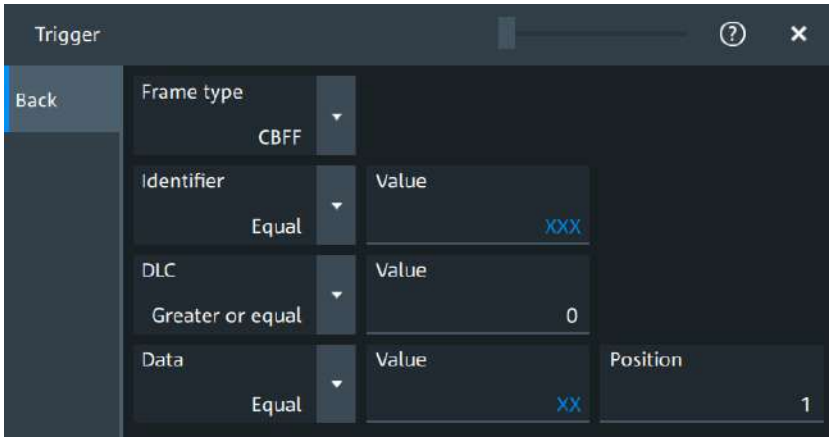

**Type**
Selects the trigger type for CAN analysis.

Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.

"Start of frame" Triggers on the first edge of the dominant SOF bit (synchronization bit).

"End of frame" Triggers on the end of frame.

"Frame type" Triggers on a specified frame type (data, remote, error, or overload). For data and remote frames, also the identifier format is considered.



"Identifier" Sets the trigger to a specific message identifier or an identifier range.



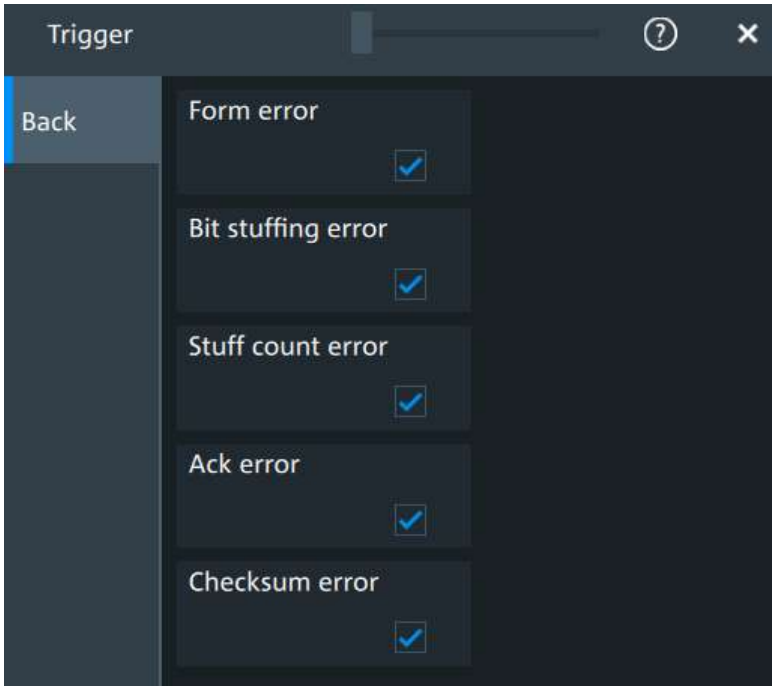

"Identifier + Data"

Sets the trigger to a combination of identifier and data condition. The instrument triggers at the end of the last byte of the specified data pattern.




"Error condition"

Identifies various errors in the frame.



Remote command:
TRIGger:SBHW:CAN:TYPE on page 1681

**Frame type**
CAN has different frame types which can be used as trigger condition. Depending on the selected frame, you can specify further parameters.

"CBFF" Classical base frame format data: supports a length of 11 bit for the CAN identifier. The data frame is the only frame for actual data transmission.




*Figure 14-29: Settings for CBFF and CEFF frames*

"CBFF-R" Classical base frame format remote: supports a length of 11 bit for the CAN identifier.
The remote frame initiates the transmission of data by another node. The frame format is the same as of data frames but without the data field.




*Figure 14-30: Settings for CBFF-R and CEFF-R frames*

"CEFF" Classical extended frame format data: supports a length of 29 bit for the CAN identifier. The data frame is the only frame for actual data transmission.

"CEFF-R" Classical extended frame format remote: supports a length of 29 bit for the CAN identifier.
The remote frame initiates the transmission of data by another node. The frame format is the same as of data frames but without the data field.

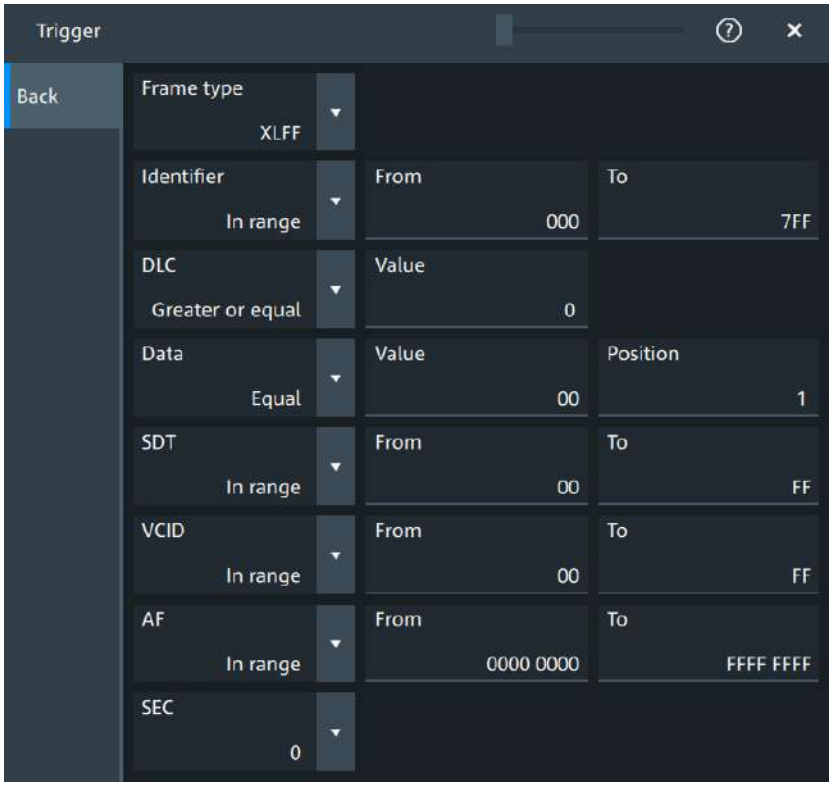
"FBFF" FD base frame format: supports a length of 11 bit for the CAN identifier.




*Figure 14-31: Settings for FBFF and FEFF frames*

"FEFF" FD extended frame format: supports a length of 29 bit for the CAN identifier.

"XLFF" XL frame format: supports a length of 29 bit for the CAN identifier.




"Error" When a node recognizes an error, it cancels transmission by sending an error frame.
The ID type is irrelevant for error frames.

"Overload" When a node needs a delay between data and/or remote frames, it sends and overload frame.
The ID type is irrelevant for overload frames.

Remote command:
`TRIGger:SBHW:CAN:FTYPe` on page 1679

**Identifier**
Specifies the identifier pattern.

"Condition" Sets the comparison condition to a specific value or a range.

"Value"/"From" Specifies the value or sets the start value of a range.

"To" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger:SBHW:CAN:ICONdition` on page 1680
`TRIGger:SBHW:CAN:IMAX` on page 1680
`TRIGger:SBHW:CAN:IMIN` on page 1680

**DLC**
The data length code (DLC) defines the number of data bytes to be found. You can set a fixed value for the DLC or define a minimum value.

Remote command:
TRIGger:SBHW:CAN:DLC on page 1677
TRIGger:SBHW:CAN:DLCCondition on page 1677

**Data pattern**
Specifies the data pattern.

"Condition" Sets the comparison condition to a specific value or a range.

"Value" Specifies the value or sets the start value of a range.

Remote command:
TRIGger:SBHW:CAN:DCONdition on page 1677
TRIGger:SBHW:CAN:DMIN on page 1678

**ID type**
Selects the length of the identifier:

"11 bit" Identifier length of the CAN base frame format. The instrument triggers on the sample point of the identifier extension flag (IDE) bit.

"29 bit" Identifier length of the CAN extended frame format. The instrument triggers on the sample point of the remote transmission request (RTR) bit.

"Any" The ID type and ID pattern are not relevant for the trigger condition.
If the trigger type is "Identifier", the instrument triggers on any identifier in the specified frame type.
If the trigger type is "Identifier + Data", set the "ID type" to "Any" if you want to trigger only on data.

Remote command:
TRIGger:SBHW:CAN:ITYPe on page 1681

**BRS, ESI**
If "Frame type" = "FBFF"/"FEFF", you can trigger on specific bits.

"BRS" The bit rate switch bit. Value 1 means that the bit rate switches from the arbitration rate to the faster data rate.

"ESI" The error state indicator bit. Set "X" if the bit is not relevant.

Remote command:
TRIGger:SBHW:CAN:FDATa:BRS on page 1678
TRIGger:SBHW:CAN:FDATa:ESI on page 1678

**SDT**
Specifies a value or a range for the service data unit type.

"Condition" Sets the comparison condition to a specific value or a range.

"Value"/"From" Specifies the value or sets the start value of a range.

"To" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
TRIGger:SBHW:CAN:XDATa:SDT:CONDition on page 1683
TRIGger:SBHW:CAN:XDATa:SDT:MAX on page 1683
TRIGger:SBHW:CAN:XDATa:SDT:MIN on page 1683

**VCID**
Specifies a value or a range for the virtual CAN network ID (VCID).

| | |
|---|---|
| "Condition" | Sets the comparison condition to a specific value or a range. |
| "Value"/"From" | Specifies the value or sets the start value of a range. |
| "To" | Sets the maximum value of a range for "Condition" = "In range"/"Out of range". |

Remote command:
TRIGger:SBHW:CAN:XDATa:VCID:CONDition on page 1684
TRIGger:SBHW:CAN:XDATa:VCID:MAX on page 1684
TRIGger:SBHW:CAN:XDATa:VCID:MIN on page 1684

**AF**
Specifies a value or a range for the CAN XL acceptance field (AF).

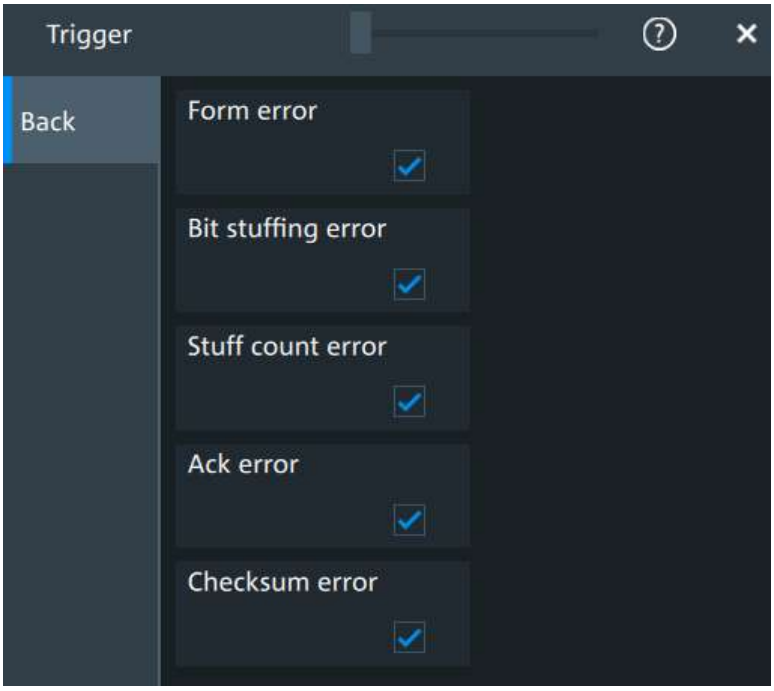
| | |
|---|---|
| "Condition" | Sets the comparison condition to a specific value or a range. |
| "Value"/"From" | Specifies the value or sets the start value of a range. |
| "To" | Sets the maximum value of a range for "Condition" = "In range"/"Out of range". |

Remote command:
TRIGger:SBHW:CAN:XDATa:AF:CONDition on page 1682
TRIGger:SBHW:CAN:XDATa:AF:MAX on page 1682
TRIGger:SBHW:CAN:XDATa:AF:MIN on page 1682

**SEC**
Sets a value for the simple extended content (SEC) field. It indicates, if the CAN XL data frame uses the CADsec protocol.

Remote command:
TRIGger:SBHW:CAN:XDATa:SEC on page 1684

**Error conditions**
If a CAN detects an error, it transmits an error flag at the next bit. The MXO 3 detects errors in the message and triggers on these errors even if no CAN node sends an error flag.




"Form error" A form error occurs when a fixed-form bit field contains one or more illegal bits.

"Bit stuffing error" The frame segments Start of Frame, Arbitration Field, Control Field, Data Field and CRC Sequence are coded by the bit stuffing method. The transmitter automatically inserts a complementary bit into the bitstream when it detects five consecutive bits of identical value in the bitstream to be transmitted. A stuff error occurs when the 6th consecutive equal bit level in the mentioned fields is detected.

"Stuff count error" A stuff bit error occurs, if more than five consecutive bits of the same level occur on the bus.

"Ack error" An acknowledgment error occurs when the transmitter does not receive an acknowledgment - a dominant bit during the ACK Slot.

"Checksum error" CAN uses the Cyclic Redundancy Check, which is a complex checksum calculation method. The transmitter calculates the CRC and sends the result in the CRC sequence. The receiver calculates the CRC in the same way. A CRC error occurs when the calculated result differs from the received value in the CRC sequence.

Remote command:
`TRIGger:SBHW:CAN:FORMerror` on page 1679
`TRIGger:SBHW:CAN:BITSterror` on page 1676
`TRIGger:SBHW:CAN:FDATa:SCERror` on page 1679
`TRIGger:SBHW:CAN:ACKerror` on page 1676
`TRIGger:SBHW:CAN:CRCerror` on page 1677

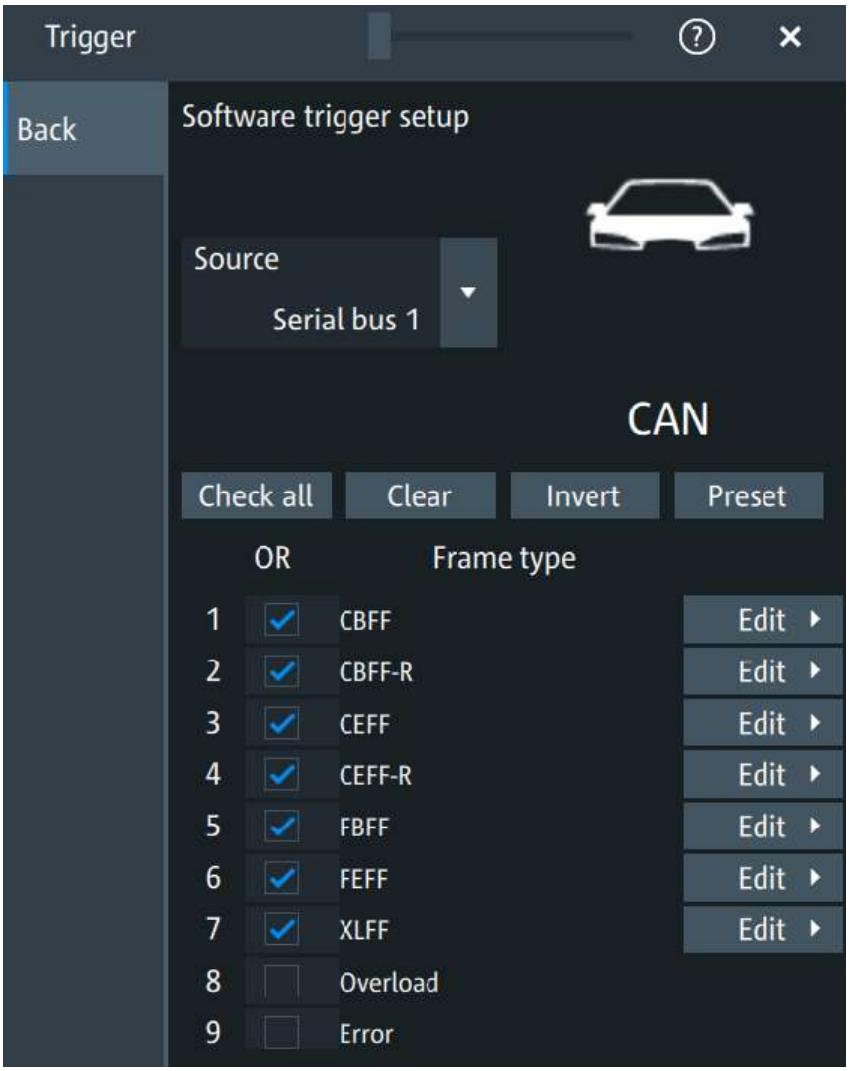
## 14.10.5 CAN software trigger

### 14.10.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.10.5.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "CAN" > "Trigger" tab > "Setup Software Trigger" > "Setup SB Trigger".




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

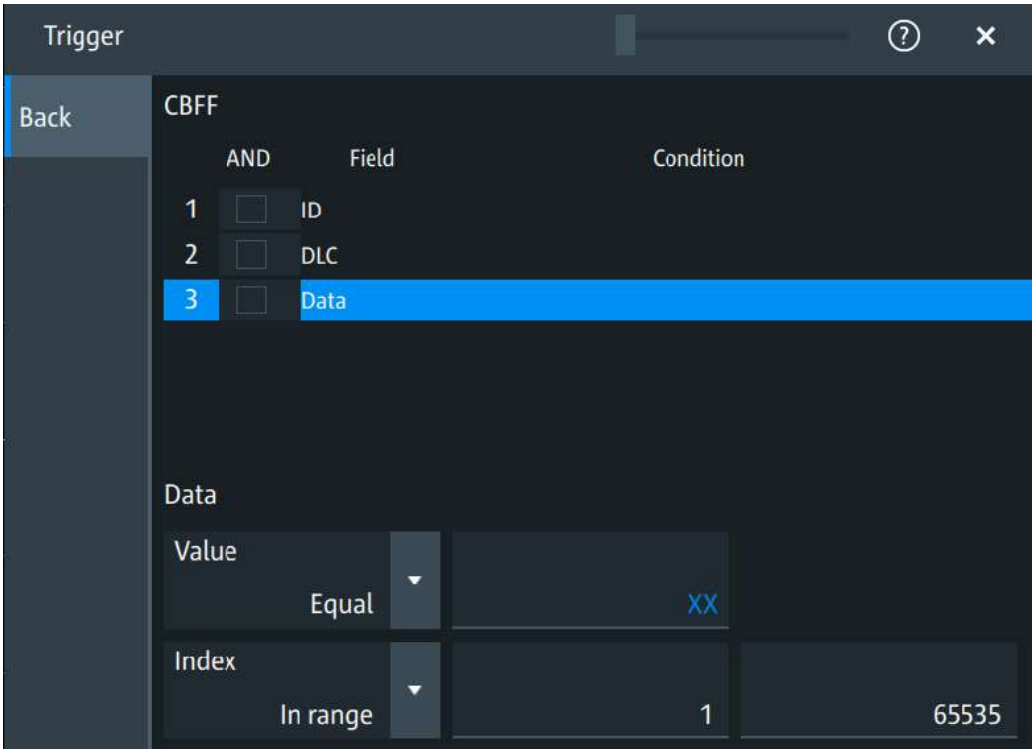
Remote command:
`TRIGger:SBSW:CAN:CHKall` on page 1685
`TRIGger:SBSW:CAN:CLR` on page 1686
`TRIGger:SBSW:CAN:INVert` on page 1686
`TRIGger:SBSW:CAN:RST` on page 1686

**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

Remote command:
`TRIGger:SBSW:CAN:FRENable` on page 1686
`TRIGger:SBSW:CAN:FRAMe<fr>:ENABle` on page 1686

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.




"Field" Enables the field type that you want to trigger on for the selected frame.

Remote command:
`TRIGger:SBSW:CAN:FIENable` on page 1688
`TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:ENABle` on page 1688

"Condition" Displays the value condition for the selected field.

Remote command:
`TRIGger:SBSW:CAN:BIT` on page 1686
`TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:BIT` on page 1686

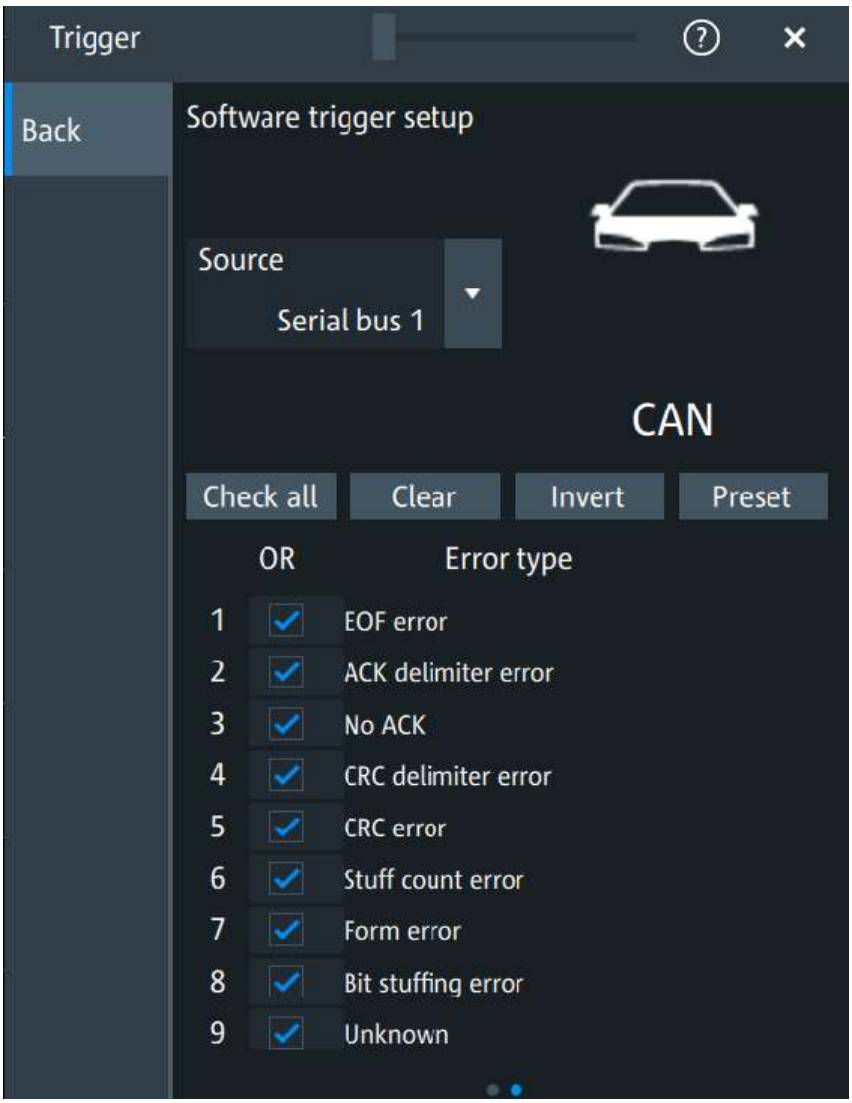
"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`TRIGger:SBSW:CAN:DMAX` on page 1687
`TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DMAX` on page 1687
`TRIGger:SBSW:CAN:DMIN` on page 1687
`TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DMIN` on page 1687
`TRIGger:SBSW:CAN:DOPerator` on page 1688
`TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DOPerator` on page 1688

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:CAN:IMAX on page 1688
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IMAX on page 1688
TRIGger:SBSW:CAN:IMIN on page 1689
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IMIN on page 1689
TRIGger:SBSW:CAN:IOPerator on page 1689
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IOPerator on page 1689

**Error type**
Enables triggering on the selected error type.




Remote command:
TRIGger:SBSW:CAN:ERENable on page 1689
TRIGger:SBSW:CAN:ERRor<m>:ENABle on page 1689

### 14.10.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.10.7 CAN decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

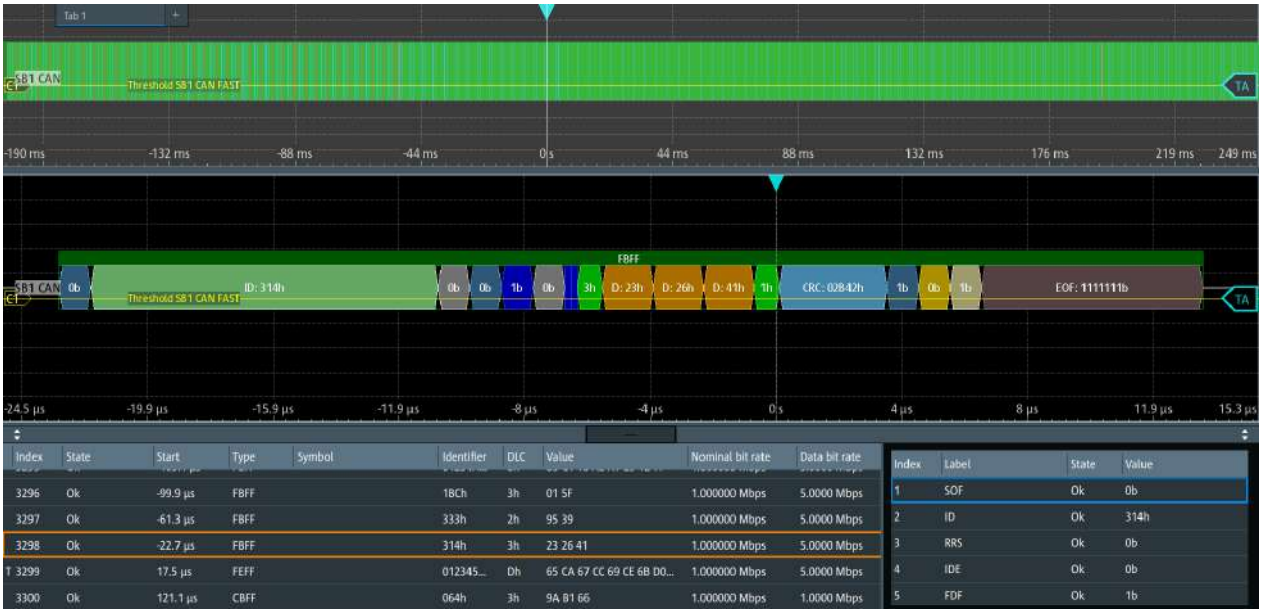



***Figure 14-32: Decoded CAN signal***

The decode results table contains information about all decoded frames.

***Table 14-16: Content of the decode results table***

| Column | Description |
|---|---|
| "Index" | Number of the decoded frame |
| "State" | Overall state of the frame.<br>"Insuffcient waveform length" indicates that the frame is not completely contained in the acquisition. Change the horizontal scale, or move the reference point to the left to get a longer acquisition. |
| "Start" | Time of frame start |
| "Type" | Frame type |
| "Symbol" | Symbolic label of the frame. Available only, if "Show symbols" is enabled in the "Display" tab and a symbols list is loaded. |
| "Identifier" | Identifier value in hexadecimal format. Select the data format in the "Display" tab. |
| "DLC" | Data length code, coded number of data bytes. Select the data format in the "Display" tab. |

| Column | Description |
|---|---|
| "Value" | Value of the data frame. Remote frames do not transmit data, therefore "- - -" is displayed. Select the data format in the "Display" tab. |
| "Nominal bit rate" | Number of bits per second |
| "Data bit rate" | Number of data bits per second |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

***Table 14-17: Content of the frame details table***

| Column | Description |
|---|---|
| "Index" | Number of the decoded field |
| "Label" | Field type. Values depend on the decoded frame "Type", see Table 14-18. |
| "State" | Overall state of the frame |
| "Value" | Value of the field |

Depending on the decoded type of frame, the following fields are available:

***Table 14-18: Decoded fields, depending on the frame type***

| Frame | Fields |
|---|---|
| CBFF/CBFF-R | CRC: cyclic redundancy check |
| CEFF/CEFF-R | CRC: cyclic redundancy check |
| FBFF/FEFF | BRS: bit rate switch |
| | ESI: error state indicator |
| | SBC: stuff bit count |
| | CRC: cyclic redundancy check |
| XLFF | SBC: stuff bit count |
| | SDT: service data unit |
| | SEC: simple extended content |
| | VCID: virtual CAN network ID |
| | AF: acceptance field |
| | PCRC: preamble cyclic redundancy check |
| | FCRC: frame CRC |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.11.6, "Decode results", on page 1690.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example of a CAN export file

```
Index,Start,Stop,State,Type,ID value,DLC,Nominal bit rate,Data bit rate,CRC,BRS,ESI,
  SBC,SDT,SEC,VCID,AF,PCRC
1,-0.19,-0.188,'OK','XLFF',064h,7FFh,1000000,10000000,0000AABBh,-,-,6h,FFh,1b,AAh,
  000ABCDh,01FFh
2,-0.188311,-0.188243,'OK','CBFF',064h,3h,1000000,1000000,35CEh,-,-,-,-,-,-,-,-
3,-0.188239,-0.188195,'OK','CBFR',064h,3h,1000000,1000000,5AA0h,-,-,-,-,-,-,-,-
4,-0.188191,-0.188157,'OK','FBFF',064h,2h,1000000,5000000,035FFh,0b,1b,Eh,-,-,-,-,-
5,-0.188153,-0.188075,'OK','CEFF',01401B26h,2h,1000000,1000000,32A0h,-,-,-,-,-,-,-,-
6,-0.188071,-0.188008,'OK','CEFR',01900064h,3h,1000000,1000000,32AAh,-,-,-,-,-,-,-,-
7,-0.188004,-0.187951,'OK','FEFF',01900064h,3h,1000000,5000000,032AAh,1b,0b,Fh,-,-,-,-,-
8,-0.187947,-0.187899,'OK','XLFF',064h,046h,1000000,10000000,000032AAh,-,-,6h,03h,0b,0Ah,
  0000ABCFh,1FFFh
9,-0.187895,-0.187881,'INC','ERR',---,---,1000000,1000000,-,-,-,-,-,-,-,-,-
10,-0.187877,-0.187863,'UNKN','OVER',---,---,1000000,1000000,-,-,-,-,-,-,-,-,-


Details frame 2
Index,Data
1,D0h
2,E7h
3,20h

Details frame 4
Index,Data
1,D1h
2,E8h

Details frame 5
Index,Data
1,D2h
2,E9h

Details frame 7
Index,Data
1,D3h
2,EAh
3,21h
```

# 14.11 LIN (automotive electronics, option R&S MXO3-K520)

The Local Interconnect Network (LIN) is a simple, low-cost bus system used within automotive network architectures. LIN is usually a subnetwork of a CAN bus. The primary purpose of LIN is the integration of uncritical sensors and actuators with low-bandwidth requirements. Common applications in a motor vehicle are the control of doors, windows, wing mirrors, and wipers.

**Requirements**

For performing LIN decode measurements, you need the following:

- MXO 3 with 1 available channel. The channel can be:
  - Analog channels (C1-C4)
  - Logic channels (D0-D15) (requires option R&S MXO3-B1)
- One probe
- Option R&S MXO3-K520

- About the LIN protocol ..... 704
- LIN configuration ..... 705
- LIN filter ..... 708
- LIN format configuration ..... 710
- LIN hardware trigger ..... 715
- LIN software trigger ..... 718
- Measure ..... 721
- LIN decode results ..... 721

## 14.11.1 About the LIN protocol

This section provides an overview of protocol characteristics, frame format, identifiers and trigger possibilities. For detailed information, order the LIN specification on http://www.lin-subbus.org/ (free of charge).

**LIN characteristics**

The main characteristics of LIN are:

- Single-wire serial communications protocol, based on the UART byte-word interface
- Single primary, multiple secondaries - usually up to 12 nodes
- Primary-controlled communication: primary coordinates communication with the LIN schedule and sends the identifier to the secondaries
- Synchronization mechanism for clock recovery by secondary nodes without crystal or ceramic resonator

**Data transfer**

Basic communication concept of LIN:

- Communication in an active LIN network is always initiated by the primary.

- The primary sends a message header including the synchronization break, the synchronization byte, and the message identifier.
- The identified node sends the message response: one to eight data bytes and one checksum byte.
- Header and response form the message frame.

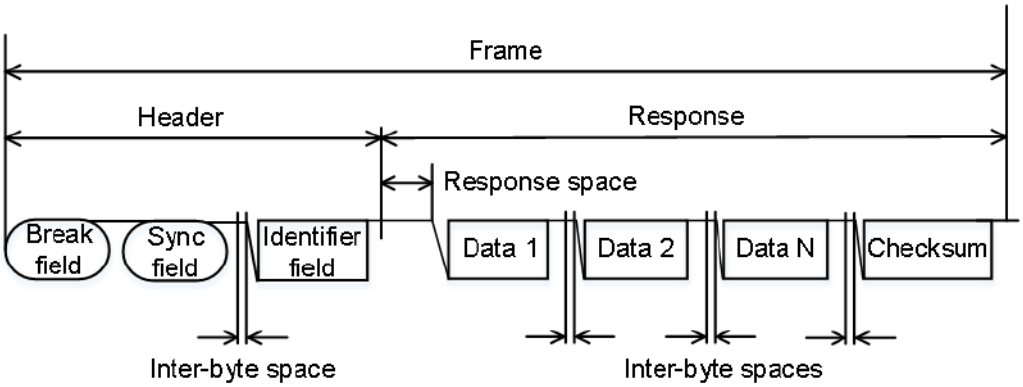



*Figure 14-33: LIN frame with header and response*

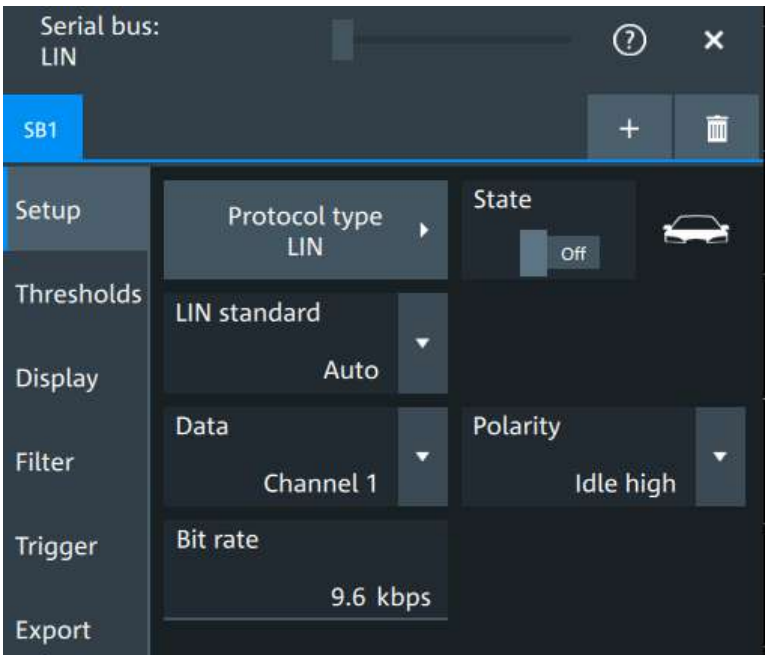
Data bytes are transmitted with LSB first.

The identifier byte consists of 6 bits for the frame identifier and two parity bits. This combination is known as protected identifier.

## 14.11.2 LIN configuration

### 14.11.2.1 LIN configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "LIN" > "Setup".




**LIN standard**
Selects the version of the LIN standard that is used in the DUT. The setting mainly defines the checksum version used during decoding.

The most common version is LIN v.2.x. For mixed networks, or if the standard is unknown, set the LIN standard to "Auto".

Remote command:
`SBUS<sb>:LIN:STANdard` on page 1705

**Data**
Sets the source waveform of the data line.

Remote command:
`SBUS<sb>:LIN:DATA:SOURce` on page 1704

**Polarity**
Defines the idle state of the bus. The idle state is the recessive state and corresponds to a logic high.

Remote command:
`SBUS<sb>:LIN:POLarity` on page 1705

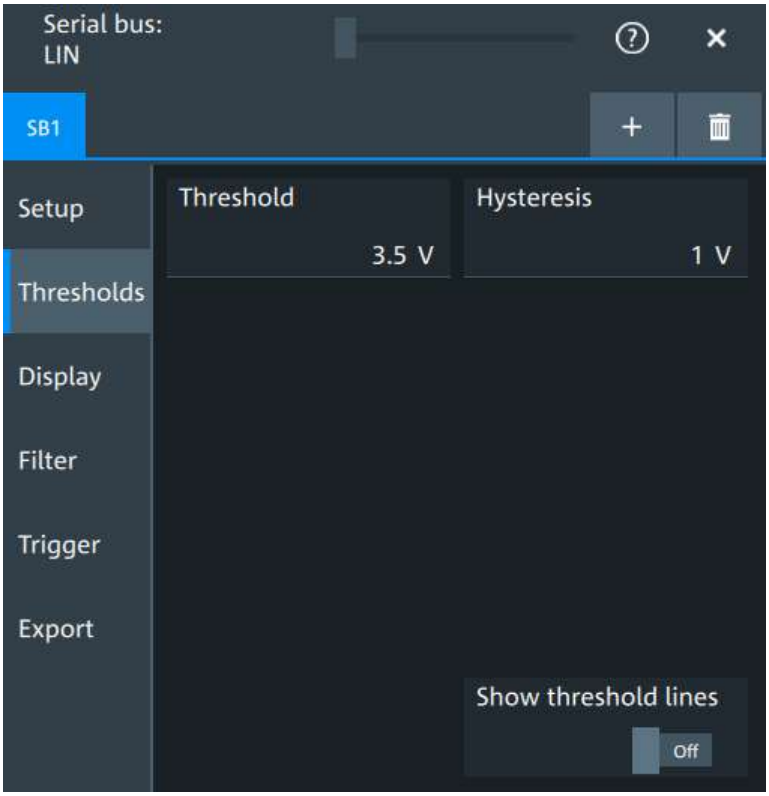
**Bit rate**
Sets the number of transmitted bits per second. The maximum bit rate for LIN is 20 kbit/s.

If the LIN standard is "J2602", the bit rate is 10.417 kbit/s and cannot be changed.

Remote command:
`SBUS<sb>:LIN:BITRate` on page 1703

### 14.11.2.2 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "LIN" > "Thresholds".




**Threshold**
Sets the threshold for the data channel. Enter the value directly in the fields.

Additional to the threshold, you can also set a hysteresis.

Remote command:
`SBUS<sb>:LIN:DATA:THReshold` on page 1704
`SBUS<sb>:LIN:DATA:HYSTeresis` on page 1704

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

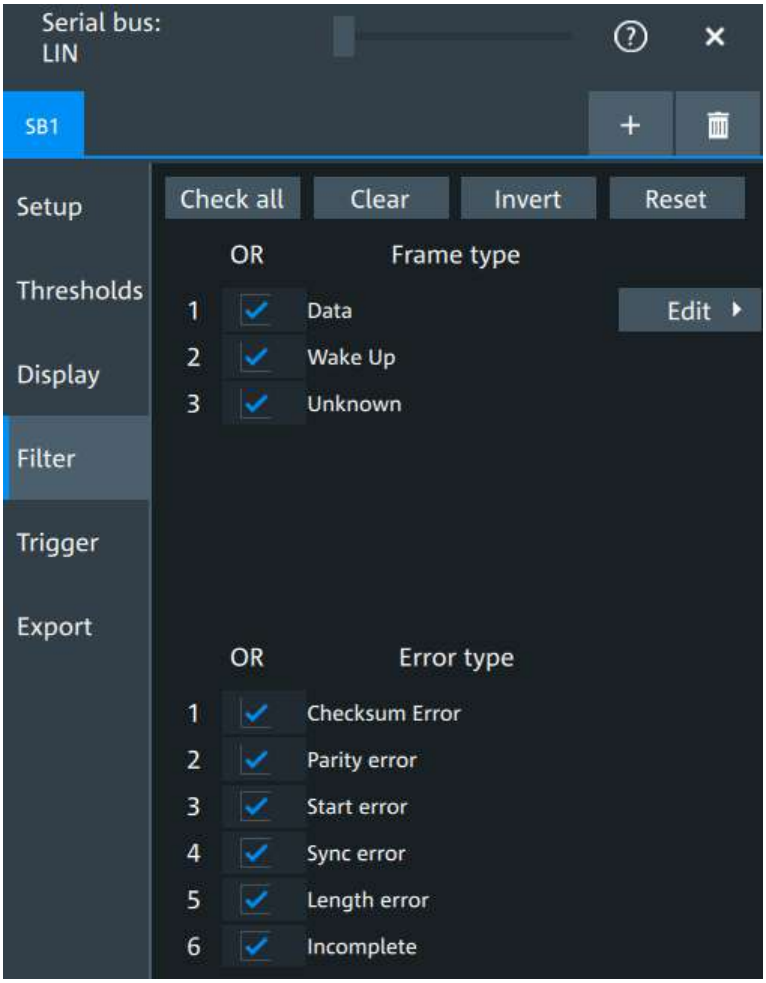
Remote command:
`SBUS<sb>:THReshold` on page 1364

### 14.11.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

### 14.11.3 LIN filter

Access: "Menu" > "Apps" > "Protocol" tab > "LIN" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
`SBUS<sb>:LIN:FILTer:CHKall` on page 1707
`SBUS<sb>:LIN:FILTer:CLR` on page 1707

SBUS<sb>:LIN:FILTer:INVert on page 1707
SBUS<sb>:LIN:FILTer:RST on page 1708

**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

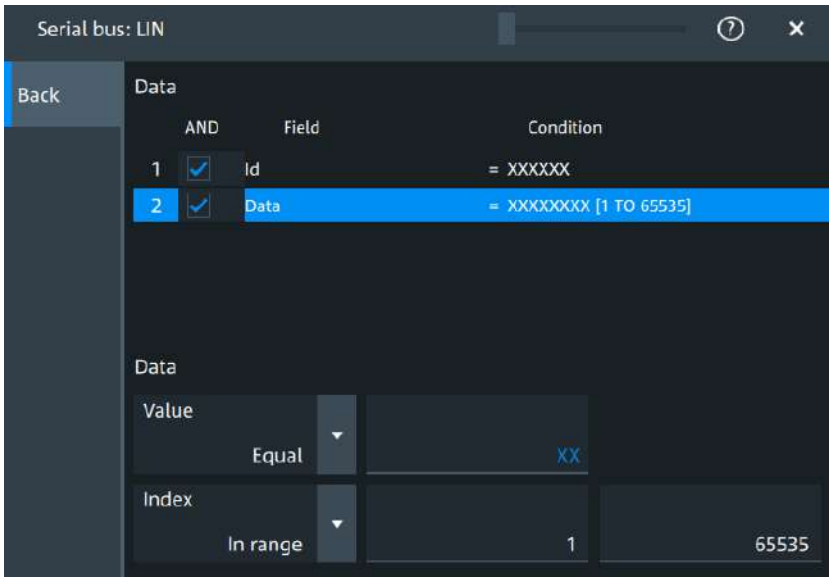
For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

Available frames are "Data", "Wake" and "Unknown".

Remote command:
SBUS<sb>:LIN:FILTer:FRENable on page 1710
SBUS<sb>:LIN:FILTer:FRAMe<fr>:ENABle on page 1710

**Edit**
Opens a dialog to define the details of the selected frame.

"Field" Enables the field type that you want to filter on for the selected frame.




The available fields are "Id" and "Data".

Remote command:
SBUS<sb>:LIN:FILTer:FIENable on page 1710
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:ENABle on page 1710

"Condition" Displays the value condition for the selected field.

Remote command:
SBUS<sb>:LIN:FILTer:BIT on page 1708
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:BIT on page 1708

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:LIN:FILTer:DMAX on page 1708
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1708
SBUS<sb>:LIN:FILTer:DMIN on page 1709
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1709
SBUS<sb>:LIN:FILTer:DOPerator on page 1709
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1709

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:LIN:FILTer:IMAX on page 1710
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1710
SBUS<sb>:LIN:FILTer:IMIN on page 1711
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1711
SBUS<sb>:LIN:FILTer:IOPerator on page 1711
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1711

**Error type**
Enables filtering on the selected error type.

The available errors are "Checksum error", "Parity error","Start error", "Sync error", "Length error" and "Incomplete".

Remote command:
SBUS<sb>:LIN:FILTer:ERENable on page 1709
SBUS<sb>:LIN:FILTer:ERRor<n>:ENABle on page 1709

### 14.11.4 LIN format configuration

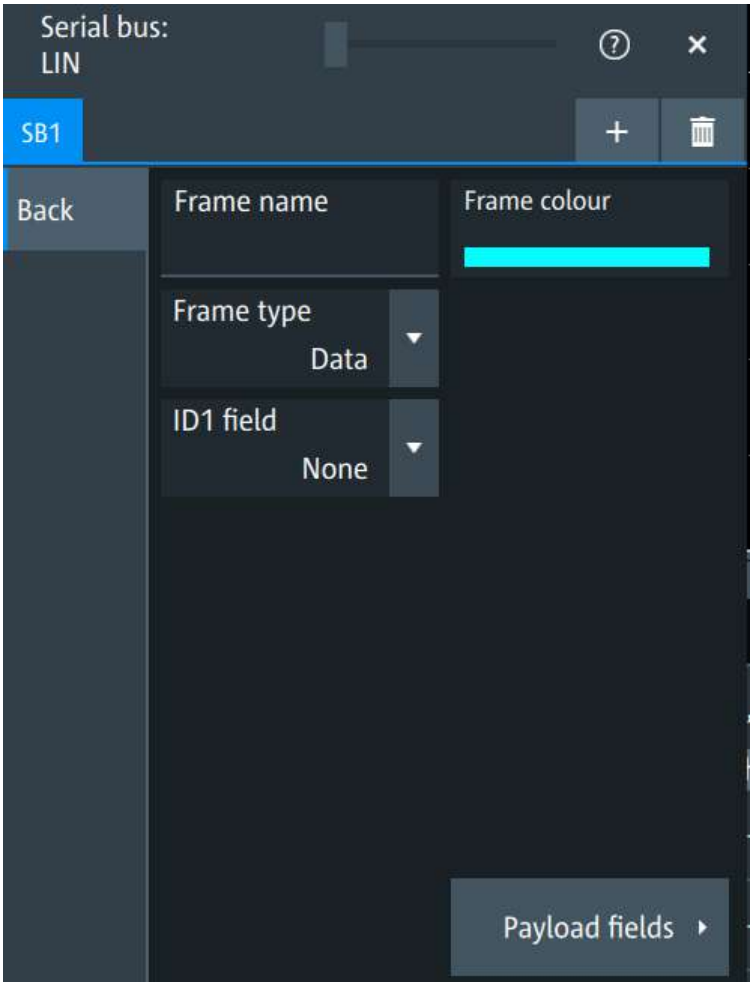
Access: "Menu" > "Apps" > "Protocol" tab > "LIN" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
SBUS<sb>:LIN:FORMat:SELect on page 1712

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:LIN:FORMat:ADDFrame on page 1713

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:LIN:FORMat:CLR on page 1713
SBUS<sb>:LIN:FORMat:FCOunt? on page 1714

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:LIN:FORMat:LOAD on page 1713

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:LIN:FORMat:SAVE on page 1713

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:NAME on page 1714

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:COLor on page 1715

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:TYPE on page 1715

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID1Field on page 1715
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID2Field on page 1715
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID3Field on page 1715

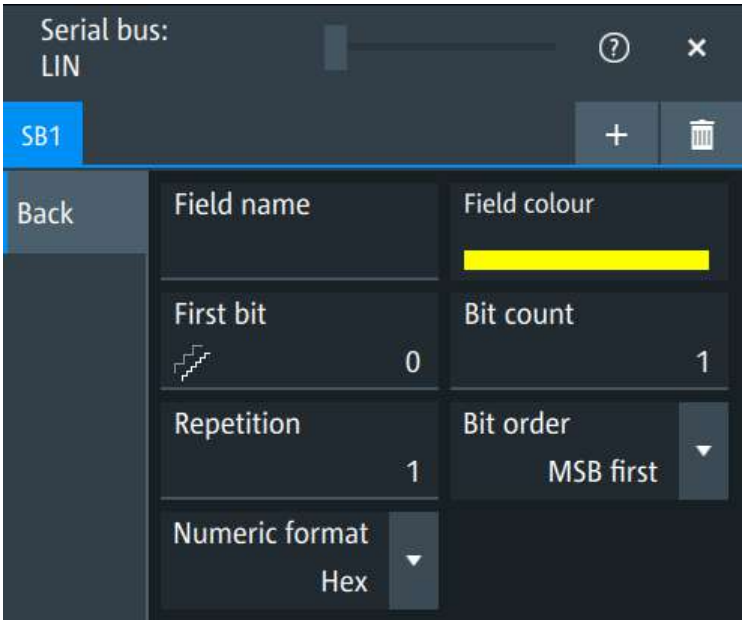
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID1Value on page 1716
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID2Value on page 1716
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID3Value on page 1716

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ADDField on page 1714

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1716

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1716

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1717

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.
- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  - Bit count = 2: truncated. Result: =00
  - Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  - Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  - Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1717

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
`SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:REPetition` on page 1717

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
`SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:BITorder` on page 1718

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:
- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  - "=1HL111000": valid (read as binary)
  - "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  - "=1010": valid (read as HEX, with a total of 16 bits)

– "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)
– "=5X12": invalid
– "=1H33": invalid

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1718

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1719

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1719

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1720

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1720

**Parameter A,B**
Defines the parameters of the linear function.

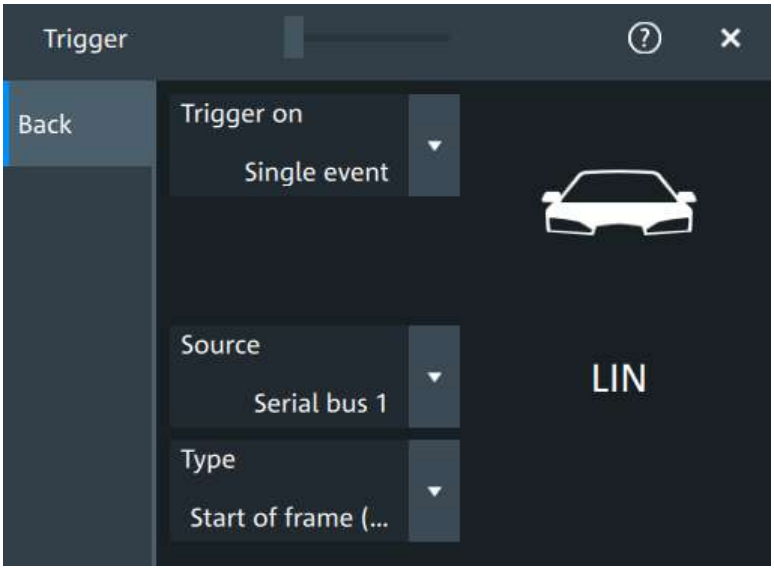
- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1720
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1720

### 14.11.5 LIN hardware trigger

Access: "Menu" > "Apps" > "Protocol" tab > "LIN" > "Trigger" tab > "Setup Hardware Trigger"




**Type**
Selects the trigger type for LIN analysis.

Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.

"Start of frame (Sync)" Triggers on the stop bit of the sync field.

"Identifier" Sets the trigger to one specific identifier or an identifier range. Enter only the 6-bit identifier without parity bits, not the protected identifier. Description of trigger type specific settings: Identifier setup: Condition, Frame ID min, Frame ID max.




"Identifier + Data" Sets the trigger to a combination of identifier and data condition. The instrument triggers at the end of the last byte of the specified data pattern.




The identifier conditions are the same as for the "Identifier" trigger type, see Identifier setup: Condition, Frame ID min, Frame ID max. Data conditions are set with Data setup: Condition, Pattern, Position.

"Wake-up frame" Triggers after a wake-up frame.

"Error condition" Identifies various errors in the frame, see Error conditions.

Remote command:
`TRIGger:SBHW:LIN:TYPE` on page 1721

**Identifier setup: Condition, Frame ID min, Frame ID max**
The identifier setup consists of the condition and one or two identifier pattern.

Condition Sets the comparison condition to a specific value or a range.

"Frame ID min" Defines the bit pattern of the identifier. Enter only the 6-bit identifier without parity bits, not the protected identifier.

"Frame ID max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger:SBHW:LIN:IMIN` on page 1722
`TRIGger:SBHW:LIN:IMAX` on page 1723
`TRIGger:SBHW:LIN:ICONdition` on page 1723

**Data setup: Condition, Pattern, Position**
The data setup consists of the the condition, the position of the pattern, and one data pattern.

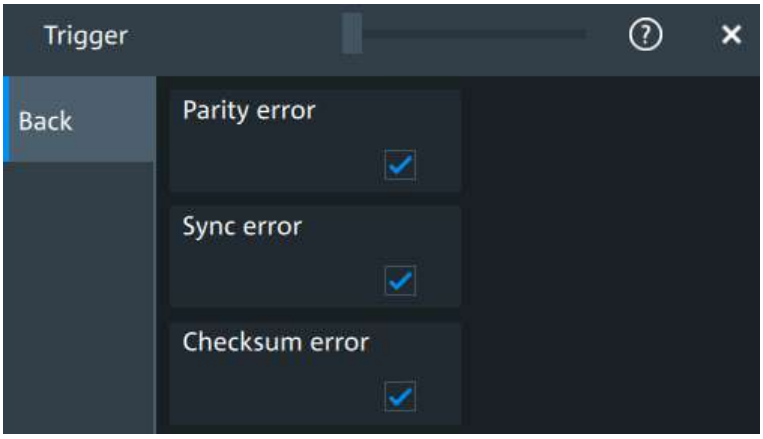
Condition Sets the operator to define a specific data pattern ("Equal" or "Not equal") or a range smaller or bigger than a specific pattern.

"Pattern" Defines the data pattern.

"Position" Sets the number of data events that are ignored, before trigger condition check of the data starts.

Remote command:
`TRIGger:SBHW:LIN:DCONdition` on page 1724
`TRIGger:SBHW:LIN:DMIN` on page 1723
`TRIGger:SBHW:LIN:DPOSition` on page 1723

**Error conditions**
Triggers if one or more of the following errors occur.




"Parity error" Parity bits are the bits 6 and 7 of the identifier. They verify the correct transmission of the identifier.

"Sync error" Synchronization error.

"Checksum error" The checksum verifies the correct data transmission. It is the last byte of the frame response. The checksum includes not only the data but also the protected identifier (PID).

Remote command:
`TRIGger:SBHW:LIN:IPERror` on page 1722
`TRIGger:SBHW:LIN:SYERror` on page 1722
`TRIGger:SBHW:LIN:CHKSerror` on page 1724

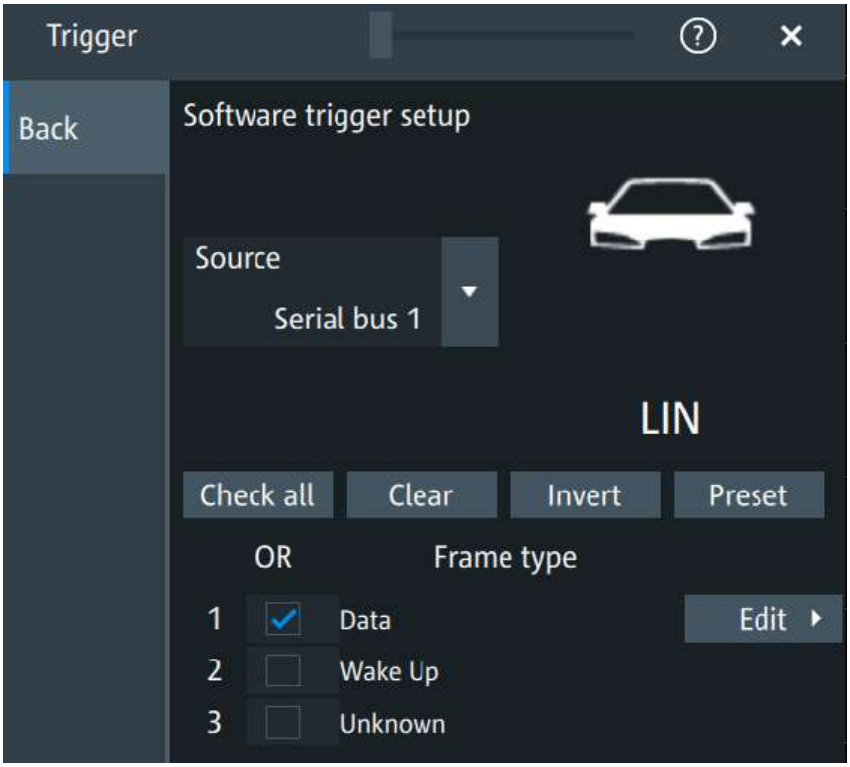
## 14.11.6 LIN software trigger

### 14.11.6.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.11.6.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "LIN" > "Trigger" tab > "Setup Software Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:LIN:CHKall` on page 1725
`TRIGger:SBSW:LIN:CLR` on page 1725
`TRIGger:SBSW:LIN:INVert` on page 1725
`TRIGger:SBSW:LIN:RST` on page 1725

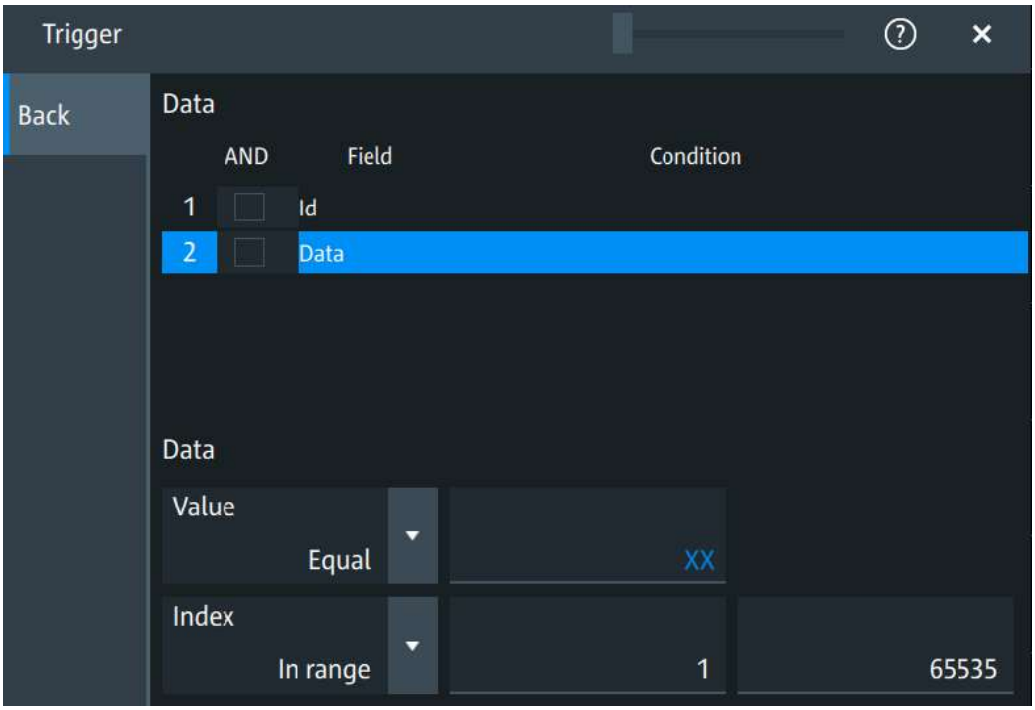
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: "Data", "Wake Up", "Unknown".

Remote command:
TRIGger:SBSW:LIN:FRENable on page 1726
TRIGger:SBSW:LIN:FRAMe<fr>:ENABle on page 1726

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.




"Field" Enables the field type that you want to trigger on for the selected frame.
The following fields are available:

Remote command:
TRIGger:SBSW:LIN:FIENable on page 1727
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:ENABle on page 1727

"Condition" Displays the value condition for the selected field.

Remote command:
TRIGger:SBSW:LIN:BIT on page 1726
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:BIT on page 1726

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
TRIGger:SBSW:LIN:DMAX on page 1726
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DMAX on page 1726
TRIGger:SBSW:LIN:DMIN on page 1727
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DMIN on page 1727
TRIGger:SBSW:LIN:DOPerator on page 1727
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DOPerator on page 1727

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:LIN:IMAX on page 1728
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IMAX on page 1728
TRIGger:SBSW:LIN:IMIN on page 1728
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IMIN on page 1728
TRIGger:SBSW:LIN:IOPerator on page 1729
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IOPerator on page 1729

**Error type**
Enables triggering on the selected error type.

The following error types are available:"Checksum error", "Parity error","Start error", "Sync error", "Length error" and "Incomplete".

Remote command:
TRIGger:SBSW:LIN:ERENable on page 1729
TRIGger:SBSW:LIN:ERRor<m>:ENABle on page 1729

### 14.11.7 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.11.8 LIN decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

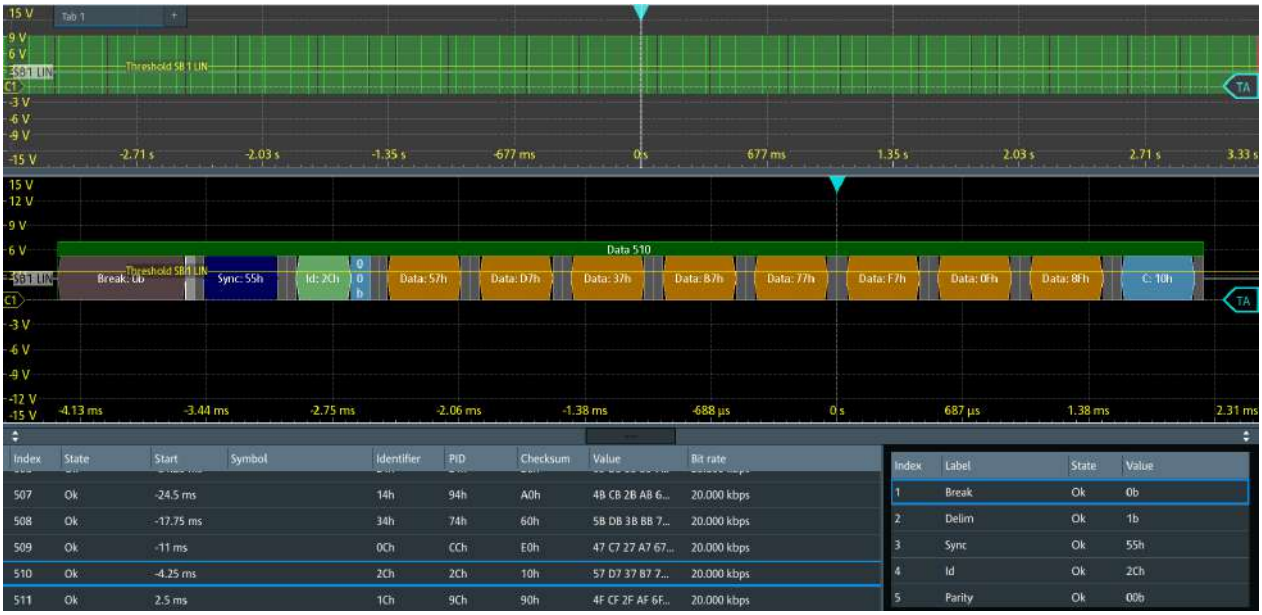



*Figure 14-34: Decoded LIN signal*

The decode results table contains information about all decoded frames.

*Table 14-19: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Index of the decoded frame |
| State | Overall state of the frame |
| Start | Time of frame start |
| Symbol | Symbolic label of the frame. Available only, if "Show symbols" is enabled in the "Display" tab and a symbols list is loaded. |
| Identifier | Identifier value |
| PID | Protected identifier |
| Checksum | Checksum value |
| Values | Value of the data bytes. Select the data format in the "Display" tab. |
| Bit rate | Value of the bit rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

*Table 14-20: Content of the frame details table*

| Column | Description |
|---|---|
| Index | Index of the field |
| Label | Name of the field |
| State | State of the field |
| Value | Value of the field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.12, "LIN (option R&S MXO3-K520)", on page 1703.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example of LIN export file

```
Index,Start,Stop,State,Identifier,PID,Checksum,Bit rate
1,-3.44,-3.43375,'OK',02h,C2h,D0h,20000
2,-3.43325,-3.427,'OK',22h,22h,30h,20000
3,-3.4265,-3.42025,'OK',12h,92h,B0h,20000
4,-3.41975,-3.4135,'OK',32h,72h,70h,20000
5,-3.413,-3.40675,'OK',0Ah,CAh,F0h,20000

Details frame 1
Index,State,Value
1,'OK',40h
2,'OK',C0h
3,'OK',20h
4,'OK',A0h
5,'OK',60h
6,'OK',E0h
7,'OK',10h
8,'OK',90h

Details frame 2
Index,State,Value
1,'OK',50h
2,'OK',D0h
3,'OK',30h
4,'OK',B0h
5,'OK',70h
6,'OK',F0h
7,'OK',08h
8,'OK',88h
```

## 14.12 SENT (automotive electronics, option R&S MXO3-K520)

Single Ended Nibble Transmission (SENT) is a serial transmission interface protocol originally specified for the communication of sensors and control units in automotive electronics.

SENT is a protocol standard governed by the Society of Automotive Engineers (SAE J2716). For detailed information, refer to the SENT standard specification on http://www.sae.org.

The SENT protocol is used exclusively in automotive applications. Examples are electrical power steering, and advanced driver assistance such as parking assist or sensing of pressure, throttle position, pedal position, airflow mass, liquid level, etc.

**Requirements**

For performing SENT decode measurements, you need the following:

- MXO 3 with 1 available channel. The channel can be:
  - Analog channels (C1-C4)
  - Logic channels (D0-D15) (requires option R&S MXO3-B1)
- One probe
- Option R&S MXO3-K520

- About the SENT protocol ..... 724
- SENT configuration ..... 728
- SENT filter ..... 732
- SENT format configuration ..... 736
- SENT software trigger ..... 741
- Measure ..... 745
- SENT decode results ..... 745

### 14.12.1 About the SENT protocol

This section provides an overview of the protocol characteristics, encoding scheme, identifiers and trigger possibilities.

The SENT protocol transmits signal values point-to-point from a sensor to a controller (electronic control unit ECU), unidirectional. In contrast to conventional measurements, you can receive multiple data parameters via the SENT interface in a single transmission. Nevertheless, SENT is characterized by its simplicity and yet high customizability to meet the individual requirements of the applications.

SENT operates via a three-wire connection, a signal line, a supply voltage line for the sensor and a ground line. It transmits data digitally in variable timing units and evaluates the time between two falling edges (single edges). The signal is amplitude modulated with a constant amplitude voltage. Thus the influences of disturbance signals are not critical.

**SENT key features**

The main characteristics of SENT are:

- Serial communication protocol
- 3 wires: SENT (signal line), 5V (voltage line), GND (ground line)
- Output only, from sensor to receiver
- Point-to-point transmission, no bus
- Digital transmission
- High baud rate
- Data transmission in variable timing units of 4 bits (1 nibble) between two falling edges
- Transmitter-specific clock period (tick)
- Time measured between single falling edges

#### 14.12.1.1 SENT transmission concept

A sensor converts the analog measured data to a digital signal, and thus transmits a series of pulses to the receiver. The receiver, e.g. an ECU processes the received signal also digitally.

The format of a SENT message frame has a fixed pulse order and a transmitter-specific clock period. The total transmission time varies depending on the clock variation of the transmitter and the transmitted data values. The data pulses embedded in the transmission sequence represent one or multiple data parameters to be communicated. The last pulses in a message frame are the CRC check pulse, allowing the receiver to perform some diagnostic tests, and an optional pause pulse.

A SENT transmission starts without a request from the receiver. Consecutive sequences are transmitted continuously after the falling edge of the last pulse.

The SENT protocol distinguishes between two channel types:

- **Fast channel:** transmits primary data, i.e. sensor readings like temperature, pressure, mass air flow, throttle position.
- **Slow channel:** transmits secondary data consisting of transfer characteristics, sensor ID, type, manufacturer diagnostic, etc.
  The slow channel transmission provides two serial message formats *Short* and *Enhanced* for customizing the secondary data.

The data of both channels is transmitted simultaneously, by including two bits of a slow channel message in the message frame of the fast channel. Even though it requires many fast channel messages to complete a slow channel message, you can use this function to transmit several slow channel messages with minimal impact on the primary sensor data and the data rate.

#### 14.12.1.2 SENT message definitions

**SENT terms**

See the specific terms and definitions used in the SENT protocol:

- **Tick (clock tick):** basic unit of time
  - Transmitter-specific nominal clock period
  - 3 µs < clock tick <90 µs, with max. 20 % clock variation
- **Nibble:** minimum unit of data
  - Used to transmit data
  - Variable timing units between two falling edges

**SENT fast channel**

The SENT protocol enables you to transmit measurements of multiple sensors in one transmission sequence with data signals of varying length. The diagram in Figure 14-35 shows, for example, the encoding scheme for two 12-bit data signals.

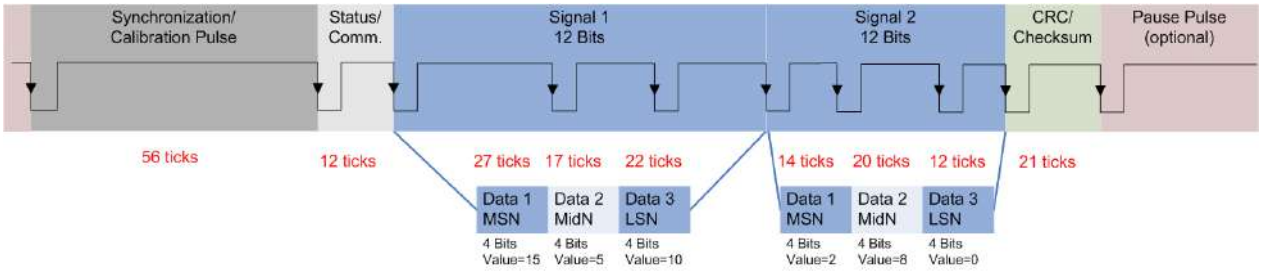



*Figure 14-35: Example of a SENT transmission sequence*

The format of a SENT transmission sequence consists of the following pulses:

- **Synchronization/Calibration pulse:**
  - Initial sequence of the receiver
  - The start condition is the falling edge of the last pulse (CRC or Pause)
  - The nominal pulse period is 56 clock ticks
  - Measures the actual clock variation of the transmitter and calculates the tick timing
- **Status/Communication Pulse (Nibble)**
  - One 4 bit pulse
  - Communicates status and enables the sensor to include slow channel message bits
    0: (LSB) specific application
    1: specific application
    2: Serial Data message or specific application (e.g. Infineon TLE4998S)
    3: (MSB) 1= message start; 0=Serial Data message or specific application (e.g. Infineon TLE4998S)
  - 12 to 27 clock ticks
  - Not included in CRC frame calculation

- **Data Pulses (Nibbles)**
  - One up to six 4 bit data nibbles
  - 12 to 27 clock ticks pulse period
  - Initial logic 0 time with ≥5 ticks, subsequent logical 1 with variable duration
- **CRC/Checksum**
  - One 4 bit pulse
  - Used for error checking of data nibbles (status nibble not included)
  - Detects single bit, odd number of nonconsecutive and single burst errors
- **Pause Pulse**
  - One optional pulse
  - Variable pulse length: 12 to 768 clock ticks
  - Can be used to create a transmission with a constant number of clock ticks

### SENT slow channel

#### Short Serial Messages

For transmission of a slow channel message, 2 bits are included in a fast channel message, see the status nibble (Bit 2,3) in Figure 14-36.

A short serial message needs 16 fast channel messages until it is completely transmitted. Prerequisite for the complete transmission of the slow channel message is 16 consecutive error-free fast channel transmissions.

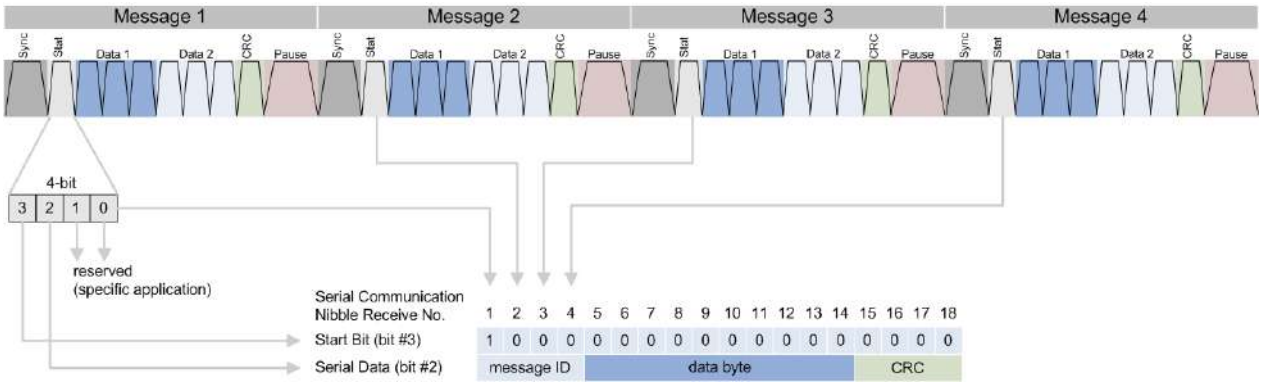



*Figure 14-36: One serial message, composed of 16 SENT consecutive fast channel transmissions*

#### Enhanced Serial Messages

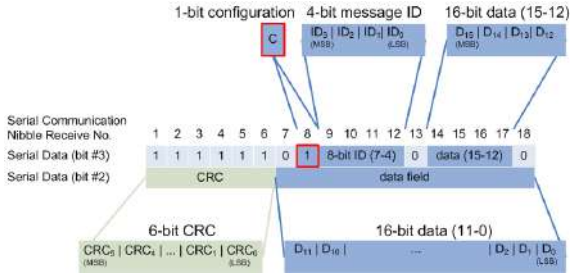
The transmission of an enhanced serial message format requires 18 fast channel transmissions. Each slow channel message is assigned a message ID, which is transmitted with the data.

The enhanced serial message format provides two alternatives for configuring the message:

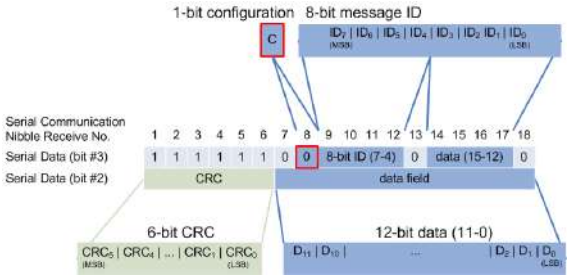
- 4-bit ID and 16-bit data
- 8-bit ID and 12-bit data

The graphs below illustrate the variants.

***Table 14-21: Enhanced serial message formats***




**16-bit data and 4-bit message ID**

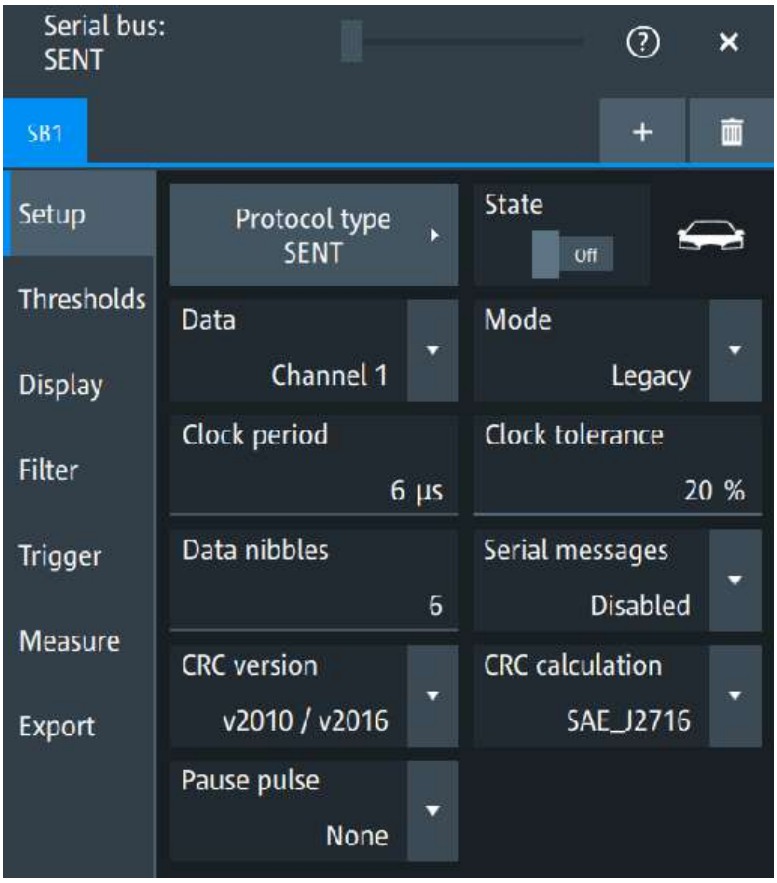



**12-bit data and 8-bit message ID**

## 14.12.2 SENT configuration

### 14.12.2.1 SENT configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "SENT" > "Setup".



Make sure that the tab of the correct serial bus is selected.

**Protocol type**
Selects the protocol type to be decoded.

Remote command:
SBUS<sb>:TYPE on page 1362

**Data**
Sets the source of the data line.

Remote command:
SBUS<sb>:SENT:DATA:SOURce on page 1738

**Mode**
Selects the operating mode of the SENT protocol.

If "Mode" = "Legacy", the "SPC" mode is not supported.

The single point calibration ("SPC") mode was introduced in the SENT 2.0 specification. If "SPC" mode is selected, the SENT protocol allows for the calibration of a sensor's output by sending a single calibrated value. It is particularly useful for applications where a sensor needs to be calibrated to ensure accurate readings. The SPC mode typically involves the transmission of a single data frame that contains the calibrated value, which can then be used by the receiving device to interpret the sensor's output correctly.

Remote command:
SBUS<sb>:SENT:MODE on page 1739

**Clock period**
Sets the transmitter-specific nominal clock period (clock tick).

The clock period and signal length determine the speed of transmission.

Remote command:
SBUS<sb>:SENT:CLKPeriod on page 1736

**Clock tolerance**
Specifies a tolerated deviation of the clock.

Remote command:
SBUS<sb>:SENT:CLKTolerance on page 1737

**Data nibbles**
Sets the number of data units in a single transmission sequence.

The maximum number of data nibbles is 6.

Remote command:
SBUS<sb>:SENT:DNIBbles on page 1739

**Serial message**
Selects if serial messages are enabled or disabled.

Remote command:
SBUS<sb>:SENT:SFORmat on page 1741

**CRC version**
Selects the version that the CRC check is based on.

"Legacy" Based on the CRC calculation version used earlier than 2010.

"v2010/v2016" Based on the CRC calculation version updated in 2010/2016.

Remote command:
SBUS<sb>:SENT:CRCVersion on page 1737

**CRC calculation**
Selects the method for CRC calculation.

"SAE_J2716" Calculates the CRC according to the SAE standard.
For this method, the checksum is calculated over all nibbles except the communication and status nibble.

"TLE_4998X" Calculates the CRC according to the standard computing method for Infineon TLE_4998X sensors.
For this method, the checksum is calculated over all nibbles except the communication nibble.

Remote command:
SBUS<sb>:SENT:CRCMethod on page 1737

**Pause pulse**
Determines whether a pause pulse is transmitted after the checksum nibble.

You can use this pulse to create a transmission with a constant number of clock ticks. The pause pulse length can be between a minimum of 12 clock ticks up to 768 (3*256) ticks at a maximum.

Remote command:
SBUS<sb>:SENT:PPULse on page 1740

**Frame length**
Determines the frame length in terms of ticks. The dialog displays this settings parameter, if the signal has a constant frame length.

Remote command:
SBUS<sb>:SENT:PPFLength on page 1740

#### 14.12.2.2 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "SENT" > "Thresholds".

**Threshold**
Sets the threshold for the data channel. Enter the value directly in the fields.

Additional to the threshold, you can also set a hysteresis.

Remote command:
SBUS<sb>:SENT:DATA:THReshold on page 1738
SBUS<sb>:SENT:DATA:HYSTeresis on page 1738

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
SBUS<sb>:THReshold on page 1364

### 14.12.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

**Show symbols**
You can load symbol lists, and activate its usage for decoding. As a result, an additional "Symbol" column appears in the "Decode results" table, containing the symbolic label.

For details, see Section 14.12.2.4, "SENT symbols", on page 731.

### 14.12.2.4 SENT symbols

Label lists are protocol-specific. For the SENT protocol, you can upload symbol lists as in the format shown in the following example.

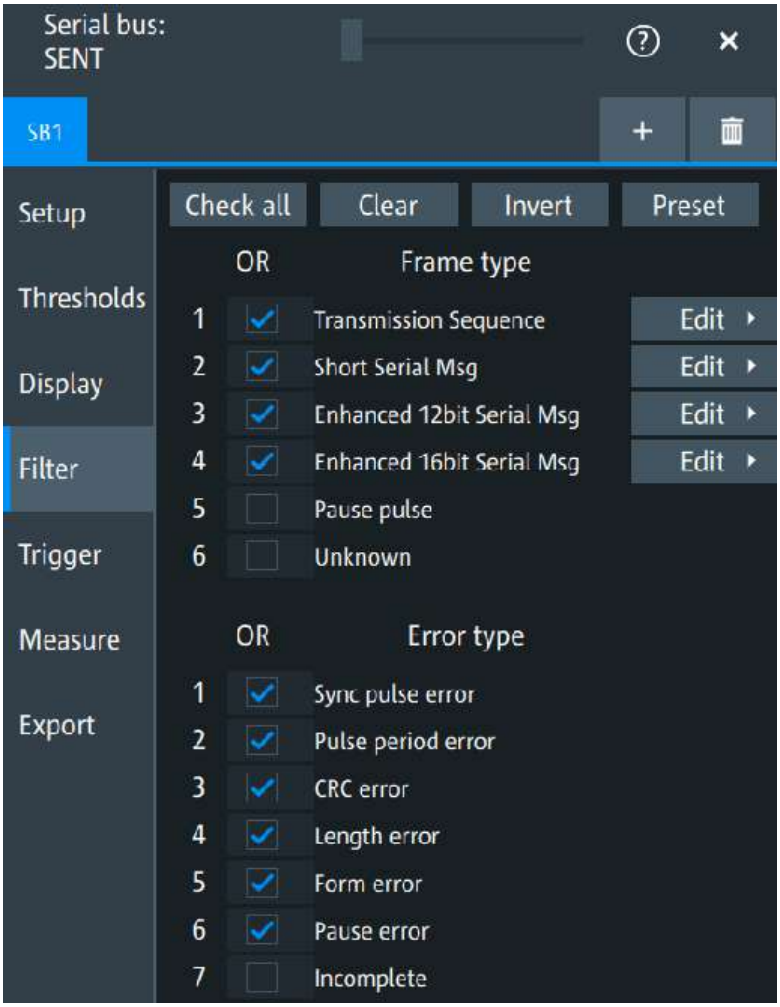
**Example: SENT symbols file**

```
<sb:FRAME NAME="Diagnostic Error Codes" STATE="ON">
  <!-- Start of a Frame Definition -->
  <!-- This block defines the information of a Transmission Sequence
   or Serial Message:
   NAME => Symbolic Label of the Frame
   STATE [ON/OFF] => When ON, this frame Translation is taken into consideration.
                     When OFF, this frame Translation is skipped.-->
<sb:DESCRIPTION> used to diagnose the current SENT System</sb:DESCRIPTION>
  <!-- Doesn't affect the Translation -->
<sb:ID-VALUE>01</sb:ID-VALUE>
  <!-- ID Value of the Serial Message (in decimal) -->
  <!-- Absence of the ID-VALUE field implies that the current Frame Translation
   is to be used for Transmission Sequences and not for a Serial Message -->
<sb:ID-LENGTH>8</sb:ID-LENGTH>
  <!-- ID Length of the Serial Message (in bits) -->
<sb:DATA-SIZE>12</sb:DATA-SIZE>
  <!-- Data Length of the Serial Message (in bits) -->
<sb:SIGNALS>
```

```
  <!-- This block defines the information of the Signals embedded
   in the Data Field of the Frame (Transmission Sequence or Serial Message) -->
<sb:SIGNAL ID="Diagnostic">
  <!-- Unique ID of the Signal (no effect on Translation) -->
<sb:SHORT-NAME>Diagnostic Code</sb:SHORT-NAME>
  <!-- Name of the Signal -->
<sb:DESCRIPTION></sb:DESCRIPTION>
  <!-- Info Field (no effect on Translation) -->
<sb:BIT-POSITION>11</sb:BIT-POSITION>
  <!-- Starting Bit position of the Signal
   (The whole Data Field is represented as MSB -> LSB Sequence) -->
<sb:BIT-LENGTH>12</sb:BIT-LENGTH>
  <!-- Number of Bits representing the Signal Value -->
<sb:BYTE-ORDER>MSB</sb:BYTE-ORDER>
  <!-- Byte Order of the Signal Value [MSB or LSB], Default: MSB -->
<sb:VALUE-TYPE>ENUM</sb:VALUE-TYPE>
  <!-- Representation of the Bits [ENUM, UNSIGNED_INT, INT, FLOAT, DOUBLE],
   Default: UNSIGNED_INT
   The Signal Value is calculated according to the following:
   Translated_Value = Encoded_Value * FACTOR + OFFSET -->
<sb:FACTOR>1.0</sb:FACTOR>
  <!-- Signal Factor (decimal value)-->
<sb:OFFSET>0.0</sb:OFFSET>
  <!-- Signal Offset (decimal value)-->
<sb:MIN>0</sb:MIN>
  <!-- Minimum Signal Value (decimal value) -->
<sb:MAX>4096</sb:MAX>
  <!-- Maximum Signal Value (decimal value) -->
<sb:ENUM-VALUES>
  <!-- This block is only valid (and taken into consideration)
   when the VALUE-TYPE is ENUM
   It defines the Enumeration List Translation of the Signal -->
<sb:ENUM INDEX="0" LABEL="No Error"/>
   <!-- INDEX is the Enum Value (corresponds to the Signal Value in decimal),
    LABEL is the matching Translated Signal Value -->
<sb:ENUM INDEX="1" LABEL="Channel 1 out of range high"/>
</sb:ENUM-VALUES>
  <!-- End of Signal Enumeration List Definition -->
</sb:SIGNAL>
  <!-- End of a Signal Definition -->
  <!-- More Signals can be defined here! -->
</sb:SIGNALS>
  <!-- End of list of Signals Definition -->
</sb:FRAME>
  <!-- End of Frame Definition -->
```

### 14.12.3 SENT filter

Access: "Menu" > "Apps" > "Protocol" tab > "SENT" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
`SBUS<sb>:SENT:FILTer:CHKall` on page 1743
`SBUS<sb>:SENT:FILTer:CLR` on page 1743
`SBUS<sb>:SENT:FILTer:INVert` on page 1744
`SBUS<sb>:SENT:FILTer:RST` on page 1744

**Enable**
Enables the filtering on SENT frames. Only the frames that match the selected filter conditions are displayed.

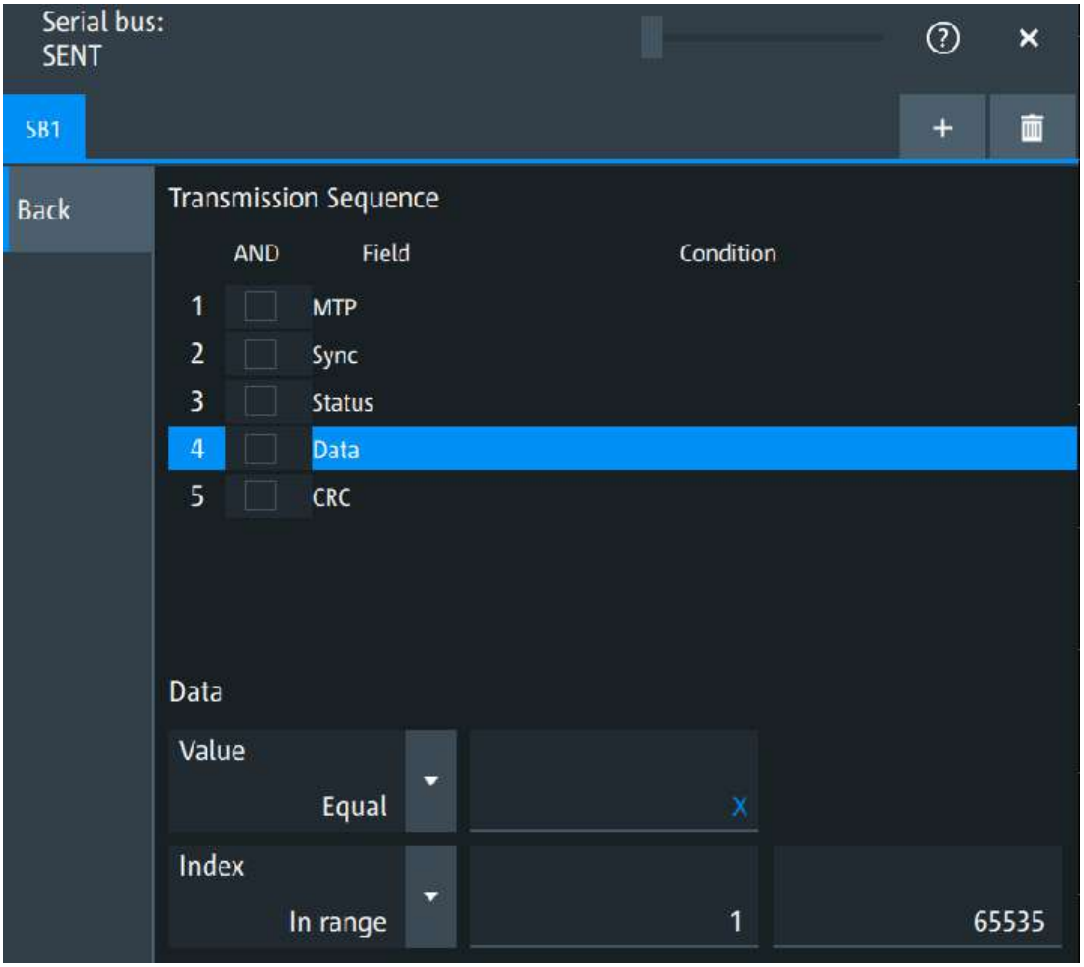
Remote command:
`SBUS<sb>:SENT:FILTer:FRENable` on page 1744
`SBUS<sb>:SENT:FILTer:FRAMe<fr>:ENABle` on page 1744

**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frame types are available: "Transmission Sequence", "Short Serial Msg", "Enhanced 12bit Serial Msg", "Enhanced 16bit Serial Msg", "Pause pulse", "Unknown".

**Edit**
Opens a dialog to define the details of the selected frame.



"Field" Enables the field type that you want to filter on for the selected frame. The following fields are available: "MTP", "Sync", "Status", "Data", "CRC", "ID".

Remote command:
SBUS<sb>:SENT:FILTer:FIENable on page 1746
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:ENABle on page 1746

"Condition" Displays the value condition for the selected field.

Remote command:
SBUS<sb>:SENT:FILTer:BIT on page 1744
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:BIT on page 1744

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:SENT:FILTer:DMAX on page 1745
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1745
SBUS<sb>:SENT:FILTer:DMIN on page 1745
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1745
SBUS<sb>:SENT:FILTer:DOPerator on page 1746
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1746

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:SENT:FILTer:IMAX on page 1746
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1746
SBUS<sb>:SENT:FILTer:IMIN on page 1747
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1747
SBUS<sb>:SENT:FILTer:IOPerator on page 1747
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1747

**Error type**
Enables filtering on the selected error type.

The available error types are "Sync pulse error", "Pulse period error", "CRC error", "Length error", "Form error", "Pause error", "Incomplete".

Remote command:
SBUS<sb>:SENT:FILTer:ERENable on page 1748
SBUS<sb>:SENT:FILTer:ERRor<n>:ENABle on page 1748

### 14.12.4 SENT format configuration

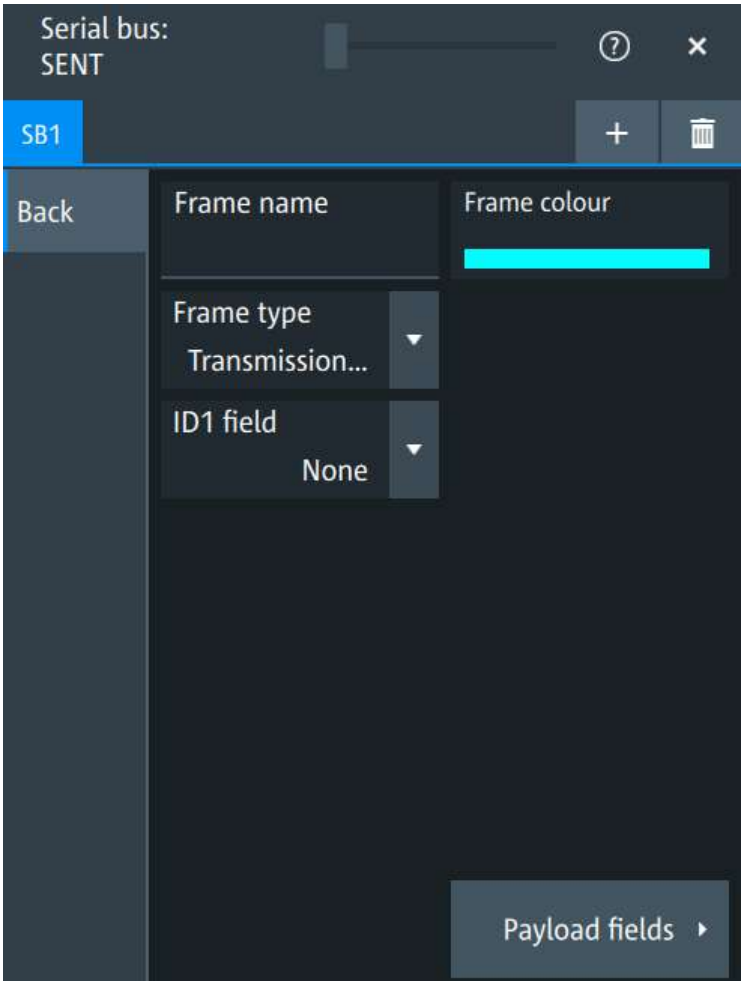
Access: "Menu" > "Apps" > "Protocol" tab > "SENT" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
`SBUS<sb>:SENT:FORMat:SELect` on page 1749

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
`SBUS<sb>:SENT:FORMat:ADDFrame` on page 1749

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:SENT:FORMat:CLR on page 1750
SBUS<sb>:SENT:FORMat:FCOunt? on page 1750

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:SENT:FORMat:LOAD on page 1749

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:SENT:FORMat:SAVE on page 1749

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:SENT:FORMat:FRAMe<fr>:NAME on page 1751

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:SENT:FORMat:FRAMe<fr>:COLor on page 1751

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:SENT:FORMat:FRAMe<fr>:TYPE on page 1751

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID1Field on page 1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID2Field on page 1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID3Field on page 1752

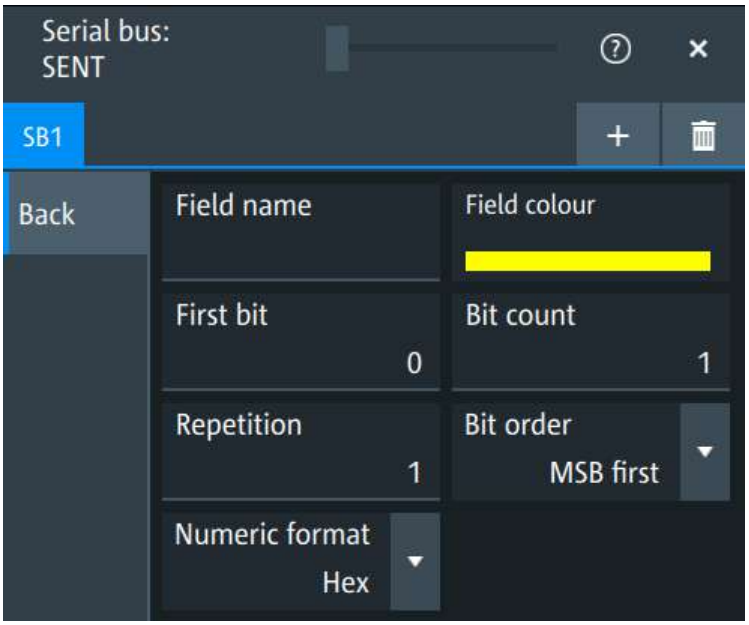
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID1Value on page 1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID2Value on page 1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID3Value on page 1752

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ADDField on page 1750

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1752

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1752

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1753

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.
- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  - Bit count = 2: truncated. Result: =00
  - Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  - Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  - Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:BITCount on page 1753

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:REPetition on page 1754

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:BITorder on page 1754

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".

- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:

- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  - "=1HL111000": valid (read as binary)
  - "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  - "=1010": valid (read as HEX, with a total of 16 bits)
  - "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)
  - "=5X12": invalid
  - "=1H33": invalid

Remote command:
`SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FORMat` on page 1754

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
`SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ADDValue` on page 1755

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
`SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue` on page 1756

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
`SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel` on page 1756

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
`SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:UNIT` on page 1756

**Parameter A,B**
Defines the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
`SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:PARA` on page 1757
`SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:PARB` on page 1757

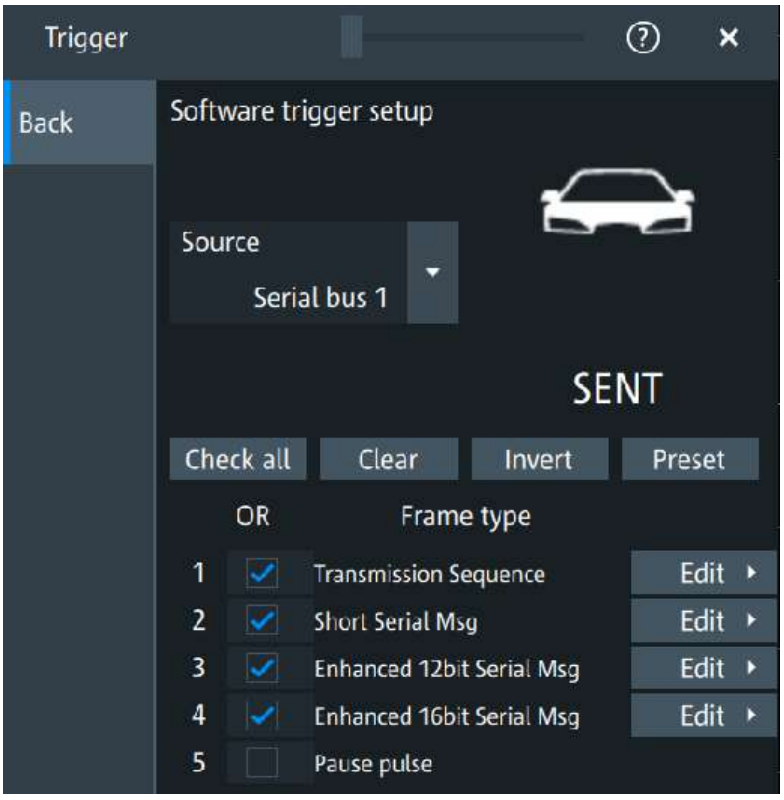
## 14.12.5 SENT software trigger

### 14.12.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.12.5.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "SENT" > "Trigger" tab > "Setup Software Trigger" > "Setup SB Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:SENT:CHKall` on page 1758
`TRIGger:SBSW:SENT:CLR` on page 1758
`TRIGger:SBSW:SENT:INVert` on page 1758
`TRIGger:SBSW:SENT:RST` on page 1758

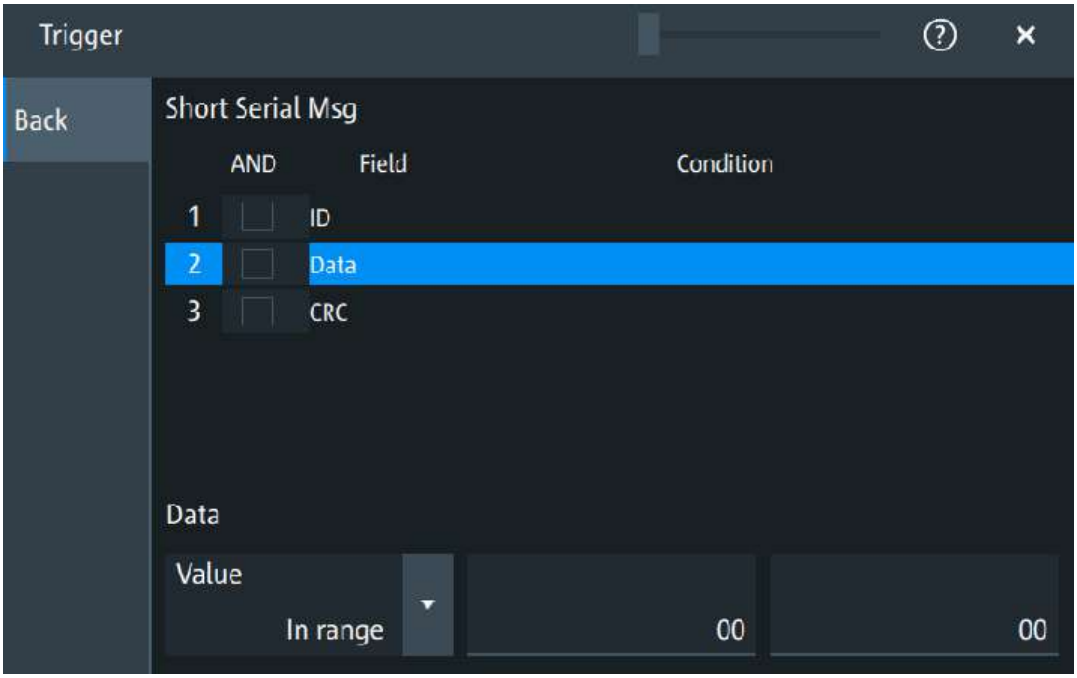
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frame types are available: "Transmission Sequence", "Short Serial Msg", "Enhanced 12bit Serial Msg", "Enhanced 16bit Serial Msg", "Pause pulse", "Unknown".

Remote command:
`TRIGger:SBSW:SENT:FRENable` on page 1759
`TRIGger:SBSW:SENT:FRAMe<fr>:ENABle` on page 1759

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.




"Field" Enables the field type that you want to trigger on for the selected frame.
The following fields are available: "MTP", "Sync", "Status", "Data", "CRC", "ID".

Remote command:
`TRIGger:SBSW:SENT:FIENable` on page 1759
`TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:ENABle` on page 1759

"Condition" Displays the value condition for the selected field.

Remote command:
`TRIGger:SBSW:SENT:BIT` on page 1759
`TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:BIT` on page 1759

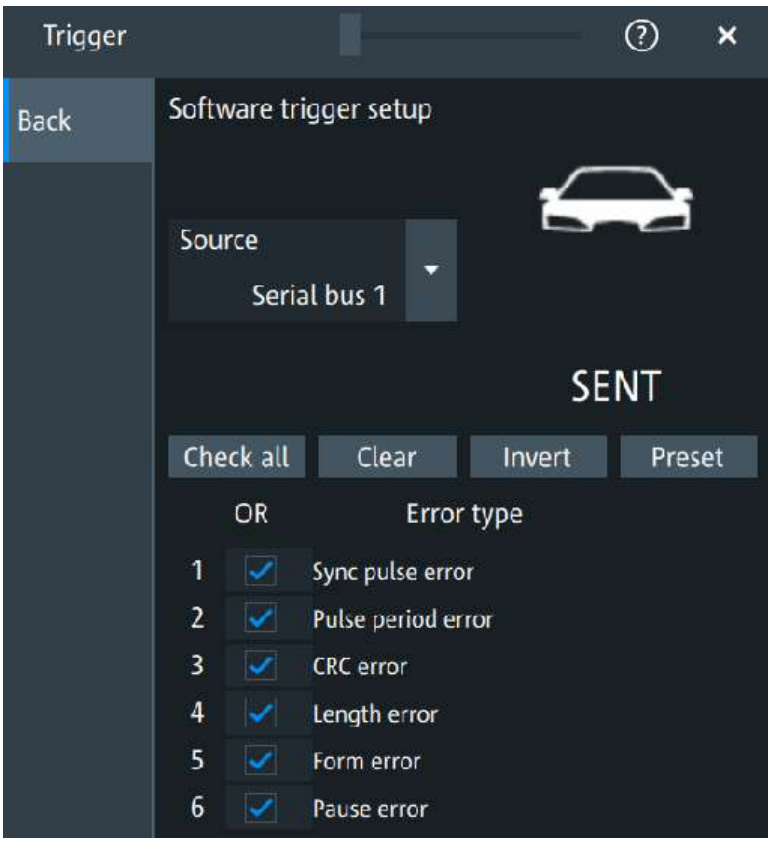
"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`TRIGger:SBSW:SENT:DMAX` on page 1760
`TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DMAX` on page 1760
`TRIGger:SBSW:SENT:DMIN` on page 1760
`TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DMIN` on page 1760
`TRIGger:SBSW:SENT:DOPerator` on page 1760
`TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DOPerator` on page 1760

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:SENT:IMAX on page 1761
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IMAX on page 1761
TRIGger:SBSW:SENT:IMIN on page 1761
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IMIN on page 1761
TRIGger:SBSW:SENT:IOPerator on page 1762
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IOPerator on page 1762

**Error type**
Enables triggering on the selected error type.




The following error types are available: "Sync pulse error", "Pulse period error", "CRC error", "Length error", "Form error", "Pause error".

Remote command:
TRIGger:SBSW:SENT:ERENable on page 1762
TRIGger:SBSW:SENT:ERRor<m>:ENABle on page 1762

### 14.12.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.12.7 SENT decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

#### Decode results table

The "Decode results" table contains information about all decoded frames.




*Figure 14-37: Decoded SENT signal*

The decode results table contains information about all decoded frames.

*Table 14-22: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Index of the decoded frame |
| State | Overall state of the frame |
| Start | Time of frame start |
| Frame type | The type of frame |
| Symbolic name | Symbolic label of the frame. Available only, if "Show symbols" is enabled in the "Display" tab and a symbols list is loaded. |
| Sync time | Synchronisation time |

| Column | Description |
|---|---|
| Status [bin] | Status value |
| Data nibbles | Values of the data nibbles. Select the data format in the "Display" tab. |
| ID value | Identifier value |
| Clock rate | Value of the clock rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

***Table 14-23: Content of the frame details table***

| Column | Description |
|---|---|
| Index | Number of the decoded field |
| Label | Name of the field |
| State | State of the field |
| Value | Value of the field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.13.5, "Decode results", on page 1763.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example SENT export file

```
Index,Start,Stop,State,Type,Symbol,SyncDur,Status,CRC,ID,MTP,SensorRespTime,BR
1 -0.0534 -0.0523 'OK' 'TRSQ' 0.000280 3h Bh - - - 199900
2 -0.0521 -0.0509 'PULS' 'TRSQ' 0.000280 3h Bh - - - 199900
3 -0.050  -0.0499 'OK' 'TRSQ' 0.000280 0h Fh - - - 199900
4 -0.0496 -0.0486 'OK' 'TRSQ' 0.000280 0h 4h - - - 199900
5 -0.0483 -0.0472 'CRC' 'TRSQ' 0.000280 0h 4h - - - 199900
6 -0.0470 -0.0460 'OK' 'TRSQ' 0.000280 0h 1h - - - 199900
7 -0.0457 -0.0447 'OK' 'TRSQ' 0.000280 0h 0h - - - 199900
8 -0.0447 -0.0435 'OK' 'TRSQ' 0.000280 0h Bh - - - 199900
9 -0.0432 -0.0422 'OK' 'TRSQ' 0.000280 0h Eh - - - 199900
10 -0.041 -0.0410 'OK' 'TRSQ' 0.000280 0h 3h - - - 199900

Details frame 1
Index    State    Value
```

```
1    'OK'    14h
2    'OK'    15h
3    'OK'    16h
4    'OK'    17h
5    'OK'    18h
6    'OK'    19h
7    'OK'    1Ah
8    'OK'    1Bh

Details frame 2
Index    State    Value
1    'OK'    1Ch
2    'OK'    1Dh
3    'OK'    1Eh
4    'OK'    1Fh
5    'OK'    20h
6    'OK'    21h
7    'OK'    22h
8    'OK'    23h
9    'OK'    24h
```

## 14.13 ARINC 429 (aerospace electronics, option R&S MXO3-K530)

The ARINC 429 is a specification that defines the characteristics of an avionic data bus used on commercial and transport aircraft.


- About the ARINC 429 ..... 747
- ARINC 429 configuration ..... 748
- ARINC 429 filter ..... 752
- ARINC 429 format configuration ..... 754
- ARINC 429 software trigger ..... 760
- Measure ..... 762
- ARINC 429 decode results ..... 762


### 14.13.1 About the ARINC 429

In an ARINC 429 system, a single transmitter/source is connected to 1 to 20 receivers/sinks on one twisted wire pair. The bus uses differential signals. The ARINC 429 standard uses a simplex communication - data may be transmitted in only one direction. The information is transmitted over the bus in defined series of words.

**Word Format**

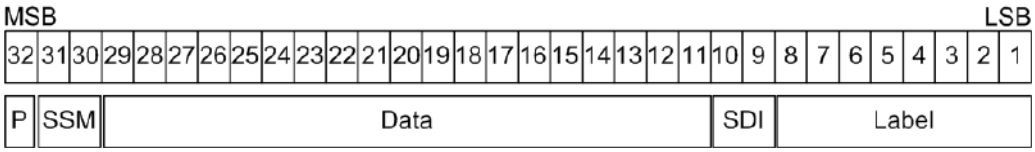



*Figure 14-38: Structure of an ARINC 429 word*

An ARINC 429 word is 32-bits and consists of the following parts (see Figure 14-38):

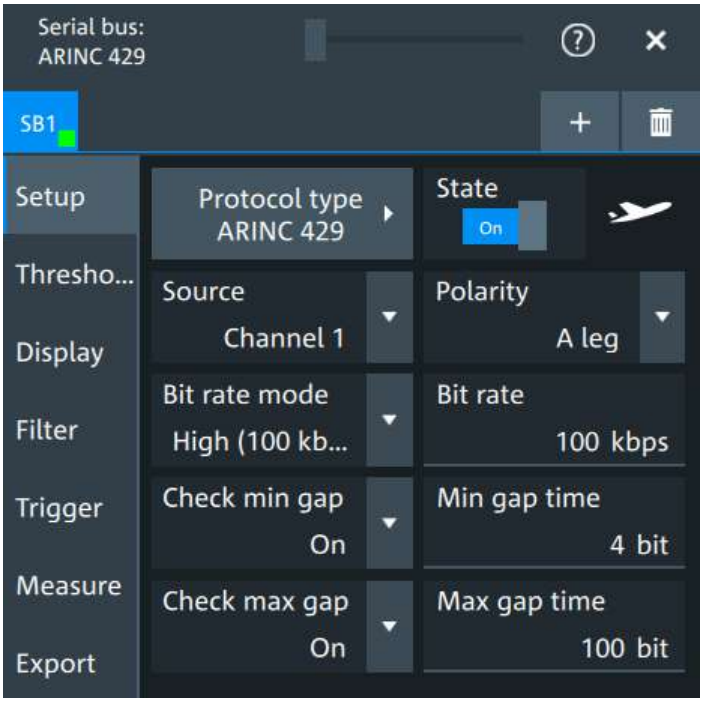
- Parity: the most significant bit (MSB). Checks if there are bit errors during the transmission. The total number of logic 1 bits for the word shall be odd.
- Sign/Status Matrix (SSM): the value of these bits depends on the data type. It may be used to report the status of hardware equipment.
- Data:
  - Binary (BNR): stores the data as a binary number.
  - Binary Coded Decimal (BCD): uses 4 data field bits to represent a decimal digit.
  - Discrete data: a combination of BNR and/ or BCD or individual bits that express specific equipment conditions.
  - Maintenance data and acknowledgment
  - Williamsburg / Buckhorn protocol: a bit-oriented protocol that is used for file transfer.
- Source/Destination Identifier (SDI): indicates the intended receiver or the transmitting subsystem.
- Label: gives information about the word's data type.

For comfortable analysis, you can load an editable label list, to interpret transferred numeric values as meaningful text labels.

## 14.13.2 ARINC 429 configuration

### 14.13.2.1 ARINC 429 configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "ARINC 429" > "Setup".




**Source**
Sets the source of the selected data line. Usually, the source is one of the analog channels.

Remote command:
`SBUS<sb>:ARINc:SOURce` on page 1773

**Polarity**
Selects the wire on which the bus signal is measured : A Leg or B Leg. The setting affects the digitization of the signal.

Remote command:
`SBUS<sb>:ARINc:POLarity` on page 1773

**Bit rate mode, Bit rate**
Selects the number of transmitted bits per second. The value can be set to high speed (100 kbit/s) or low speed (12.0- 14.5 kbit/s).

Remote command:
`SBUS<sb>:ARINc:BRMode` on page 1772
`SBUS<sb>:ARINc:BRValue` on page 1772

**Min gap time,Max gap time**
The gap time defines the idle time between two words, which is needed for word synchronization. The beginning of the first bit after the gap marks the start of a new word.

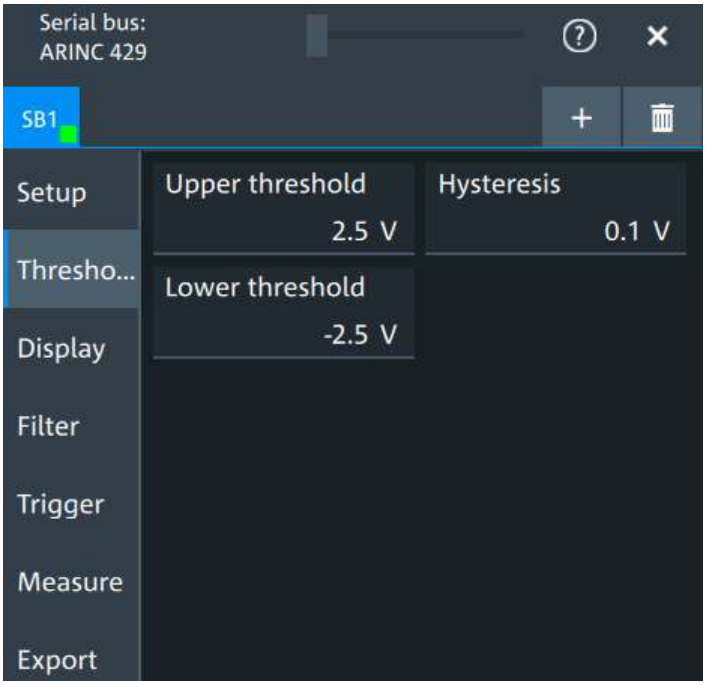
If "Check min gap" and/or "Check max gap" are enabled, the instrument detects the specified gaps during decoding.

You can define a minimum idle time "Min gap time", and/or a maximum time "Max gap time". The standard defines a minimum of 4-bit times to separate two subsequent words.

Remote command:
`SBUS<sb>:ARINc:MAXGap:BITS` on page 1775
`SBUS<sb>:ARINc:MAXGap:SELect` on page 1775
`SBUS<sb>:ARINc:MINGap:BITS` on page 1775
`SBUS<sb>:ARINc:MINGap:SELect` on page 1776

### 14.13.2.2 Thresholds

Access: "Menu" > "Apps" > "Protocol" tab > "ARINC 429" > "Thresholds".




**Threshold**
Sets the upper and lower threshold for the source channel. Enter the value directly in the fields.

Additional to the threshold, you can also set a hysteresis.

Remote command:
`SBUS<sb>:ARINc:THReshold:HIGH` on page 1773
`SBUS<sb>:ARINc:THReshold:LOW` on page 1774
`SBUS<sb>:ARINc:THReshold:HYSTeresis` on page 1774

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
SBUS<sb>:THReshold on page 1364

### 14.13.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

| | |
|---|---|
| "Off" | Disables the display of the decode layer. |
| "Ternary symbols" | Enables the display of the raw, ternary encoded symbols, consisting of the three states "-1", "0" and "+1". |
| "Bits" | Enables the display of all bits. |
| "Words" | Enables the display of all words. |

**Show symbols**
Symbol lists are protocol-specific. Label lists for ARINC 429 are available in CSV format.

For details, see Section 14.13.2.4, "ARINC 429 symbols", on page 751.

Remote command:
SBUS<sb>:ARINc:SYMBols on page 1774
SBUS<sb>:ARINc:NEWList on page 1774

### 14.13.2.4 ARINC 429 symbols

Label lists are protocol-specific. An ARINC 429 label file contains two values for each identifier:

- "Arinc Label": the ARINC 429 label value that identifies the data type and the parameters associated with it. The usual data format is octal.
- "Symbolic label": symbolic name of the label, specifying its function.

**Example: ARINC 429 PTT file**

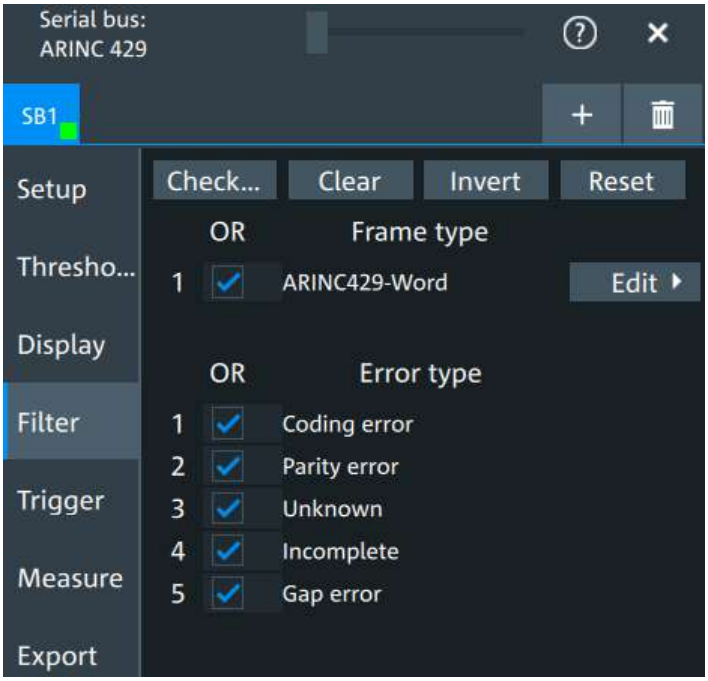
```
# ------------------------------------------------------------------------
@FILE_VERSION = 1.0
@PROTOCOL_NAME = arinc429
# ------------------------------------------------------------------------
# Labels for ARINC 429 protocol
#   Column order: Arinc Label, Symbolic Label
# ------------------------------------------------------------------------
# ----Definition----
001o, Distance to Go
002o, Time to Go
```

```
010o, Present Position - Latitude
011o, Present Position - Longitude
014o, Magnetic Heading
015o, Wind Speed
075o, Gross Weight
125o, Universal Time Coordinated
# ------------------------------------------------------------------------------
s
```

### 14.13.3 ARINC 429 filter

Access: "Menu" > "Apps" > "Protocol" tab > "ARINC 429" > "Filter" tab

In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.




**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
`SBUS<sb>:ARINc:FILTer:RST` on page 1778
`SBUS<sb>:ARINc:FILTer:INVert` on page 1778

SBUS<sb>:ARINc:FILTer:CHKall on page 1777
SBUS<sb>:ARINc:FILTer:CLR on page 1777

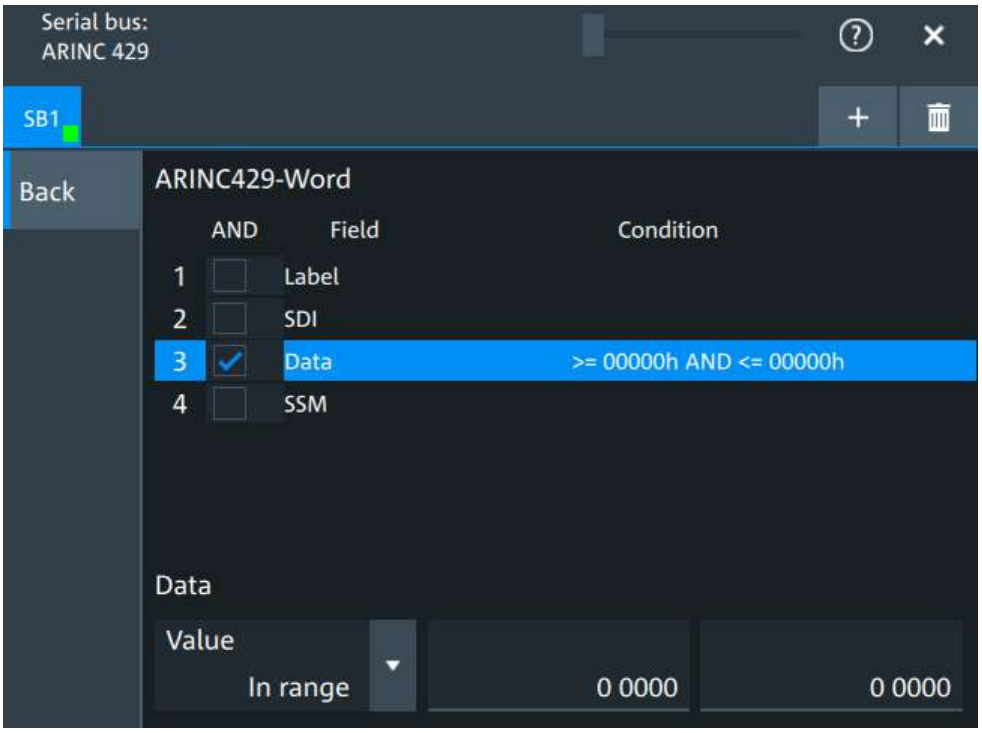
**Enable**
Enables the filtering on ARINC 429 frames. Only the frames that match the selected filter conditions are displayed.

Remote command:
SBUS<sb>:ARINc:FILTer:FRENable on page 1780
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:ENABle on page 1780

**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

**Edit**
Opens a dialog to define the details of the selected frame.




"Field" Enables the field type that you want to filter on for the selected frame. The available fields are "Label", "SDI", "Data", "SSM".

Remote command:
SBUS<sb>:ARINc:FILTer:FIENable on page 1780
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:ENABle on page 1780

"Condition" Displays the value condition for the selected field.

Remote command:
SBUS<sb>:ARINc:FILTer:BIT on page 1778
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:BIT on page 1778

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:ARINc:FILTer:DMAX on page 1779
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1779
SBUS<sb>:ARINc:FILTer:DMIN on page 1779
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1779
SBUS<sb>:ARINc:FILTer:DOPerator on page 1780
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1780

**Error type**
Enables filtering on the selected error type.

The available error types are "Coding error", "Parity error", "Unknown", "Incomplete", "Gap error".

Remote command:
SBUS<sb>:ARINc:FILTer:ERENable on page 1779
SBUS<sb>:ARINc:FILTer:ERRor<n>:ENABle on page 1779

### 14.13.4 ARINC 429 format configuration

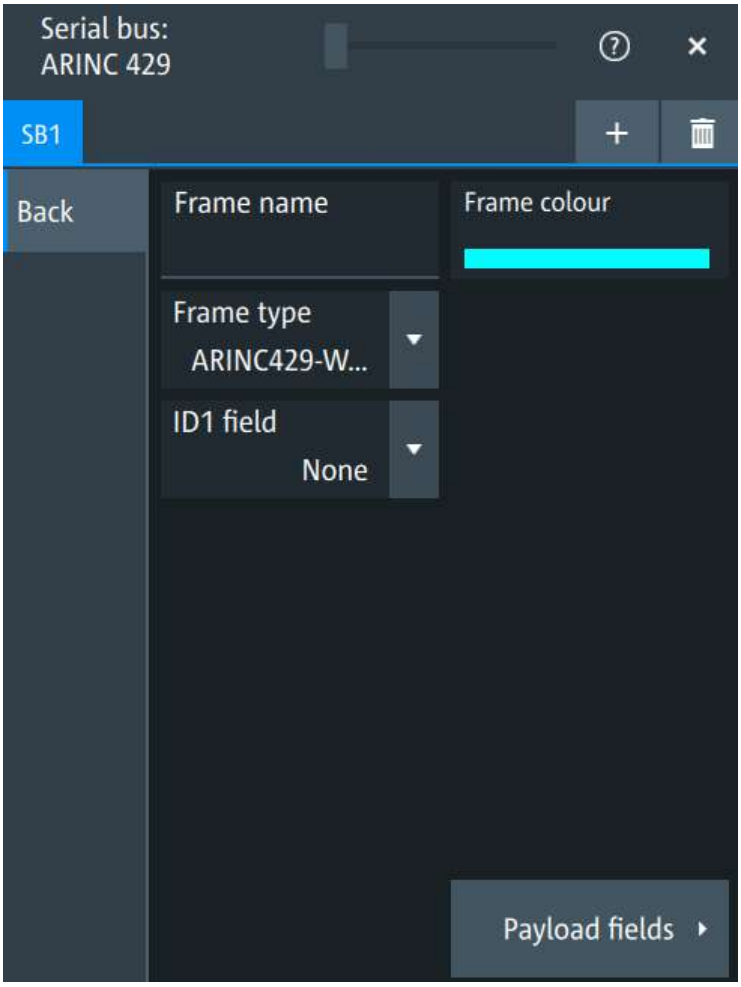
Access: "Menu" > "Apps" > "Protocol" tab > "ARINC 429" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
SBUS<sb>:ARINc:FORMat:SELect on page 1781

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:ARINc:FORMat:ADDFrame on page 1782

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:ARINc:FORMat:CLR on page 1783
SBUS<sb>:ARINc:FORMat:FCOunt? on page 1783

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:ARINc:FORMat:LOAD on page 1782

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:ARINc:FORMat:SAVE on page 1782

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:NAME on page 1784

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:COLor on page 1784

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:TYPE on page 1784

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID1Field on page 1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID2Field on page 1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID3Field on page 1784

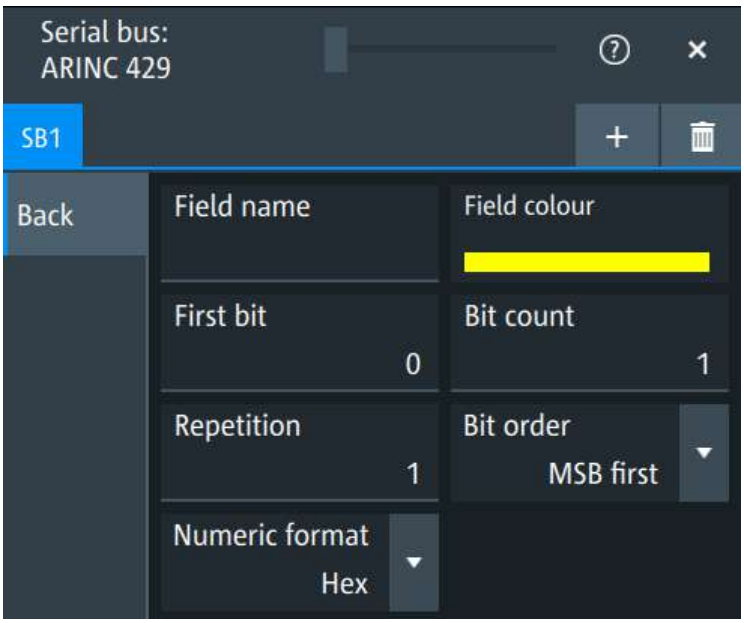
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID1Value on page 1785
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID2Value on page 1785
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID3Value on page 1785

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ADDField on page 1783

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1785

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1785

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1786

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.

- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):

- "Bit order" is "MSB":
  - Bit count = 2: truncated. Result: =00
  - Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  - Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  - Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1786

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
`SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:REPetition` on page 1786

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
`SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:BITorder` on page 1787

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:

- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  - "=1HL111000": valid (read as binary)
  - "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  - "=1010": valid (read as HEX, with a total of 16 bits)
  - "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)

– "=5X12": invalid
– "=1H33": invalid

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1787

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1788

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1788

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1789

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1789

**Parameter A,B**
Defines the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1790
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1790

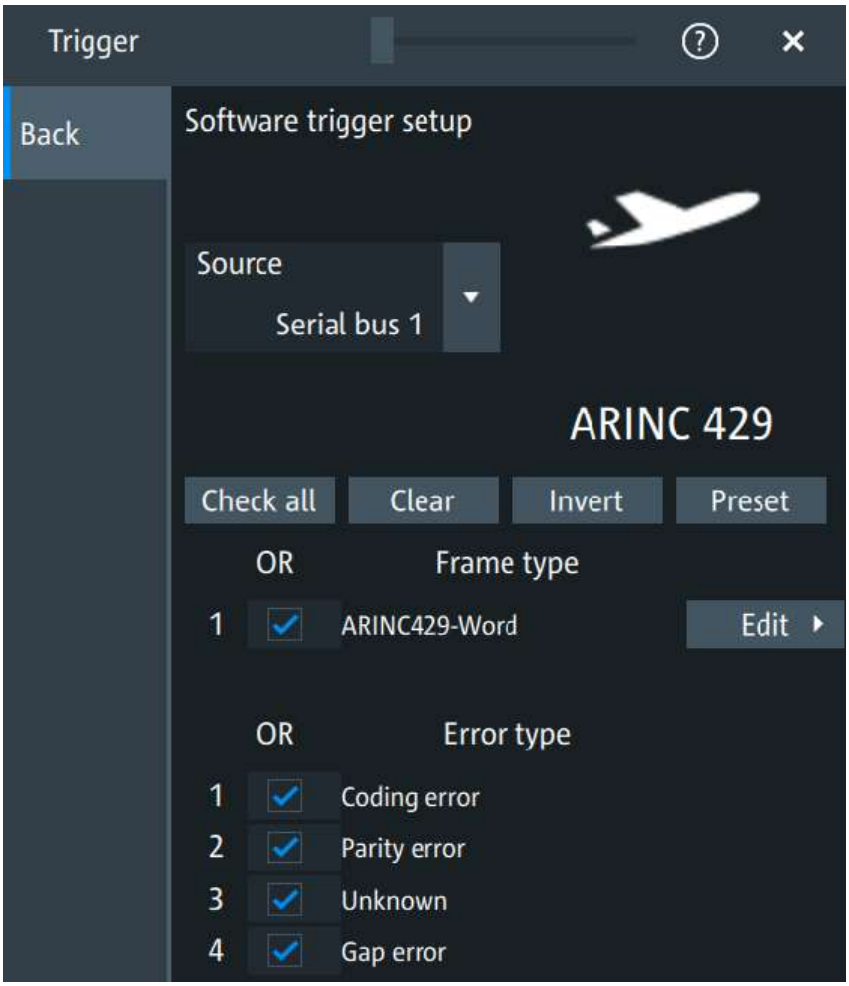
## 14.13.5 ARINC 429 software trigger

### 14.13.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.13.5.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "ARINC 429" > "Trigger" tab > "Setup Software Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:ARINc:CHKall` on page 1791
`TRIGger:SBSW:ARINc:CLR` on page 1791
`TRIGger:SBSW:ARINc:INVert` on page 1791
`TRIGger:SBSW:ARINc:RST` on page 1791

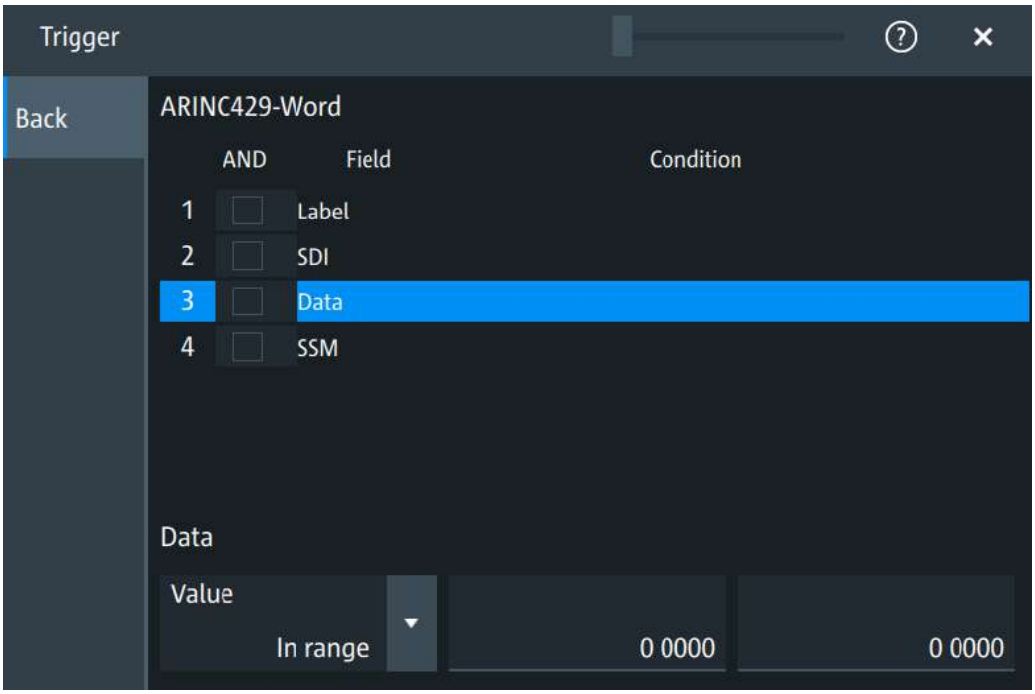
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: "ARINC429-Word".

Remote command:
`TRIGger:SBSW:ARINc:FRENable` on page 1791
`TRIGger:SBSW:ARINc:FRAMe<fr>:ENABle` on page 1791

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.



"Field" Enables the field type that you want to trigger on for the selected frame.
The available fields are "Label", "SDI", "Data", "SSM".

Remote command:
TRIGger:SBSW:ARINc:FIENable on page 1793
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:ENABle on page 1793

"Condition" Displays the value condition for the selected field.

Remote command:
TRIGger:SBSW:ARINc:BIT on page 1792
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:BIT on page 1792

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
TRIGger:SBSW:ARINc:DMAX on page 1792
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DMAX on page 1792
TRIGger:SBSW:ARINc:DMIN on page 1792
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DMIN on page 1792
TRIGger:SBSW:ARINc:DOPerator on page 1793
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DOPerator on page 1793

**Error type**
Enables triggering on the selected error type.

The available error types are "Coding error", "Parity error", "Unknown", "Gap error".

Remote command:
TRIGger:SBSW:ARINc:ERENable on page 1794
TRIGger:SBSW:ARINc:ERRor<m>:ENABle on page 1794

### 14.13.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.13.7 ARINC 429 decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

**Decode results table**

The "Decode results" table contains information about all decoded frames.




*Figure 14-39: Decoded ARINC429 signal*

The decode results table contains information about all decoded frames.

*Table 14-24: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Frame count |
| State | Overall state of the frame |
| Start | Start time of the frame |
| Symbol | Symbolic label of the frame. Available only, if "Show symbols" is enabled in the "Display" tab and a symbols list is loaded. |
| Label | The label name |
| SDI | The state of the SDI bits |
| Data | All 32 bits of the word. Select the data format in the "Display" tab. |
| SSM | The state of the SSM bits |
| 32-bit word | The value of the data bytes. Select the data format in the "Display" tab. |
| Data rate | Value of the data rate |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.14.5, "Decode results", on page 1795.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example ARINC export file

```
Index,Start,Stop,Symbol,State,Label,SDI,Data,SSM,Raw Data,Bit Rate
1,-0.103,-0.10268,---,'OK',257o,3h,72470h,01b,#HC1,#HC4,#H9F,#HF5,100000
2,-0.10268,-0.10236,---,'OK',316o,0h,22608h,00b,#H02,#H0C,#H88,#H73,100000
3,-0.10236,-0.10204,---,'OK',257o,3h,72470h,01b,#HC1,#HC4,#H9F,#HF5,100000
4,-0.10204,-0.10172,---,'OK',317o,0h,62608h,00b,#H02,#H0C,#H8C,#HF3,100000
5,-0.10172,-0.1014,---,'OK',257o,3h,72470h,01b,#HC1,#HC4,#H9F,#HF5,100000
```

## 14.14 MIL-1553 (aerospace electronics, option R&S MXO3-K530)

The MIL-1553 (or MIL-STD-1553) specification defines the characteristics of a serial data bus originally designed for use in the military avionics. Nowadays it is also used in spacecraft on-board data handling.

- About the MIL-1553 protocol....764
- MIL-1553 configuration....767
- MIL-1553 filter....769
- MIL-1553 software trigger....772
- Measure....775
- MIL-1553 decode results....775

### 14.14.1 About the MIL-1553 protocol

The MIL-1553 bus is a 2-wire bus that uses differential signals.

A MIL-1553 system consists of the following components:

- Bus Controller (BC): initiates and coordinates the data flow in the system.
- Remote Terminal (RT): interfaces various subsystems with the data bus. A system consists of up to 31 RTs and each RT can have 31 subaddresses.
  The subaddresses 0 and 31 refer to a mode code command.
- Bus Monitor (BM) (optional): listens to all messages and can record selected data for real-time or off-line analysis.

The information is transmitted over the bus in defined series of words using Manchester code, where each bit is transmitted as high-low for a logical 1 or a low-high for a logical 0. There are three types of words: command, data and status.

### Command Word

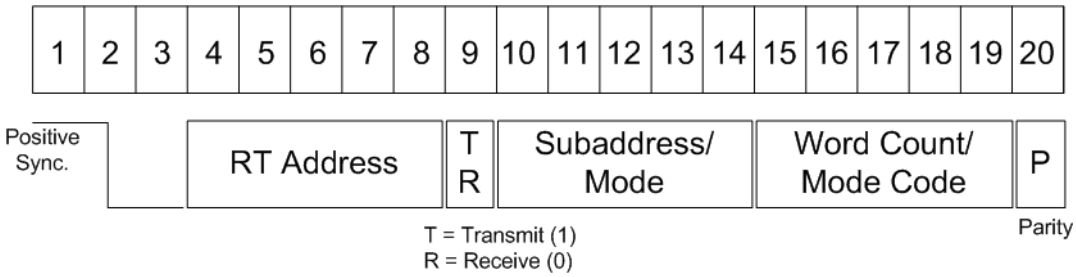



*Figure 14-40: Structure of a command word*

The format of a command word consists of the following parts (see Figure 14-40):

- Sync: an invalid Manchester waveform.
- Remote Terminal (RT) Address: the unique address of the corresponding RT.
- Transmit/Receive (T/R): indicates the action required from the RT.
- Subaddress/Mode Code: indicates the RT subaddress. The subaddresses 0 and 31 signalize the transmission of a mode code.
- Data Word Count /Mode Code: indicates the number of words that are sent/ received by the RT. A maximum of 32 words is allowed. This field may be used for the transmission of the mode code value.
- Parity: checks if there are bit errors during the transmission. The total number of logic 1 bits for the word (sync bits not included) shall be odd.

### Data Word

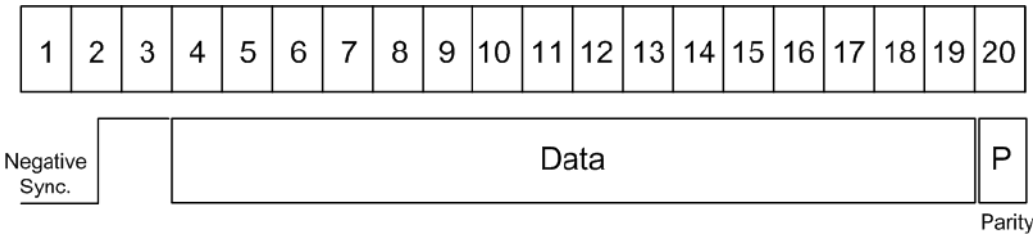



*Figure 14-41: Structure of a data word*

The format of a data word consists of the following parts (see Figure 14-41):

- Sync: an invalid Manchester waveform.
- Data: the transferred information (16 bit).
- Parity: checks if there are bit errors during the transmission. The total number of logic 1 bits for the word (sync bits not included) shall be odd.

**Status Word**

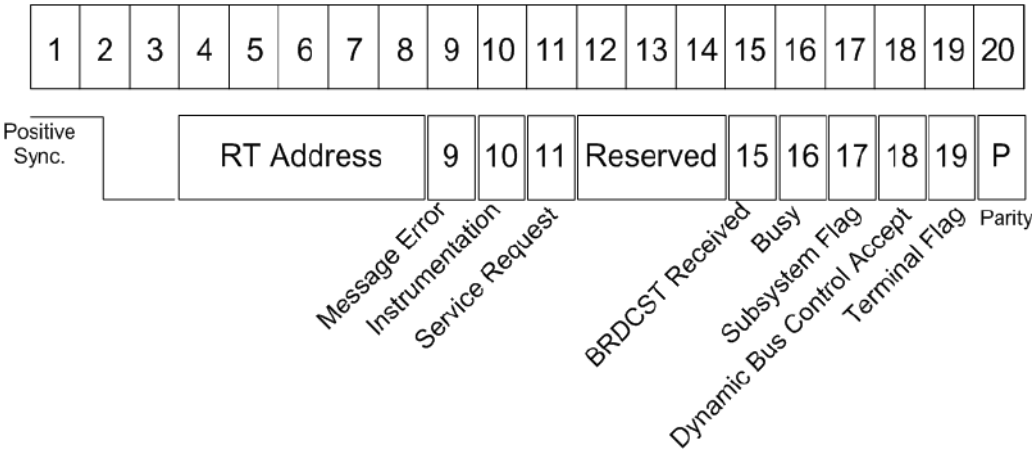



*Figure 14-42: Structure of a status word*

The format of a status word consists of the following parts (see Figure 14-42):

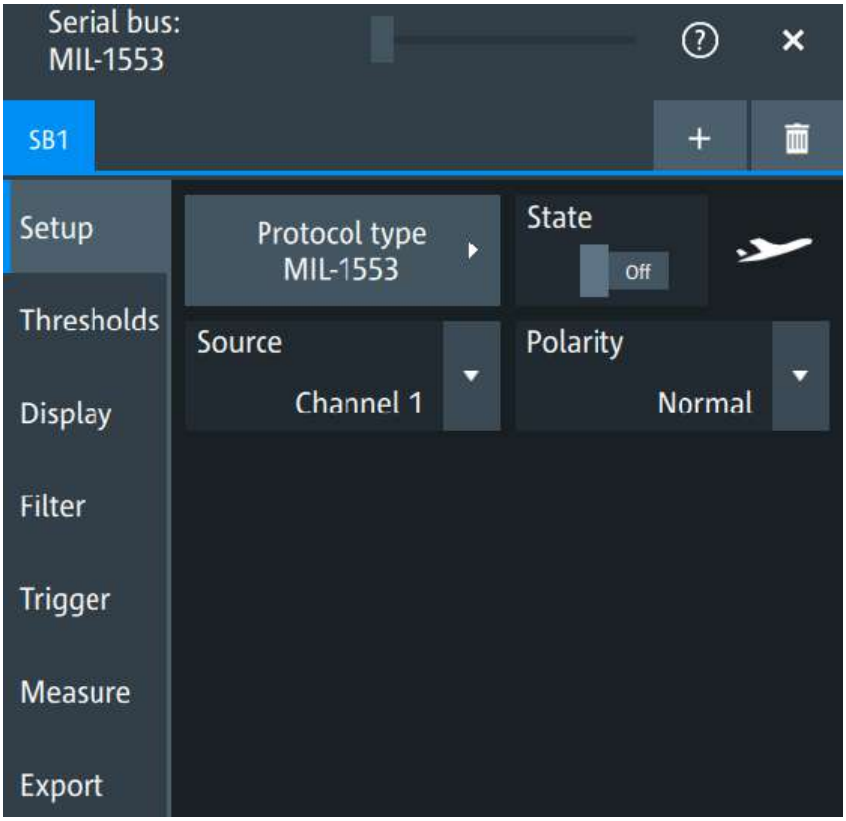
- Sync: an invalid Manchester waveform.
- Remote Terminal (RT) Address: the unique address of the corresponding RT.
- Message error: indicates an error in the command/data word transmission from the BC. A logic 1 indicates presence of a message error and a logic 0 indicates its absence.
- Instrumentation: helps to distinguish between a status word and a command word. The logic state of this bit shall be 0.
- Service Request: indicates that the RT requires service. A logic 1 indicates a presence of a service request and logic 0 indicates its absence.
- Reserved: bits reserved for future uses.
- Broadcast Command: a logic 1 indicates that the preceding valid command word was a broadcast command and a logic 0 that it was not.
- Busy: a busy state indicates that the RT or the subsystem is not able to transfer data. A logic 1 indicates a presence of a busy condition and logic 0 indicates its absence.
- Subsystem Flag: flags a subsystem fault. A logic 1 indicates a presence of a flag and logic 0 indicates its absence.
- Dynamic Bus Control Acceptance: a logic 1 indicates acceptance of a dynamic bus control and a logic 0 a rejection.
- Terminal Flag: flags an RT fault condition. A logic 1 indicates a presence of a flag and logic 0 indicates its absence.
- Parity: checks if there are bit errors during the transmission. The total number of logic 1 bits for the word (sync bits not included) shall be odd.

For comfortable analysis, you can load an editable label list, to interpret transferred numeric values as meaningful text labels.

## 14.14.2 MIL-1553 configuration

### 14.14.2.1 MIL-1553 configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "MIL-1553" > "Setup".




**Source**
Sets the source of the selected data line. Usually, the source is one of the analog channels.

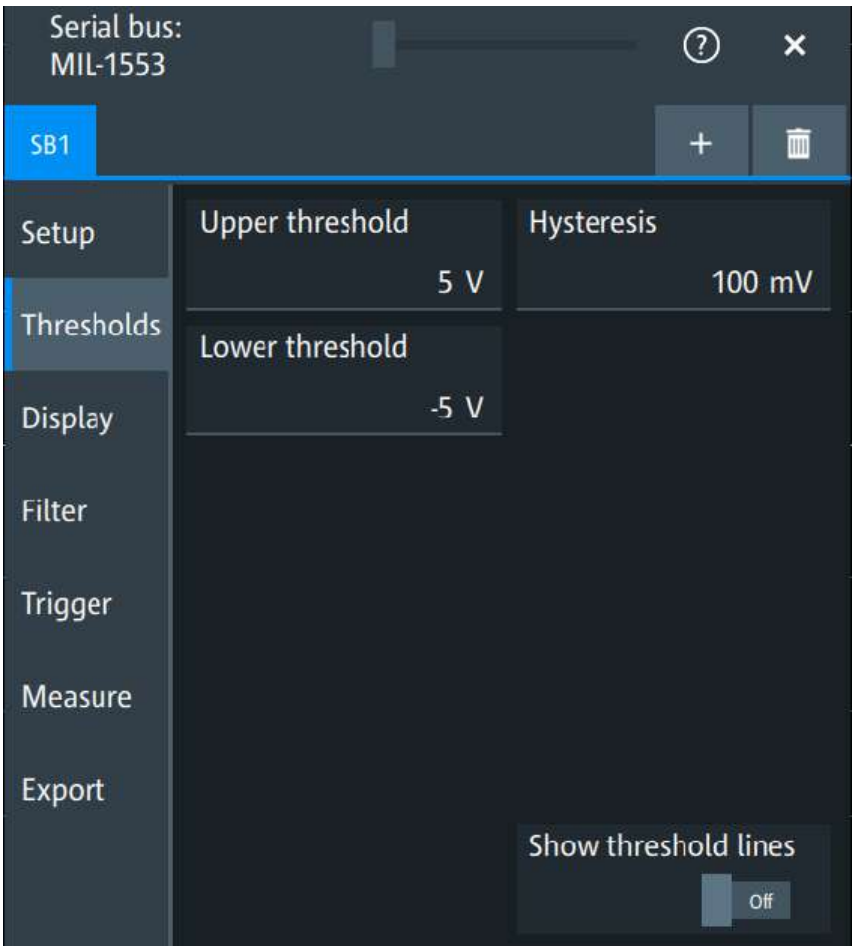
Remote command:
`SBUS<sb>:MILStd:SOURce` on page 1801

**Polarity**
Sets the wire on which the bus signal is measured : normal or inverted. The setting affects the digitization of the signal.

Remote command:
`SBUS<sb>:MILStd:POLarity` on page 1801

### 14.14.2.2 Threshold

Access: "Menu" > "Apps" > "Protocol" tab > "MIL-1553" > "Thresholds".




**Threshold**
Sets the upper and lower threshold for the source channel. Enter the value directly in the fields.

Additional to the threshold, you can also set a hysteresis.

Remote command:
`SBUS<sb>:MILStd:THReshold:HIGH` on page 1801
`SBUS<sb>:MILStd:THReshold:LOW` on page 1802
`SBUS<sb>:MILStd:THReshold:HYSTeresis` on page 1802

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
`SBUS<sb>:THReshold` on page 1364

#### 14.14.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

**Show symbols**
Symbol lists are protocol-specific. Label lists for MIL-1553 are available in CSV format.

For details, see Section 14.14.2.4, "MIL-1553 symbols", on page 769.

#### 14.14.2.4 MIL-1553 symbols

Label lists are protocol-specific. A MIL-1553 label file contains four values for each identifier:

- "RTA": hexadecimal remote terminal address value
- "Sub Addr": hexadecimal sub address value
- "Sub Address Label Name": the label name corresponding to the value of the subaddress.
- "Symbolic label": symbolic name of addressed device, specifying the device function, and the label of the sub address.

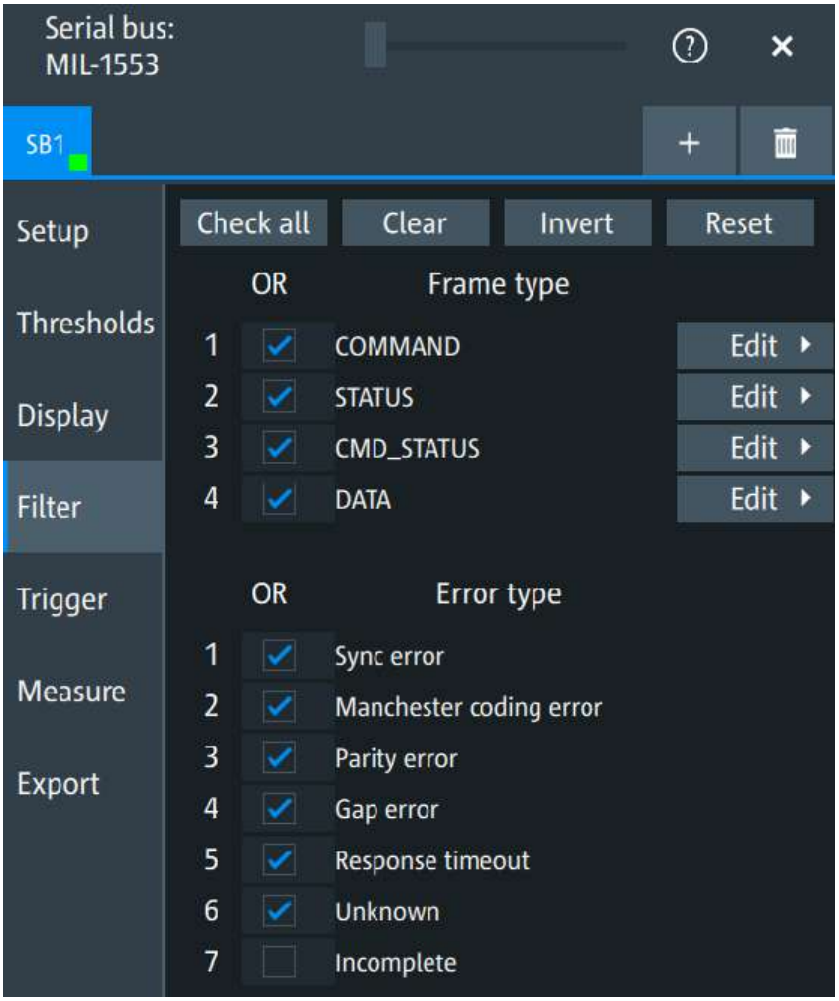
**Example: MIL PTT file**

```
# ------------------------------------------------------------
# Labels for MIL.1553 protocol
# Column order: RT address, RT label, Subaddress, Subaddress Label
# ------------------------------------------------------------------------
@PROTOCOL_NAME = mil1553
0Ah,Engine,01h,Thrust
03h,Main panel,07h,Altimeter
03h,Main panel,01h,Speed
0Eh,Only RTA
```

### 14.14.3 MIL-1553 filter

Access: "Menu" > "Apps" > "Protocol" tab > "MIL-1553" > "Filter" tab

In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.




**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
`SBUS<sb>:MILStd:FILTer:RST` on page 1805
`SBUS<sb>:MILStd:FILTer:INVert` on page 1804
`SBUS<sb>:MILStd:FILTer:CHKall` on page 1804
`SBUS<sb>:MILStd:FILTer:CLR` on page 1804

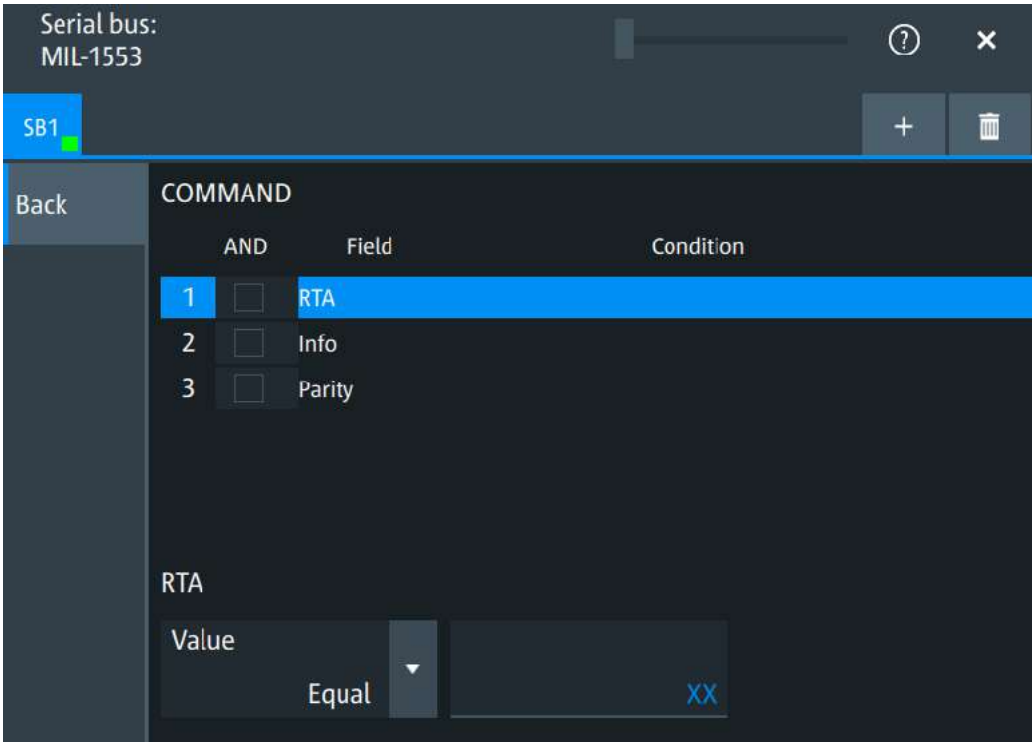
**Enable**
Enables the filtering on MIL-1553 frames. Only the frames that match the selected filter conditions are displayed.

Remote command:
SBUS<sb>:MILStd:FILTer:FRENable on page 1808
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:ENABle on page 1808

**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

**Edit**
Opens a dialog to define the details of the selected frame.




"Field" Enables the field type that you want to filter on for the selected frame. The available fields are "RTA", "Info", "Parity", "Data".

Remote command:
SBUS<sb>:MILStd:FILTer:FIENable on page 1808
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:ENABle on page 1808

"Condition" Displays the value condition for the selected field.

Remote command:
SBUS<sb>:MILStd:FILTer:BIT on page 1805
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:BIT on page 1805

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:MILStd:FILTer:DMAX on page 1806
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1806
SBUS<sb>:MILStd:FILTer:DMIN on page 1806
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1806
SBUS<sb>:MILStd:FILTer:DOPerator on page 1806
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1806

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:MILStd:FILTer:IMAX on page 1807
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1807
SBUS<sb>:MILStd:FILTer:IMIN on page 1807
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1807
SBUS<sb>:MILStd:FILTer:IOPerator on page 1808
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1808

**Error type**
Enables filtering on the selected error type.

The available error types are "Sync error", "Manchester coding error", "Parity error", "Gap error", "Response timeout", "Unknown", "Incomplete".

Remote command:
SBUS<sb>:MILStd:FILTer:ERENable on page 1805
SBUS<sb>:MILStd:FILTer:ERRor<n>:ENABle on page 1805

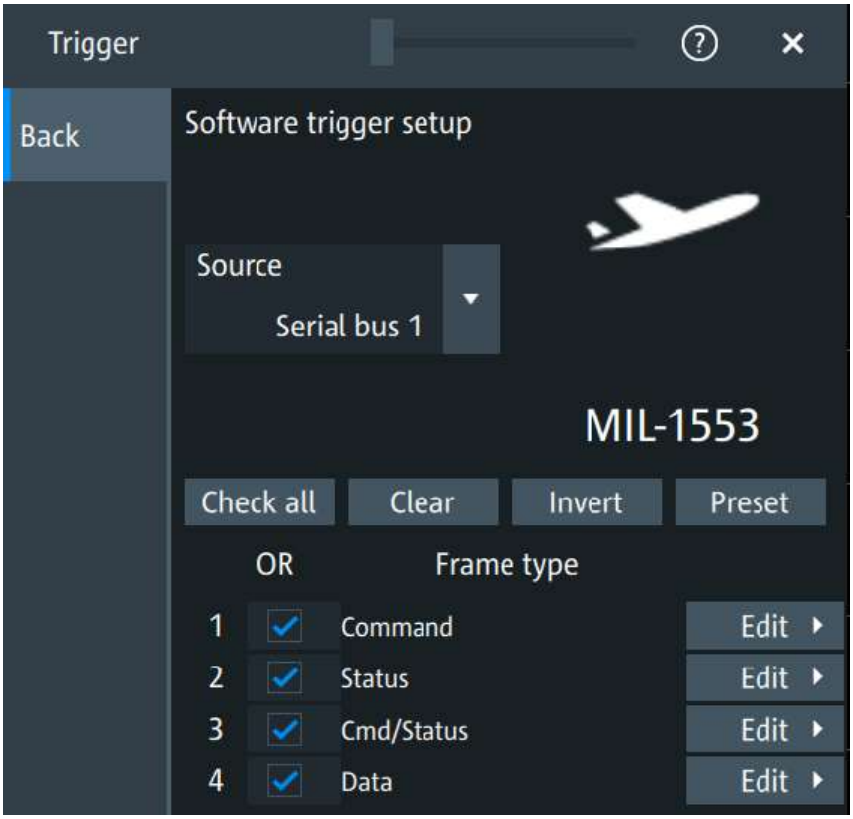
## 14.14.4 MIL-1553 software trigger

### 14.14.4.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.14.4.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "MIL-1553" > "Trigger" tab > "Setup Software Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:MILStd:CHKall` on page 1818
`TRIGger:SBSW:MILStd:CLR` on page 1818
`TRIGger:SBSW:MILStd:INVert` on page 1819
`TRIGger:SBSW:MILStd:RST` on page 1819

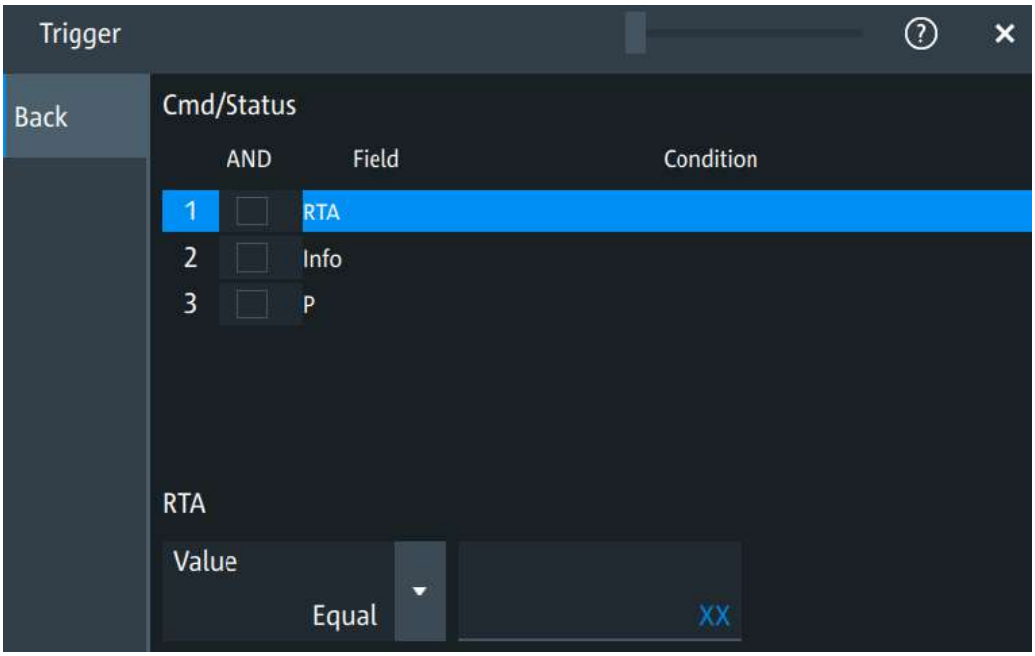
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: "Command", "Status", "Cmd/Status", "Data".

Remote command:
TRIGger:SBSW:MILStd:FRENable on page 1819
TRIGger:SBSW:MILStd:FRAMe<fr>:ENABle on page 1819

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.




"Field" Enables the field type that you want to trigger on for the selected frame.
The following fields are available: "RTA", "Info", "Parity", "Data".

Remote command:
TRIGger:SBSW:MILStd:FIENable on page 1821
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:ENABle on page 1821

"Condition" Displays the value condition for the selected field.

Remote command:
TRIGger:SBSW:MILStd:BIT on page 1819
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:BIT on page 1819

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
TRIGger:SBSW:MILStd:DMAX on page 1820
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DMAX on page 1820
TRIGger:SBSW:MILStd:DMIN on page 1820
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DMIN on page 1820
TRIGger:SBSW:MILStd:DOPerator on page 1820
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DOPerator on page 1820

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:MILStd:IMAX on page 1821
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IMAX on page 1821
TRIGger:SBSW:MILStd:IMIN on page 1822
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IMIN on page 1822
TRIGger:SBSW:MILStd:IOPerator on page 1822
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IOPerator on page 1822

**Error type**
Enables triggering on the selected error type.

The following error types are available: "Sync error", "Manchester coding error", "Parity error", "Gap error", "Response timeout".

Remote command:
TRIGger:SBSW:MILStd:ERENable on page 1822
TRIGger:SBSW:MILStd:ERRor<m>:ENABle on page 1822

### 14.14.5 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.14.6 MIL-1553 decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

### Decode results table

The "Decode results" table contains information about all decoded frames.

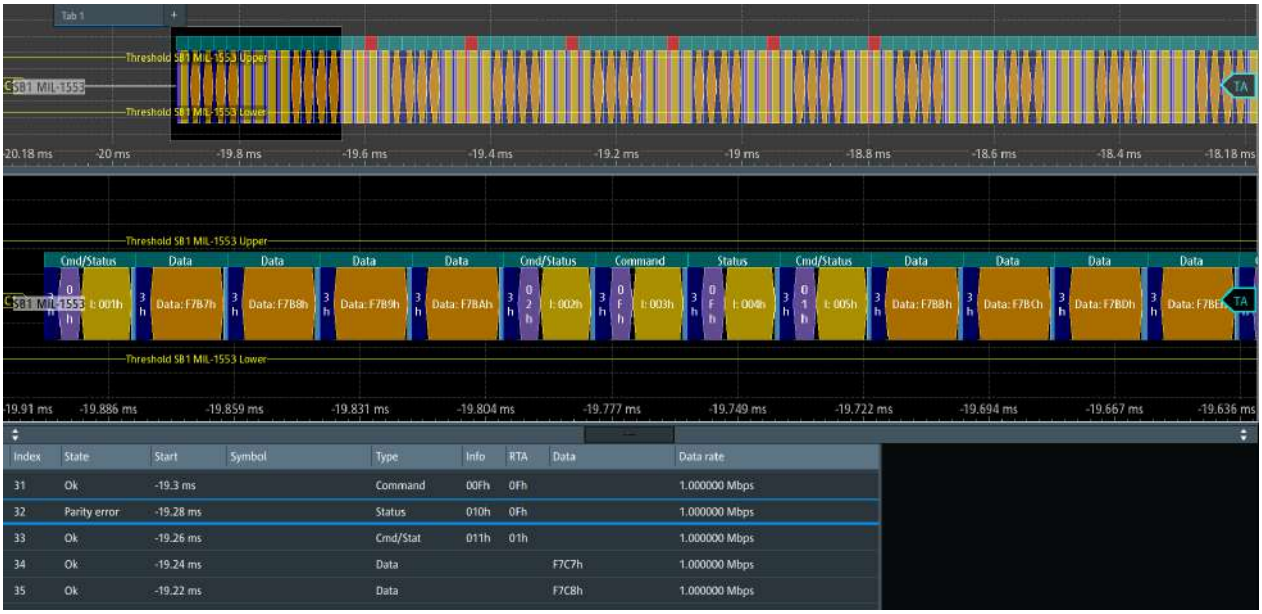



*Figure 14-43: Decoded MIL-1553 signal*

The decode results table contains information about all decoded frames.

*Table 14-25: Content of the Decode results table*

| Column | Description |
|---|---|
| Index | Index of the decoded word |
| State | Overall state of the word |
| Start | Time of word start in relation to the trigger point |
| Symbol | Symbolic label of the frame. Available only, if "Show symbols" is enabled in the "Display" tab and a symbols list is loaded. |
| Type | Word type |
| Info | The hexadecimal value of the 9th to 1th bit of a command/status word |
| RTA | Remote terminal address |
| Data | The values of the data bytes. Select the data format in the "Display" tab. |
| Data rate | Data rate |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

**Remote commands**

Remote commands to retrieve decode results are described in Section 18.18.15.5, "Decode results", on page 1823.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

**Example MIL-1553 export file**

```
Index,Start,Stop,Type,State,RTA,Info,Data,Bit Rate
1,-0.0199,-0.01988,'STAT','OK',1Dh,027h,---,1000000
2,-0.01988,-0.01986,'DATA','OK',---,---,F7C0h,1000000
3,-0.01986,-0.01984,'DATA','OK',---,---,F7CAh,1000000
4,-0.01984,-0.01982,'DATA','OK',---,---,F7D4h,1000000
5,-0.01982,-0.0198,'DATA','OK',---,---,F7DEh,1000000
```

# 14.15 SpaceWire (aerospace communications, option R&S MXO3-K530)

The SpaceWire is a communication network standard used for spacecraft. It is based on the IEEE 1355 standard of communications and coordinated by the European Space Agency (ESA).


- SpaceWire basics ..... 777
- SpaceWire configuration ..... 778
- SpaceWire filter ..... 781
- SpaceWire format configuration ..... 784
- SpaceWire software trigger ..... 790
- Measure ..... 792
- SpaceWire decode results ..... 792


## 14.15.1 SpaceWire basics

The SpaceWire links are a Point-to-point (P2P) connection between a node and another node or a router. The link is full-duplex bidirectional serial data link.

The SpaceWire has two types of characters:

- Data characters containing a parity bit, a data control flag and eight bits of data.

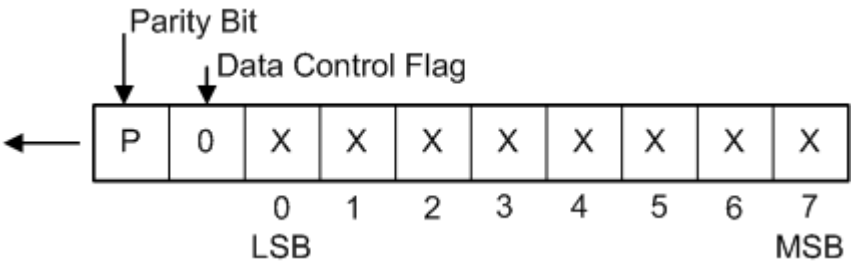



*Figure 14-44: SpaceWire data characters*

- Control characters containing a parity-bit, a data-control flag and the 2-bit control code. The data control flag is set to 1 and indicates that this is a control character.

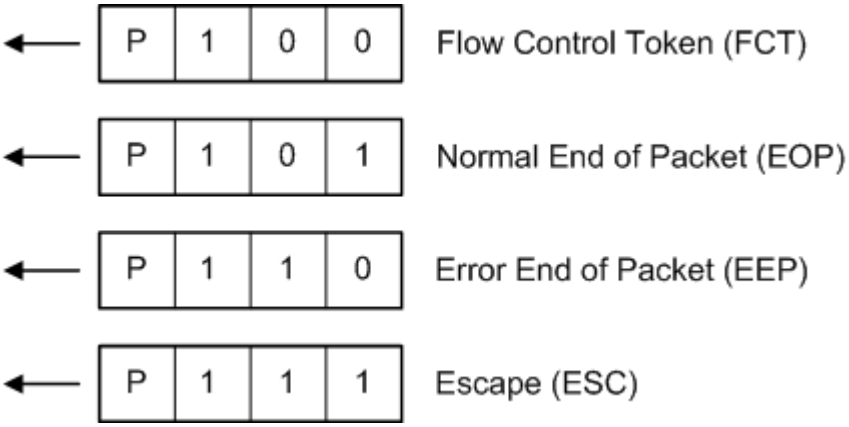



*Figure 14-45: SpaceWire control characters*

Also there are two control codes:

- NULL code consisting of an Escape (ESC) and a Flow Control Token (FCT)
- Time Code consisting of an ESC followed by a single data character

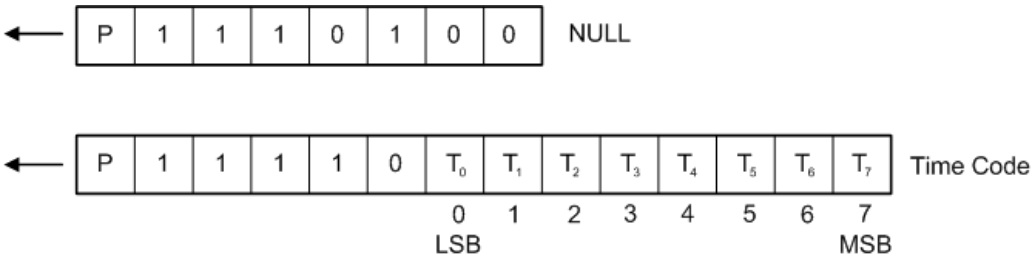
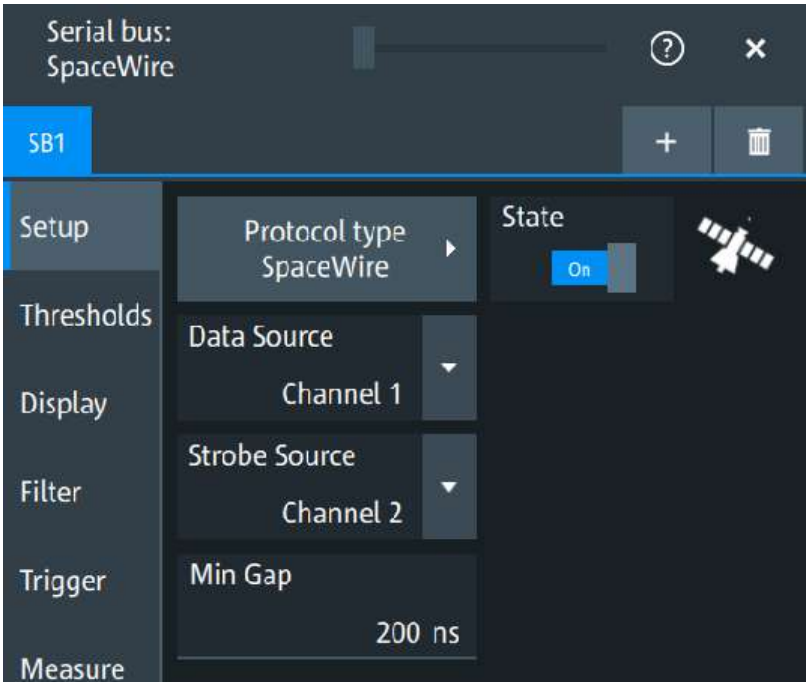



*Figure 14-46: SpaceWire control codes*

## 14.15.2 SpaceWire configuration

### 14.15.2.1 SpaceWire configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "SpaceWire" > "Setup".




**Data Source**
Selects the source for the data signal. Typically, the data source is one of the analog channels.

Remote command:
`SBUS<sb>:SWIRe:DATA:SOURce` on page 1829

**Strobe Source**
Selects the source for the strobe signal. Typically, the strobe source is an analog channel different from the data source.

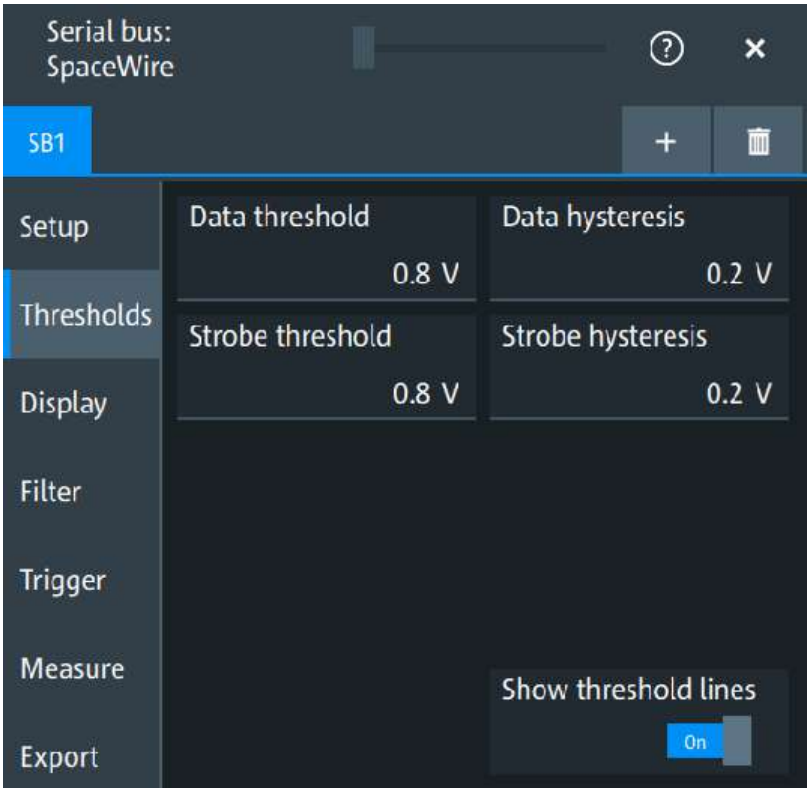
Remote command:
`SBUS<sb>:SWIRe:STRBe:SOURce` on page 1830

**Min Gap**
Specifies the minimum idle time between two frames, which is needed for synchronization. The beginning of the first bit after the gap marks the start of a new frame.

Remote command:
`SBUS<sb>:SWIRe:MGAP` on page 1831
`SBUS<sb>:SWIRe:MINGap` on page 1831

### 14.15.2.2 Thresholds

Access: "Menu" > "Apps" > "Protocol" tab > "SpaceWire" > "Thresholds".




**Thresholds**
Sets the middle threshold for the data source and strobe source channels. Enter the values directly in the fields.

Additional to the thresholds, you can also set the hysteresis for the data source and strobe source, respectively.

Remote command:
`SBUS<sb>:SWIRe:DATA:THReshold` on page 1830
`SBUS<sb>:SWIRe:DATA:HYSTeresis` on page 1830
`SBUS<sb>:SWIRe:STRBe:THReshold` on page 1830
`SBUS<sb>:SWIRe:STRBe:HYSTeresis` on page 1831

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
`SBUS<sb>:THReshold` on page 1364

### 14.15.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

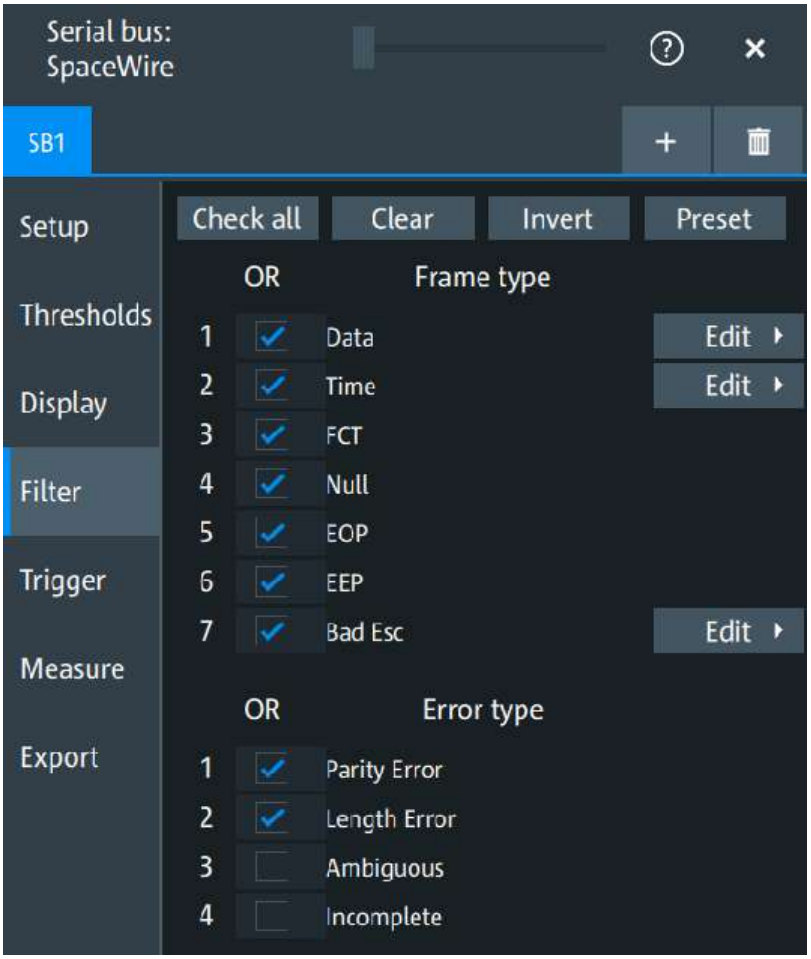
| | |
|---|---|
| "Off" | Disables the display of the decode layer. |
| "Edges" | Enables the display of all edges. |
| "Binary" | Enables the display of the raw, binary encoded symbols, consisting of the states "0" and "1". |

**Data format**
See "Data format" on page 493.

### 14.15.3 SpaceWire filter

Access: "Menu" > "Apps" > "Protocol" tab > "SpaceWire" > "Filter" tab

In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.



**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
`SBUS<sb>:SWIRe:FILTer:CHKall` on page 1832
`SBUS<sb>:SWIRe:FILTer:CLR` on page 1833
`SBUS<sb>:SWIRe:FILTer:INVert` on page 1833
`SBUS<sb>:SWIRe:FILTer:RST` on page 1833

**Enable**
Enables the filtering on SpaceWire frames. Only the frames that match the selected filter conditions are displayed.

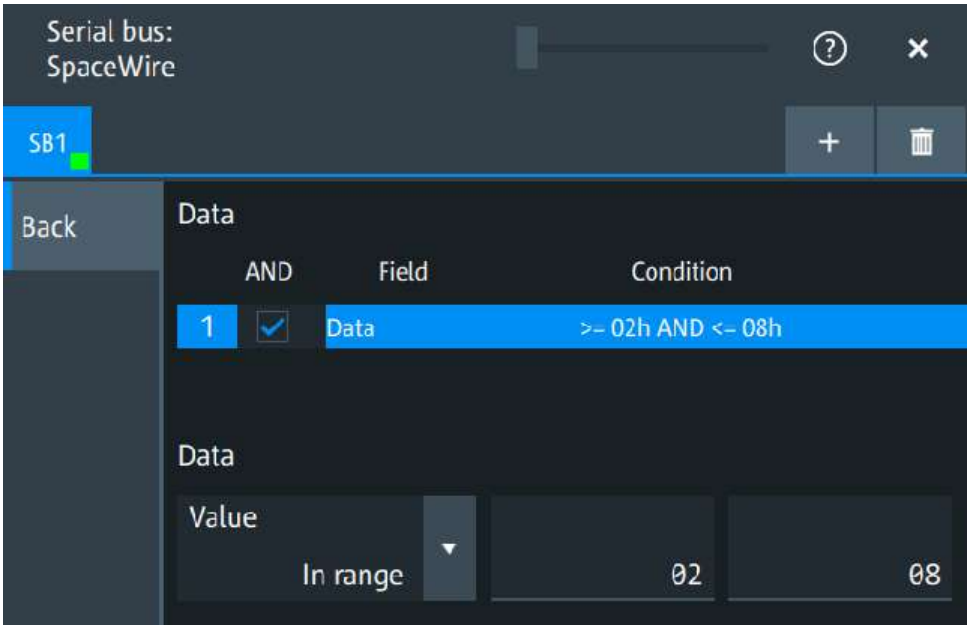
Remote command:
`SBUS<sb>:SWIRe:FILTer:FRENable` on page 1836
`SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:ENABle` on page 1836

**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

The available frame types are "Data", "Time", "FCT", "Null", "EOP", "EEP", "Bad Esc".

**Edit**
Opens a dialog to define the details of the selected frame.




"Field" Enables the field type that you want to filter on for the selected frame. The available fields are "Data", "Time", "Bad Esc".

Remote command:
SBUS<sb>:SWIRe:FILTer:FIENable on page 1835
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:ENABle on page 1835

"Condition" Displays the value condition for the selected field.

Remote command:
SBUS<sb>:SWIRe:FILTer:BIT on page 1833
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:BIT on page 1833

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:SWIRe:FILTer:DMAX on page 1834
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1834
SBUS<sb>:SWIRe:FILTer:DMIN on page 1834
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1834
SBUS<sb>:SWIRe:FILTer:DOPerator on page 1835
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1835

**Error type**
Enables filtering on the selected error type.

The available error types are "Parity error", "Length error", "Ambiguous", "Incomplete".

Remote command:
SBUS<sb>:SWIRe:FILTer:ERENable on page 1834
SBUS<sb>:SWIRe:FILTer:ERRor<n>:ENABle on page 1834

### 14.15.4 SpaceWire format configuration

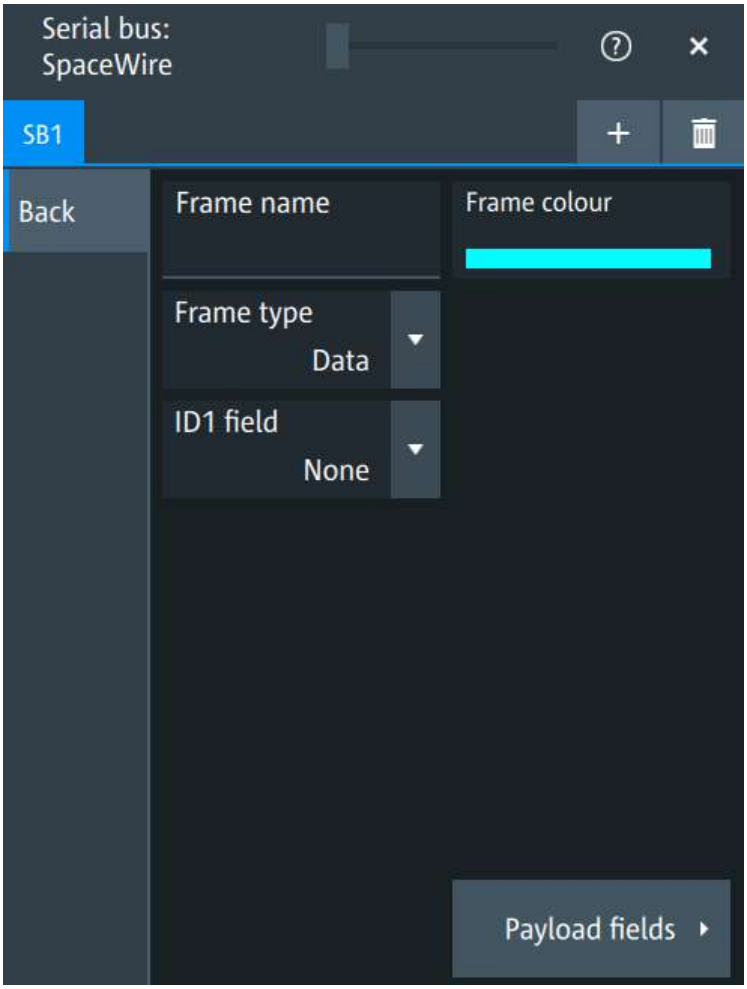
Access: "Menu" > "Apps" > "Protocol" tab > "SpaceWire" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
SBUS<sb>:SWIRe:FORMat:SELect on page 1837

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:SWIRe:FORMat:ADDFrame on page 1837

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:SWIRe:FORMat:CLR on page 1838
SBUS<sb>:SWIRe:FORMat:FCOunt? on page 1838

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:SWIRe:FORMat:LOAD on page 1837

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:SWIRe:FORMat:SAVE on page 1837

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:NAME on page 1839

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:COLor on page 1839

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:TYPE on page 1839

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID1Field on page 1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID2Value on page 1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID3Value on page 1840

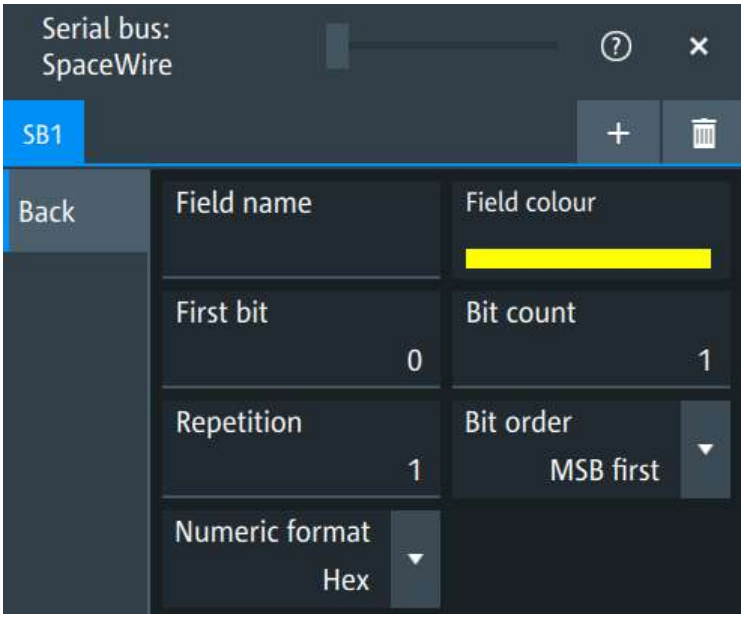
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID1Value on page 1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID2Value on page 1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID3Value on page 1840

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ADDField on page 1838

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1840

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1840

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1841

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.

- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  – Bit count = 2: truncated. Result: =00
  – Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  – Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  – Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1841

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
`SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:REPetition` on page 1842

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
`SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:BITorder` on page 1842

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:
- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  – "=1HL111000": valid (read as binary)
  – "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  – "=1010": valid (read as HEX, with a total of 16 bits)
  – "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)

– "=5X12": invalid
– "=1H33": invalid

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1842

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1843

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1844

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1844

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1844

**Parameter A,B**
Defines the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1845
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1845

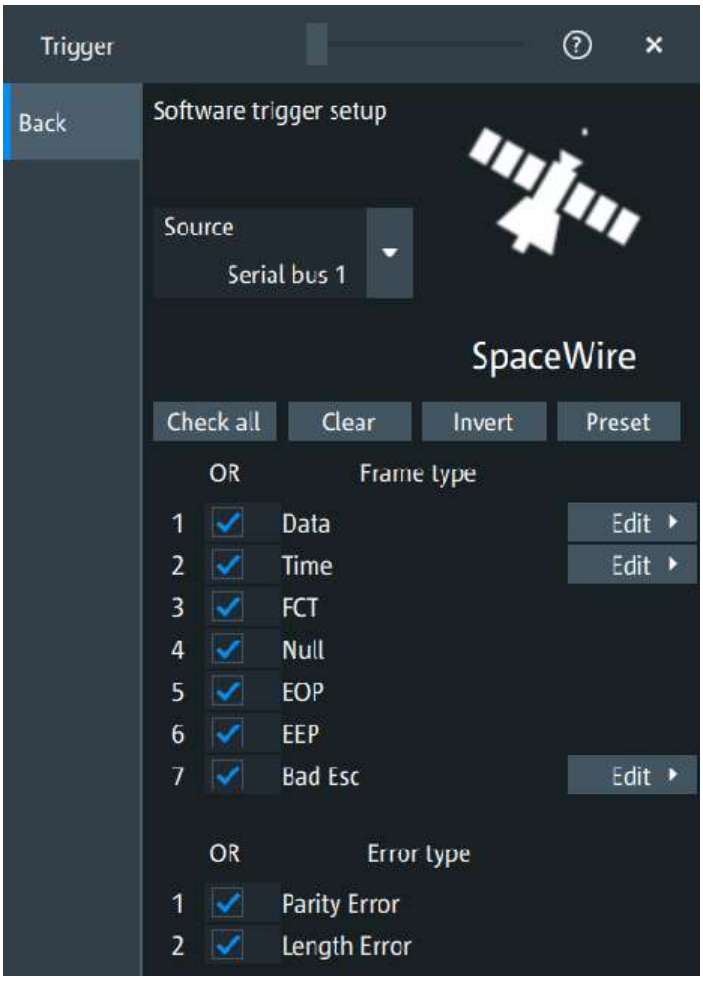
## 14.15.5 SpaceWire software trigger

### 14.15.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.15.5.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "SpaceWire" > "Trigger" tab > "Setup Software Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:SWIRe:CHKall` on page 1846
`TRIGger:SBSW:SWIRe:CLR` on page 1846
`TRIGger:SBSW:SWIRe:INVert` on page 1846
`TRIGger:SBSW:SWIRe:RST` on page 1846

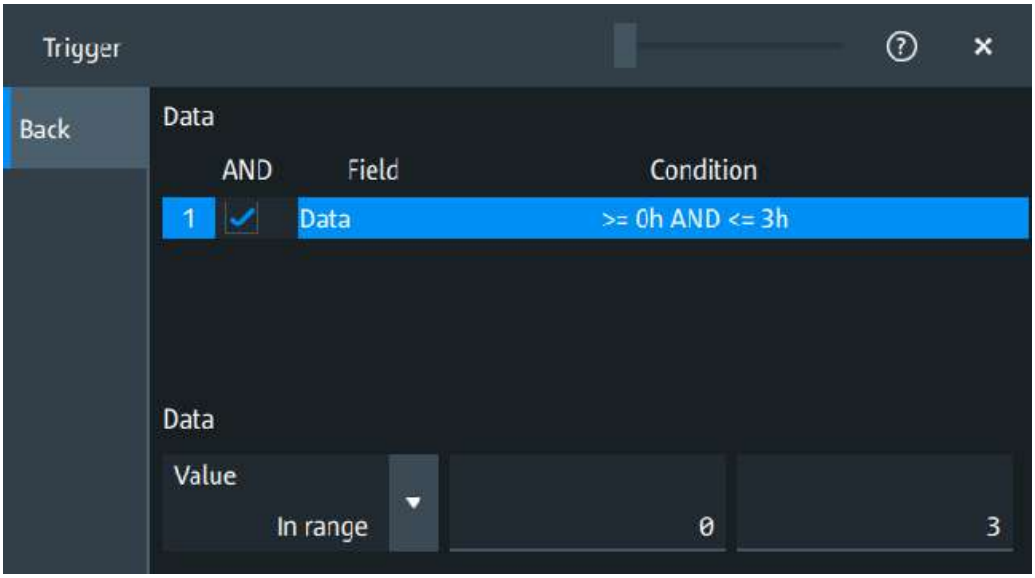
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: "Data", "Time", "FCT", "Null", "EOP", "EEP", "Bad Esc".

Remote command:
`TRIGger:SBSW:SWIRe:FRENable` on page 1847
`TRIGger:SBSW:SWIRe:FRAMe<fr>:ENABle` on page 1847

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.



"Field" Enables the field type that you want to trigger on for the selected frame.
The available fields are "Data", "Time", "Code".

Remote command:
`TRIGger:SBSW:SWIRe:FIENable` on page 1848
`TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:ENABle` on page 1848

"Condition" Displays the value condition for the selected field.

Remote command:
`TRIGger:SBSW:SWIRe:BIT` on page 1847
`TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:BIT` on page 1847

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`TRIGger:SBSW:SWIRe:DMAX` on page 1847
`TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DMAX` on page 1847
`TRIGger:SBSW:SWIRe:DMIN` on page 1848
`TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DMIN` on page 1848
`TRIGger:SBSW:SWIRe:DOPerator` on page 1848
`TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DOPerator` on page 1848

**Error type**
Enables triggering on the selected error type.

The available error types are "Parity error", "Length error".

Remote command:
`TRIGger:SBSW:SWIRe:ERENable` on page 1849
`TRIGger:SBSW:SWIRe:ERRor<m>:ENABle` on page 1849

### 14.15.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.15.7 SpaceWire decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

**Decode results table**

The "Decode results" table contains information about all decoded frames.

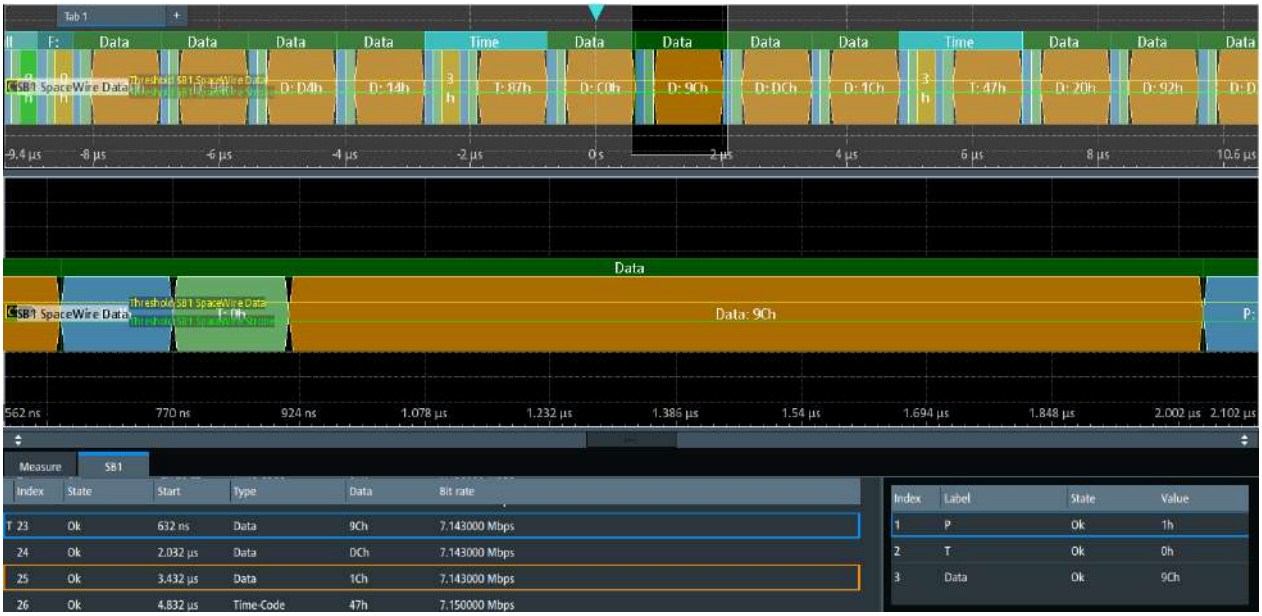



*Figure 14-47: Decoded SpaceWire signal*

The decode results table contains information about all decoded frames.

*Table 14-26: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Frame count |
| State | Overall state of the frame |
| Start | Start time of the frame |
| Type | The type of the frame |
| Data | Value of the data byte. Select the data format in the "Display" tab. |
| Bit rate | Value of the bit rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

*Table 14-27: Content of the frame details table*

| Column | Description |
|---|---|
| Index | Number of the decoded field |
| Label | Name of the field |
| State | State of the field (OK / Not OK) |
| Value | Value of the field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- SBUS<sb>:FORMat on page 1363

**Remote commands**

Remote commands to retrieve decode results are described in Section 18.18.16.5, "Decode results", on page 1850.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

**Example SpaceWire export file**

```
Index,Start,Stop,State,Type
1,-2.7928e-05,-2.6808e-05,'OK','NULL'
2,-2.6808e-05,-2.5688e-05,'OK','NULL'
3,-2.5688e-05,-2.5128e-05,'OK','FCT'
4,-2.5128e-05,-2.3728e-05,'OK','DATA'
5,-2.3728e-05,-2.2328e-05,'OK','DATA'
6,-2.2328e-05,-2.0928e-05,'OK','DATA'
7,-2.0928e-05,-1.897e-05,'OK','TCOD'
8,-1.897e-05,-1.7568e-05,'OK','DATA'
9,-1.7568e-05,-1.617e-05,'OK','DATA'
10,-1.617e-05,-1.477e-05,'OK','DATA'
11,-1.477e-05,-1.4208e-05,'OK','EEP'
```

# 14.16 SPMI (mobile electronics, option R&S MXO3-K550)

The System Power Management Interface (SPMI) is a high-speed, low-latency, bi-directional, 2-wire serial bus.

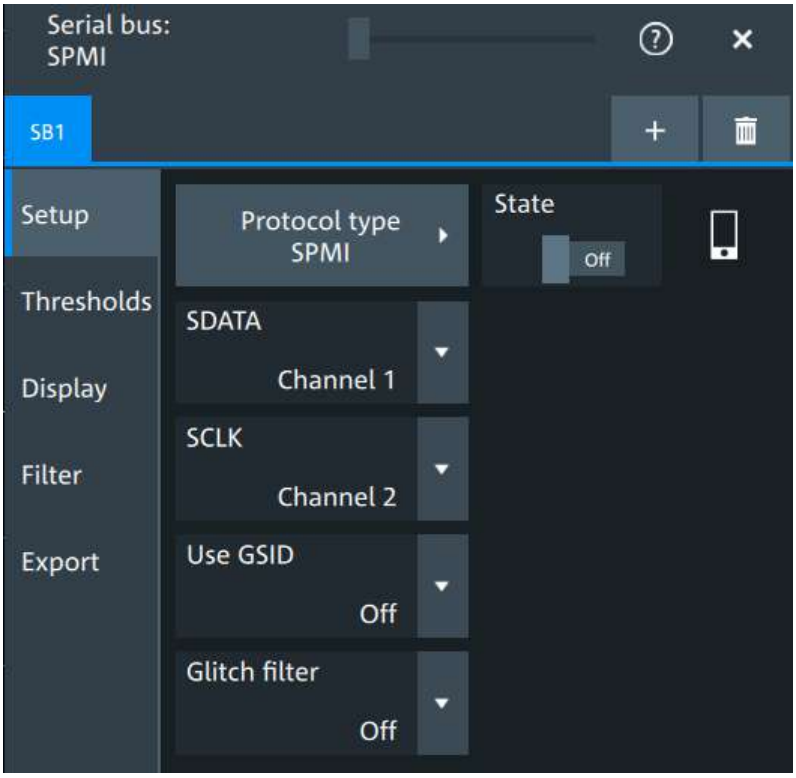
For basic information on SPMI, you can also refer to a video, available on the Rohde & Schwarz YouTube channel: Understanding SPMI.

- SPMI configuration....794
- SPMI filter....797
- SPMI format configuration....800
- SPMI software trigger....806
- Measure....810
- SPMI decode results....810

## 14.16.1 SPMI configuration

### 14.16.1.1 SPMI configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "SPMI" > "Setup".




Make sure that the tab of the correct serial bus is selected.

**Protocol type**
Selects the protocol type to be decoded.

Remote command:
`SBUS<sb>:TYPE` on page 1362

**SDATA**
Sets the source of the data line.

Remote command:
`SBUS<sb>:SPMI:SDATa:SOURce` on page 1859

**SCLK**
Sets the source of the clock line.

Remote command:
`SBUS<sb>:SPMI:SCLK:SOURce` on page 1858

**Use GSID, GSID (hex)**
Enables the use of the group sub ID (GSID). You can then enter a value for the "GSID (hex)". If the "SA" value of a frame matches this "GSID (hex)" value, the interpreter reverses the meaning of the "ACK" field: It interprets a bit value of zero in the ACK field as an error, and a bit value of one as OK.

Remote command:
SBUS<sb>:SPMI:GSIDenable on page 1857
SBUS<sb>:SPMI:GIDValue on page 1857

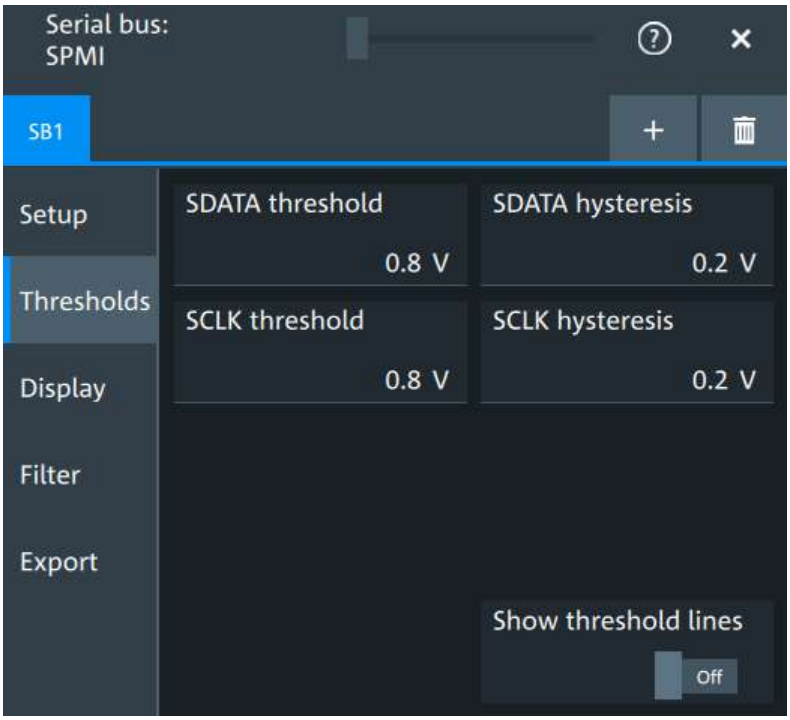
**Glitch filter, Glitch width**
A glitch filter can help to filter out short-duration voltage spikes/ glitches that can occur on the communication line.

If "Glitch filter" > "On" you can set the "Glitch width". Any signal transitions with a duration smaller than this value are considered a glitch and filtered out.

Remote command:
SBUS<sb>:SPMI:GTCHenable on page 1858
SBUS<sb>:SPMI:GTWDith on page 1858

### 14.16.1.2 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "SPMI" > "Thresholds".




**Threshold**
Sets the threshold for the data/clock channel. Enter the value directly in the fields.

Additional to the threshold, you can also set a hysteresis.

Remote command:
SBUS<sb>:SPMI:SCLK:THReshold on page 1859
SBUS<sb>:SPMI:SDATa:THReshold on page 1860
SBUS<sb>:SPMI:SCLK:HYSTeresis on page 1858
SBUS<sb>:SPMI:SDATa:HYSTeresis on page 1859

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
`SBUS<sb>:THReshold` on page 1364

#### 14.16.1.3 Display settings

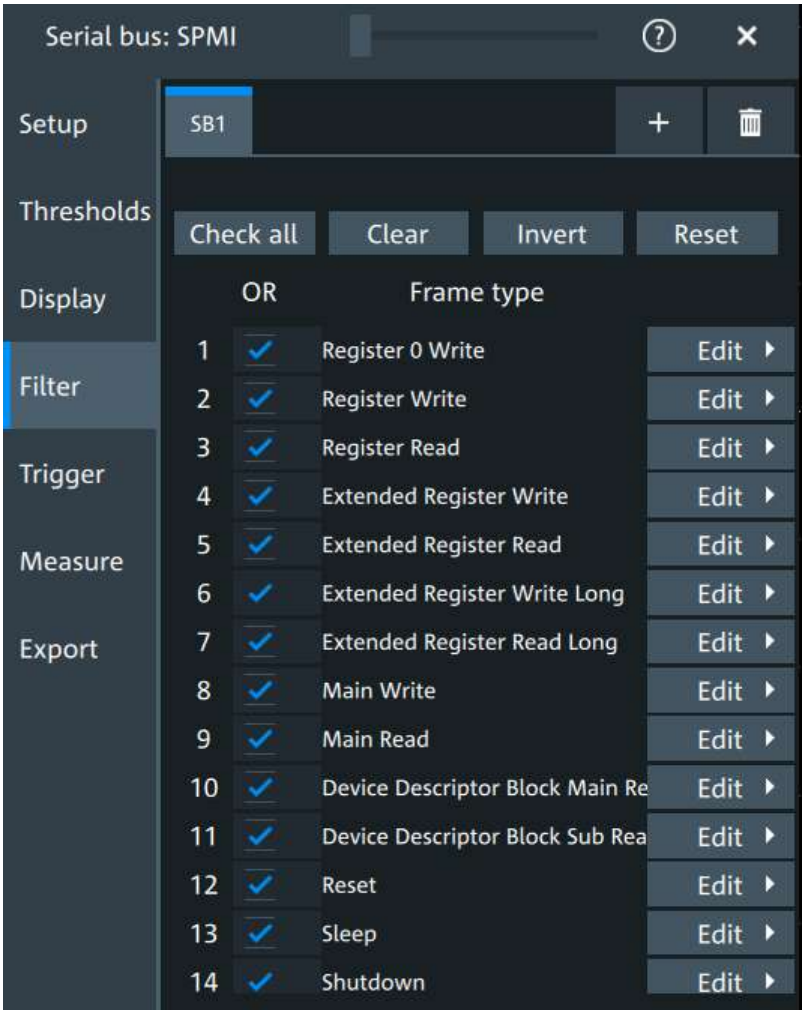
For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

### 14.16.2 SPMI filter

Access: "Menu" > "Apps" > "Protocol" tab > "SPMI" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

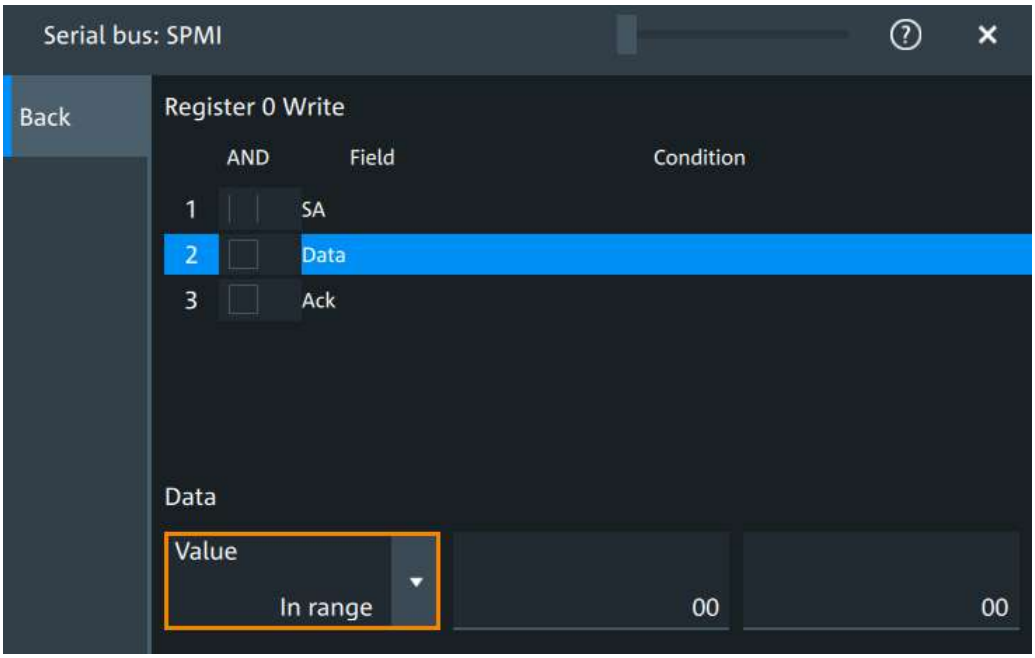
Remote command:
`SBUS<sb>:SPMI:FILTer:CHKall` on page 1862
`SBUS<sb>:SPMI:FILTer:CLR` on page 1862
`SBUS<sb>:SPMI:FILTer:INVert` on page 1862
`SBUS<sb>:SPMI:FILTer:RST` on page 1862

**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

Remote command:
`SBUS<sb>:SPMI:FILTer:FRAMe<fr>:ENABle` on page 1866
`SBUS<sb>:SPMI:FILTer:FRENable` on page 1866

**Edit**
Opens a dialog to define the details of the selected frame.




"Field" Enables the field type that you want to filter on for the selected frame. The available fields depend on the selected frame type.

Remote command:
`SBUS<sb>:SPMI:FILTer:FIENable` on page 1866
`SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:ENABle` on page 1866

"Condition" Displays the value condition for the selected field.

Remote command:
`SBUS<sb>:SPMI:FILTer:BIT` on page 1862
`SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:BIT` on page 1862

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:SPMI:FILTer:DMAX on page 1863
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1863
SBUS<sb>:SPMI:FILTer:DMIN on page 1863
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1863
SBUS<sb>:SPMI:FILTer:DOPerator on page 1864
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1864

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:SPMI:FILTer:IMAX on page 1865
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1865
SBUS<sb>:SPMI:FILTer:IMIN on page 1865
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1865
SBUS<sb>:SPMI:FILTer:IOPerator on page 1864
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1864

**Error type**
Enables filtering on the selected error type.

Remote command:
SBUS<sb>:SPMI:FILTer:ERRor<n>:ENABle on page 1864
SBUS<sb>:SPMI:FILTer:ERENable on page 1864

### 14.16.3 SPMI format configuration

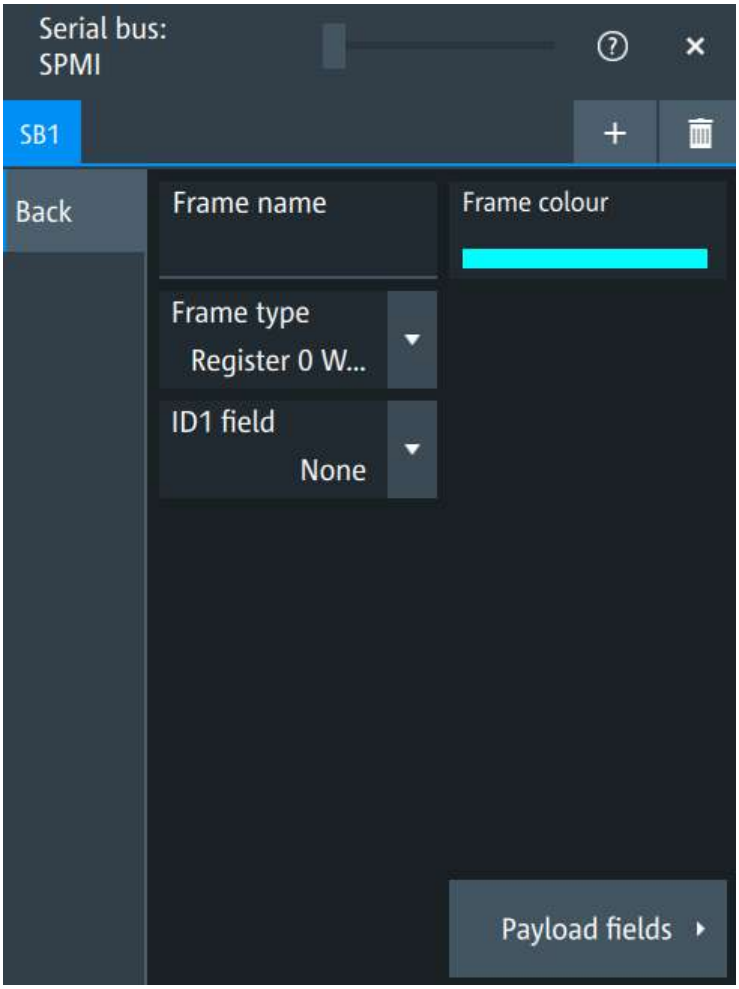
Access: "Menu" > "Apps" > "Protocol" tab > "SPMI" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
SBUS<sb>:SPMI:FORMat:SELect on page 1867

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:SPMI:FORMat:ADDFrame on page 1867

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:SPMI:FORMat:CLR on page 1868
SBUS<sb>:SPMI:FORMat:FCOunt? on page 1868

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:SPMI:FORMat:LOAD on page 1867

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:SPMI:FORMat:SAVE on page 1868

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:NAME on page 1869

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:COLor on page 1869

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:TYPE on page 1870

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID1Field on page 1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID2Field on page 1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID3Field on page 1870

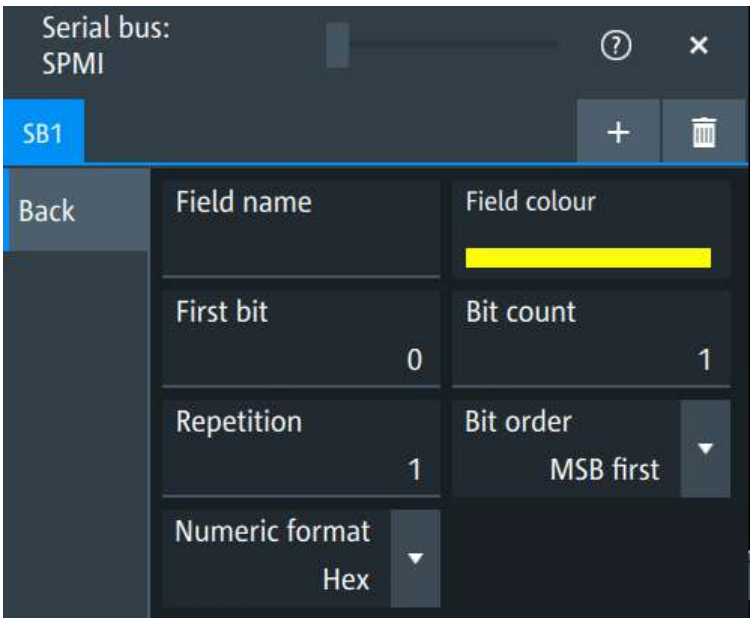
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID1Value on page 1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID2Value on page 1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID3Value on page 1870

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
`SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ADDField` on page 1869

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
`SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:NAME` on page 1871

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
`SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:COLor` on page 1871

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
`SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FSTBit` on page 1871

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.

- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  – Bit count = 2: truncated. Result: =00
  – Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  – Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  – Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1872

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
`SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:REPetition` on page 1872

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
`SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:BITorder` on page 1872

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:
- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  – "=1HL111000": valid (read as binary)
  – "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  – "=1010": valid (read as HEX, with a total of 16 bits)
  – "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)

– "=5X12": invalid
– "=1H33": invalid

Remote command:
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1873

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1874

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1874

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1874

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1875

**Parameter A,B**
Defines the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1875
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1875

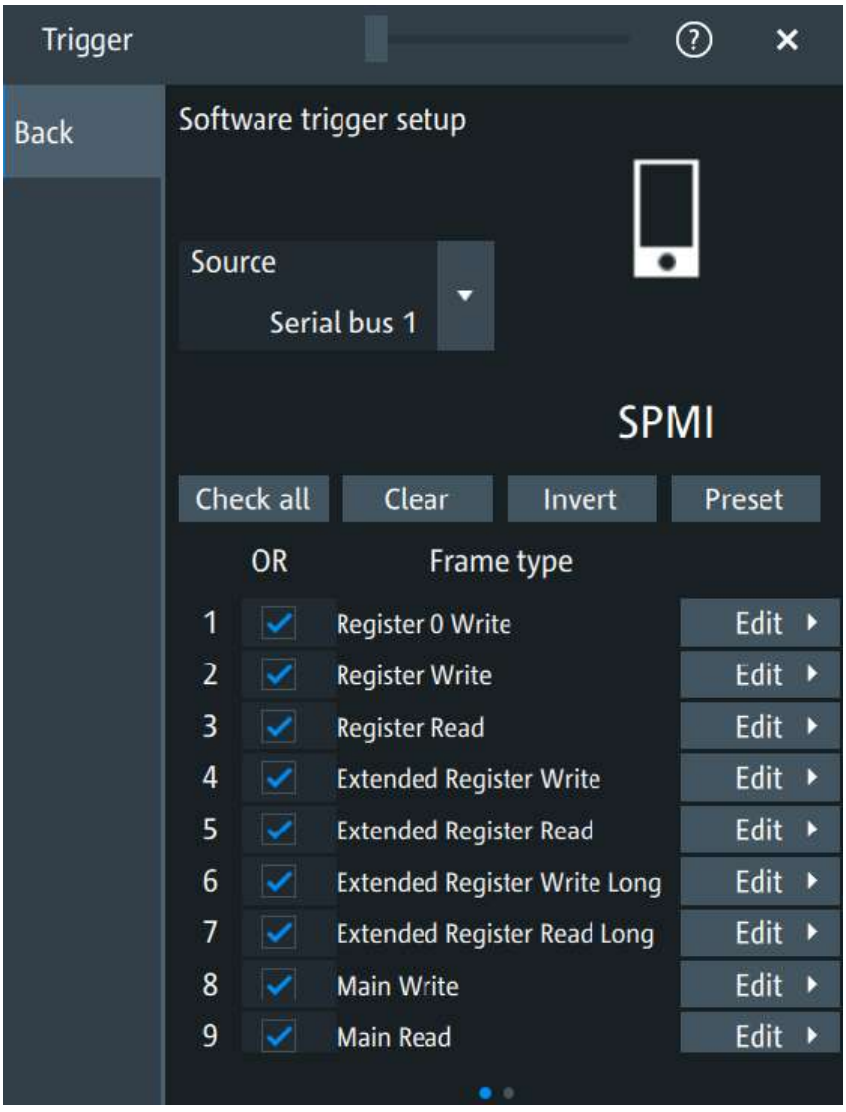
## 14.16.4 SPMI software trigger

### 14.16.4.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.16.4.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "SPMI" > "Trigger" tab > "Setup Software Trigger" > "Setup SB Trigger".




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

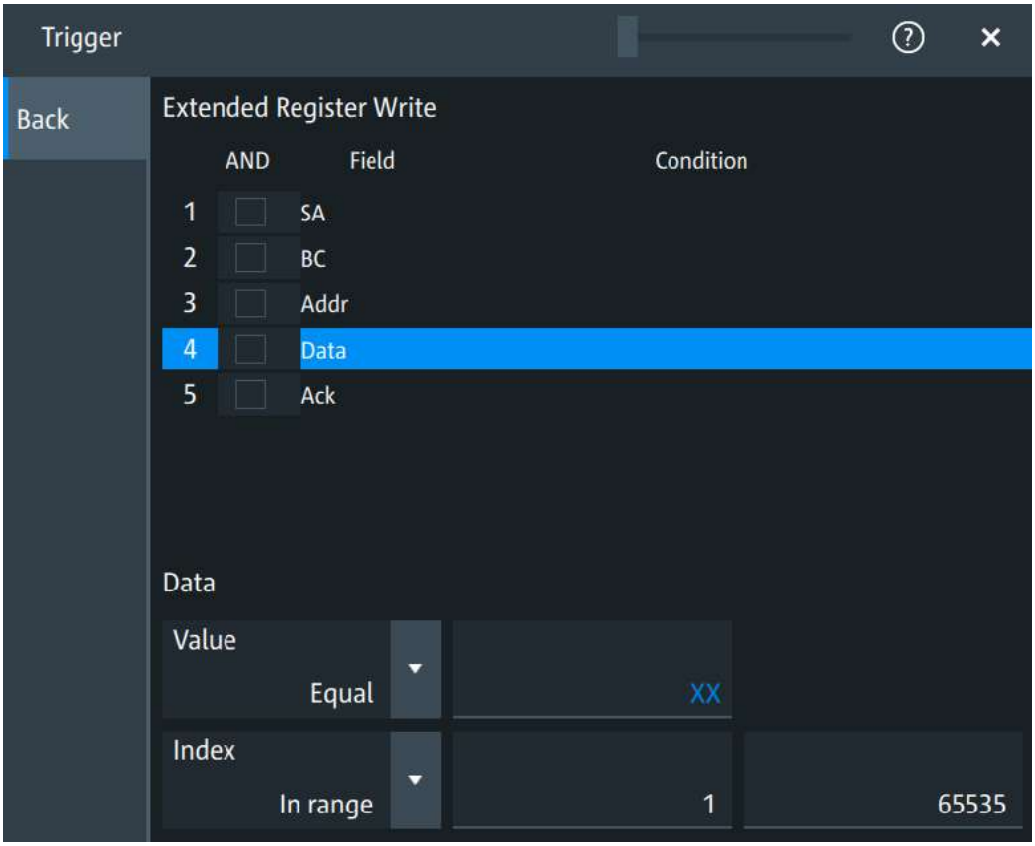
Remote command:
`TRIGger:SBSW:SPMI:CHKall` on page 1876
`TRIGger:SBSW:SPMI:CLR` on page 1877
`TRIGger:SBSW:SPMI:INVert` on page 1877
`TRIGger:SBSW:SPMI:RST` on page 1877

**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

Remote command:
`TRIGger:SBSW:SPMI:FRENable` on page 1877
`TRIGger:SBSW:SPMI:FRAMe<fr>:ENABle` on page 1877

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.




"Field" Enables the field type that you want to trigger on for the selected frame.

Remote command:
`TRIGger:SBSW:SPMI:FIENable` on page 1879
`TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:ENABle` on page 1879

"Condition" Displays the value condition for the selected field.

Remote command:
`TRIGger:SBSW:SPMI:BIT` on page 1877
`TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:BIT` on page 1877

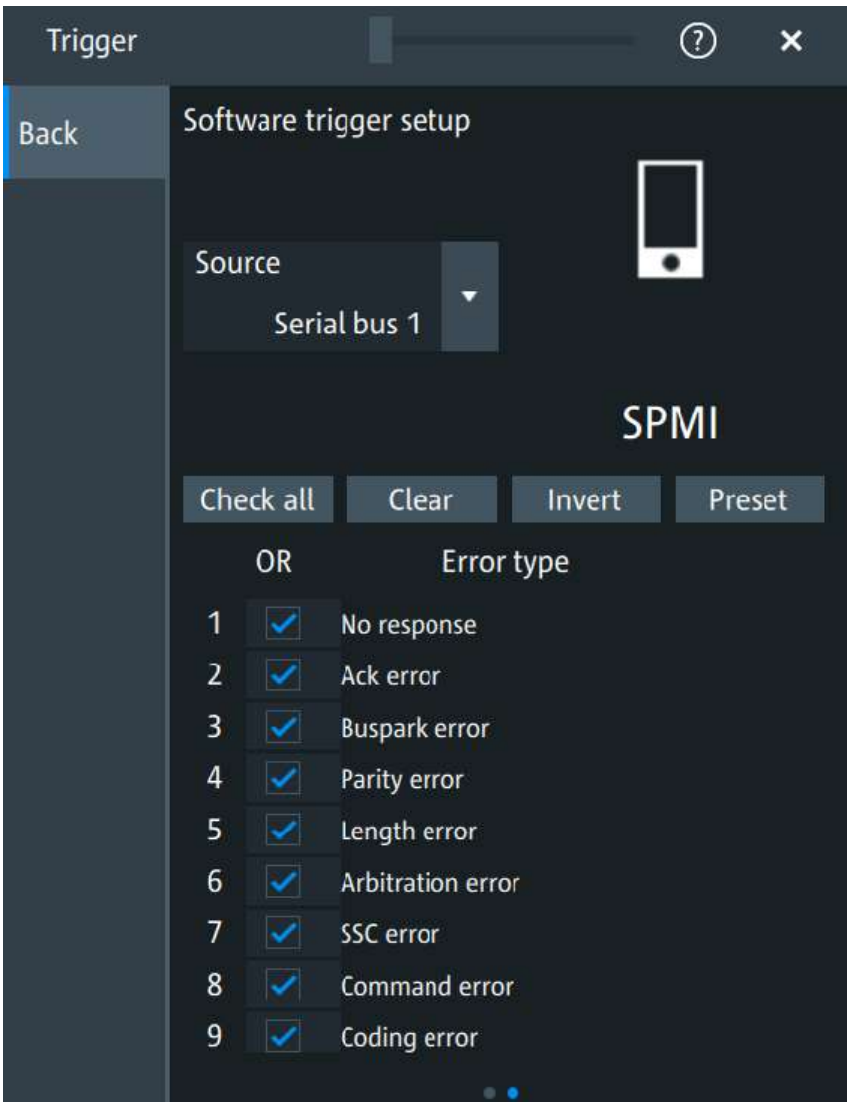
"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`TRIGger:SBSW:SPMI:DMAX` on page 1878
`TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DMAX` on page 1878
`TRIGger:SBSW:SPMI:DMIN` on page 1878
`TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DMIN` on page 1878
`TRIGger:SBSW:SPMI:DOPerator` on page 1879
`TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DOPerator` on page 1879

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
`TRIGger:SBSW:SPMI:IMAX` on page 1879
`TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IMAX` on page 1879
`TRIGger:SBSW:SPMI:IMIN` on page 1880
`TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IMIN` on page 1880
`TRIGger:SBSW:SPMI:IOPerator` on page 1880
`TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IOPerator` on page 1880

**Error type**
Enables triggering on the selected error type.




Remote command:
`TRIGger:SBSW:SPMI:ERENable` on page 1880
`TRIGger:SBSW:SPMI:ERRor<m>:ENABle` on page 1880

### 14.16.5 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.16.6 SPMI decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

For more information on decoding SPMI, you can also refer to the video, available on the Rohde & Schwarz YouTube channel: Decoding SPMI with MXO Series Oscilloscopes.

#### Decode results table

The "Decode results" table contains information about all decoded frames.

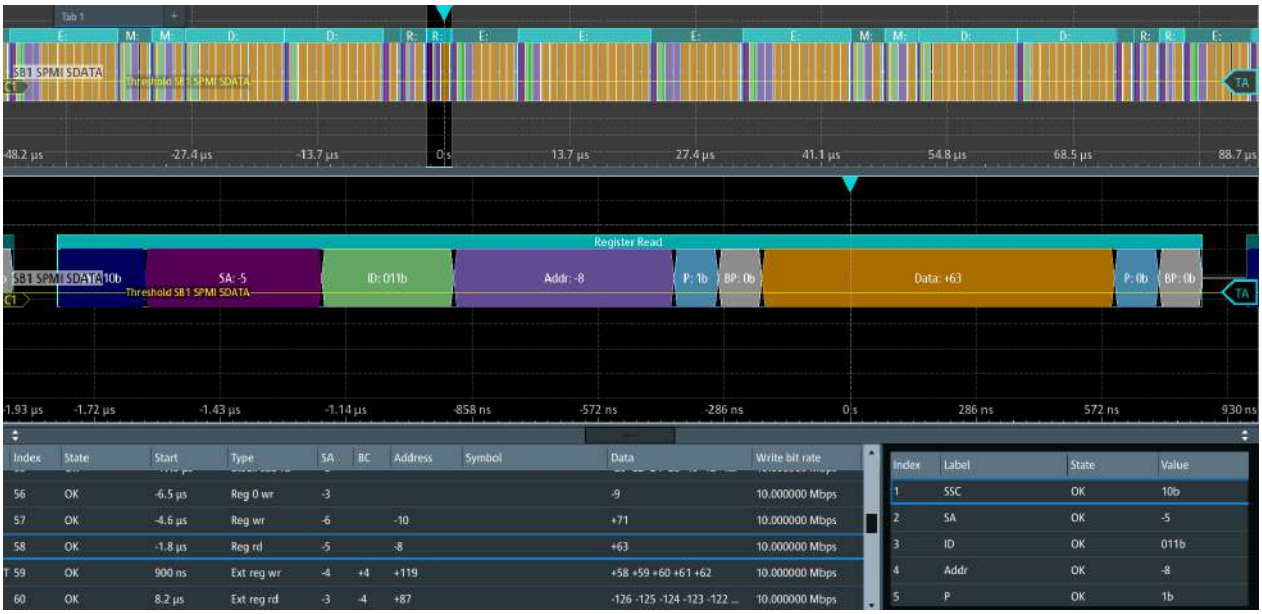



*Figure 14-48: Decoded SPMI signal*

The decode results table contains information about all decoded frames.

*Table 14-28: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Index of the decoded frame |
| State | Overall state of the frame |
| Start | Time of frame start |
| Type | The type of frame |
| SA | Secondary node address. Select the data format in the "Display" tab. |

| Column | Description |
|---|---|
| BC | Byte count value. Select the data format in the "Display" tab. |
| Address | Register address. Select the data format in the "Display" tab. |
| Symbol | Symbolic label of the frame. Available only, if "Show symbols" is enabled in the "Display" tab and a symbols list is loaded. |
| Data | Data value. Select the data format in the "Display" tab. |
| Write bit rate | Value of the write bit rate |
| Read bit rate | Value of the read bit rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

***Table 14-29: Content of the frame details table***

| Column | Description |
|---|---|
| Index | Number of the decoded field |
| Label | Name of the field |
| State | State of the field |
| Value | Value of the field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.17, "SPMI (option R&S MXO3-K550)", on page 1856.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example SPMI export file

```
Index,Start,Stop,State,Type,SA,BC,Address,Write bit rate,Read bit rate
1,-0.000405,-0.0004032,'OK','RZWR',Dh,---,---,10000000,---
2,-0.0004031,-0.0004004,'OK','RWR',Ah,---,16h,10000000,---
3,-0.0004003,-0.0003977,'OK','RRD',Bh,---,18h,10000000,10000000
4,-0.0003976,-0.0003904,'OK','ERWR',Ch,4h,77h,10000000,---
5,-0.0003903,-0.000376,'OK','ERRD',Dh,Ch,57h,10000000,10000000

Details frame 1
Index,Label,State,Value
1,'"SSC"','OK','"10b"'
```

```
2,'"SA"','OK','"Dh"'
3,'"ID"','OK','"1b"'
4,'"Data"','OK','"77h"'
5,'"P"','OK','"1b"'
6,'"BP"','OK','"0b"'
7,'"Ack"','OK','"1b"'
8,'"BP"','OK','"0b"'

Details frame 2
Index,Label,State,Value
1,'"SSC"','OK','"10b"'
2,'"SA"','OK','"Ah"'
3,'"ID"','OK','"010b"'
4,'"Addr"','OK','"16h"'
5,'"P"','OK','"1b"'
6,'"Data"','OK','"47h"'
7,'"P"','OK','"1b"'
8,'"BP"','OK','"0b"'
9,'"Ack"','OK','"1b"'
10,'"BP"','OK','"0b"'
```

## 14.17 RFFE (mobile electronics, option R&S MXO3-K550)

**Requirements**

For performing RFFE decode measurements, you need the following:

- MXO 3 with 2 available channels. The channel can be:
  - Analog channels (C1-C4)
  - Logic channels (D0-D15) (requires option R&S MXO3-B1)
- Two probes
- Option R&S MXO3-K550


- About the RFFE protocol ..... 812
- RFFE configuration ..... 814
- RFFE filter ..... 816
- RFFE format configuration ..... 820
- RFFE software trigger ..... 826
- Measure ..... 830
- RFFE decode results ..... 830


### 14.17.1 About the RFFE protocol

The RFFE interface is specified in the "MIPI® Alliance Specification for RF Front-End Control Interface". The RFFE interface is used by the radio frequency front-end interface chips in most LTE-Advanced platforms and in smart phones in general. RFFE is

a replacement for existing standards like SPI and I²C that do no meet performance requirements.

### Bus structure

RFFE is a two-wire, serial interface that connects up to 4 main devices (Radio Frequency IC, RFIC) to up to 15 sub devices (front-end modules, FEM) on a single RFFE bus. A sub device has read-write capability, or it is write only. Only one of the main devices is the active main device (bus owner main device, BOM), which can initiate command sequences on the bus.

The interface has two lines: one clock signal (SCLK) controlled by the main device, and a serial bidirectional data signal (SDATA). Furthermore, a VIO supply/reference voltage from a common source is applied to all components on the bus.

### Command sequences

Protocol messages are called command sequences in RFFE. The standard defines various command sequences to accomplish read and write access to sub devices and to non-active main devices. Command sequences are initiated by the BOM main device on the SDATA line.

In general, a command sequence consists of:

- Sequence start condition (SSC)
  Two bits: 1 followed by 0 on SDATA while SCLK is at logic level zero.
- Command frame
  Consist of a 4-bit sub device address field (SA), followed by 8 command payload bits and a single parity bit.
- Address and/or data frames, depending on the command sequence
  A frame consists of 8 data bits or 8 register address bits, followed by a single parity bit. The number of address and data frames varies depending on the command sequence type.
- Bus park cycle (BP)
  A BP cycle is sent at the end of a command sequence, and when the device transfers control of SDATA to another device.

Between the end of a command sequence and the beginning of a new command sequence, the bus is in idle condition at least for 10 ns.

The bits are sent MSB first.

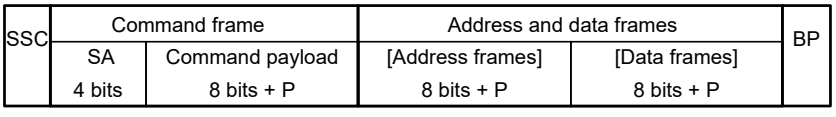



*Figure 14-49: General structure of a RFFE Write command sequence*

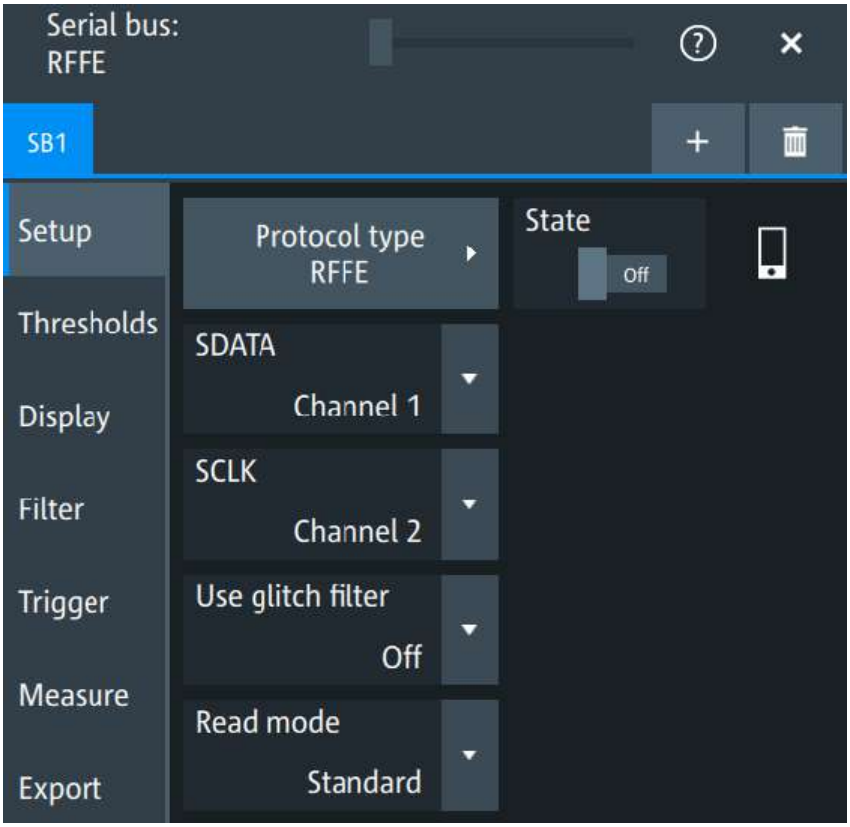
| SSC | Command frame | | Address and data frames | | | BP |
|---|---|---|---|---|---|---|
| | SA<br>4 bits | Command payload<br>8 bits + P | [Address frames]<br>8 bits + P | BP | [Data frames]<br>8 bits + P | |

*Figure 14-50: General structure of a RFFE Read command sequence*

## 14.17.2 RFFE configuration

### 14.17.2.1 RFFE configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "RFFE" > "Setup".




Make sure that the tab of the correct serial bus is selected.

**Protocol type**
Selects the protocol type to be decoded.

Remote command:
`SBUS<sb>:TYPE` on page 1362

**State**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
SBUS<sb>[:STATe] on page 1362

**SDATA**
Sets the source of the data line.

Remote command:
SBUS<sb>:RFFE:DATA:SOURce on page 1889

**SCLK**
Sets the source of the clock line.

Remote command:
SBUS<sb>:RFFE:CLOCk:SOURce on page 1888

**Use glitch filter, Glitch filter width**
Enables the glitch filter on the SCLK and SDATA lines to improve decode accuracy.

The "Glitch filter width" field sets the maximum glitch width to be ignored.

Remote command:
SBUS<sb>:RFFE:GFILter on page 1890
SBUS<sb>:RFFE:GFWidth on page 1890

**Read mode**
Selects, if the standard or synchronous read ("sRead") mode is used.

Remote command:
SBUS<sb>:RFFE:RDMD on page 1890

### 14.17.2.2 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "RFFE" > "Thresholds".

**Threshold**
Sets the threshold for the data and clock channel. Enter the value directly in the fields.

Additional to the threshold, you can also set a hysteresis.

Remote command:
SBUS<sb>:RFFE:DATA:THReshold on page 1889
SBUS<sb>:RFFE:DATA:HYSTeresis on page 1889
SBUS<sb>:RFFE:CLOCk:THReshold on page 1888
SBUS<sb>:RFFE:CLOCk:HYSTeresis on page 1888

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
SBUS<sb>:THReshold on page 1364

#### 14.17.2.3 Display settings

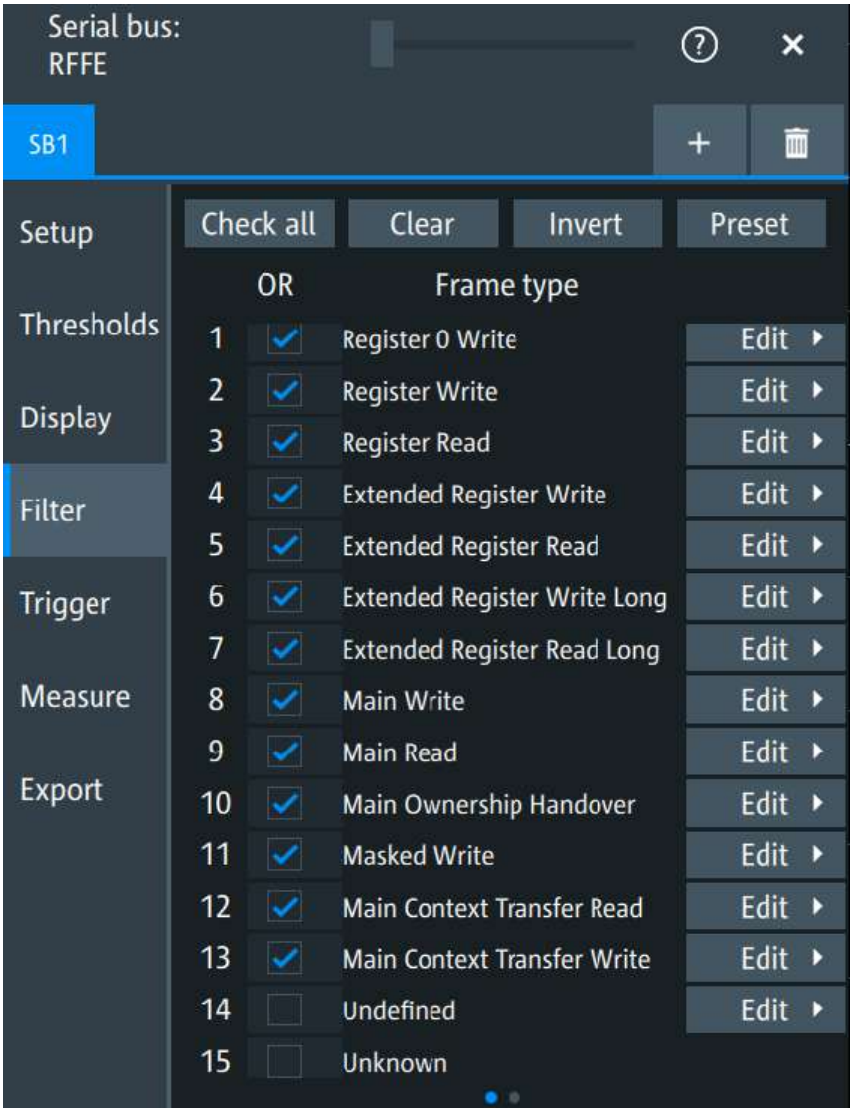
For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

### 14.17.3 RFFE filter

Access: "Menu" > "Apps" > "Protocol" tab > "RFFE" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
`SBUS<sb>:RFFE:FILTer:CHKall` on page 1892
`SBUS<sb>:RFFE:FILTer:CLR` on page 1892

SBUS<sb>:RFFE:FILTer:INVert on page 1892
SBUS<sb>:RFFE:FILTer:RST on page 1892

**Enable**
Enables the filtering on RFFE frames. Only the frames that match the selected filter conditions are displayed.

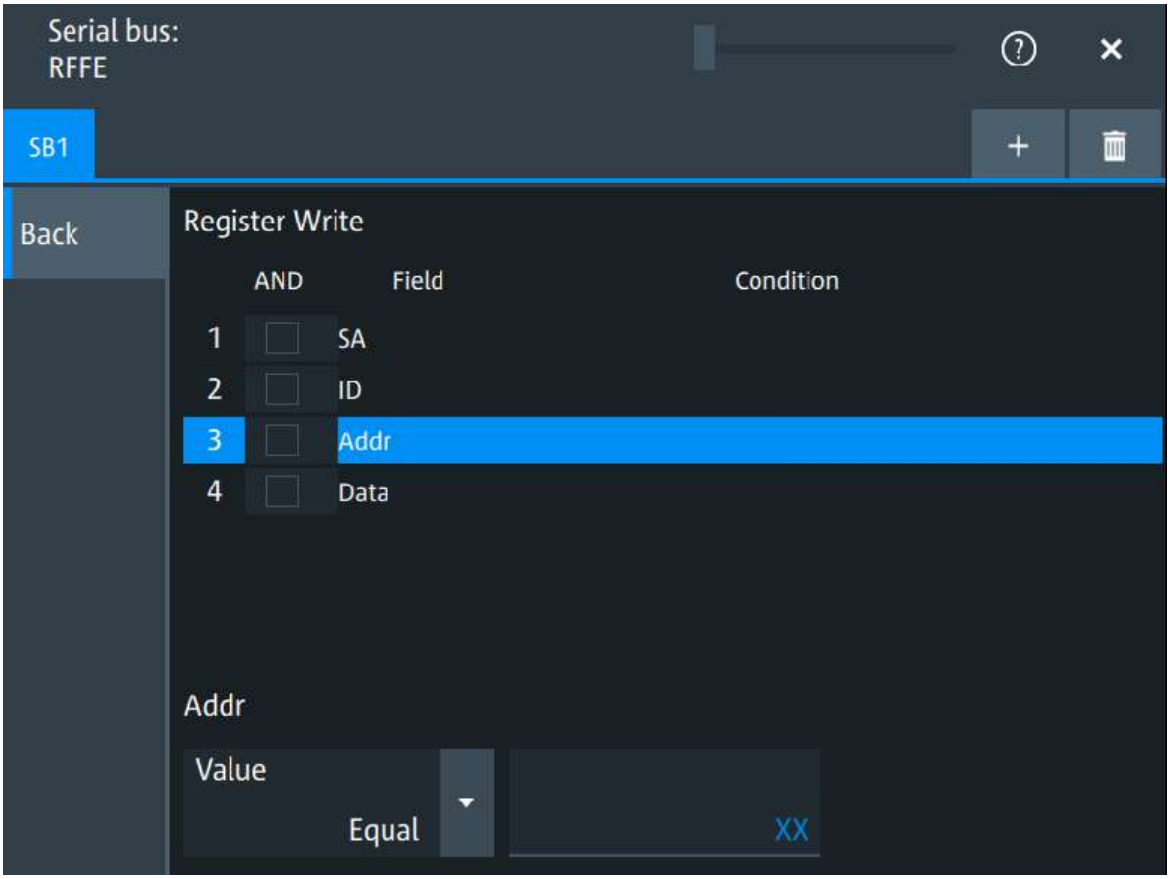
Remote command:
SBUS<sb>:RFFE:FILTer:FRENable on page 1893
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:ENABle on page 1895

**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: "Register 0 Write", "Register Write", "Register Read", "Extended Register Write", "Extended Register Read", "Extended Register Write Long", "Extended Register Read Long", "Main Write", "Main Read".

**Edit**
Opens a dialog to define the details of the selected frame.



"Field" Enables the field type that you want to filter on for the selected frame. The following fields are available: "SA", "ID", "Data", "Addr", "BC", "MID", "Data MSB", "Data LSB".

Remote command:
SBUS<sb>:RFFE:FILTer:FIENable on page 1895
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:ENABle on page 1895

"Condition" Displays the value condition for the selected field.

Remote command:
SBUS<sb>:RFFE:FILTer:BIT on page 1893
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:BIT on page 1893

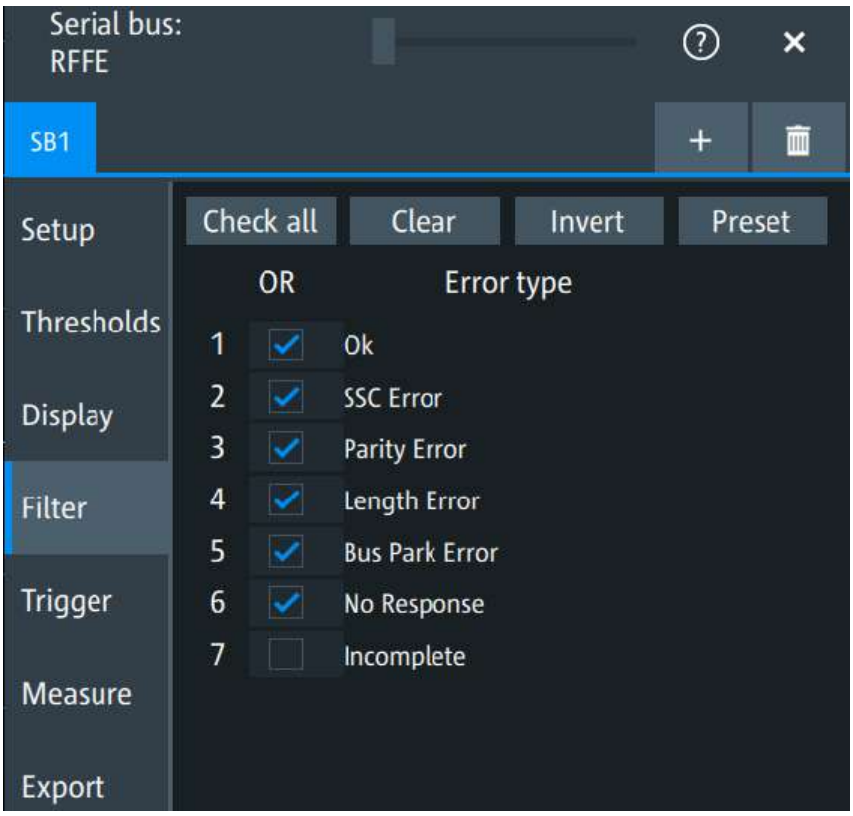
"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:RFFE:FILTer:DMAX on page 1893
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1893
SBUS<sb>:RFFE:FILTer:DMIN on page 1894
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1894
SBUS<sb>:RFFE:FILTer:DOPerator on page 1894
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1894

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:RFFE:FILTer:IMAX on page 1895
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1895
SBUS<sb>:RFFE:FILTer:IMIN on page 1895
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1895
SBUS<sb>:RFFE:FILTer:IOPerator on page 1896
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1896

**Error type**
Enables filtering on the selected error type.




The following error types are available: "OK", "SSC Error", "Parity error", "Length error", "Bus park error", "No response".

Remote command:
`SBUS<sb>:RFFE:FILTer:ERENable` on page 1896
`SBUS<sb>:RFFE:FILTer:ERRor<n>:ENABle` on page 1896

### 14.17.4 RFFE format configuration

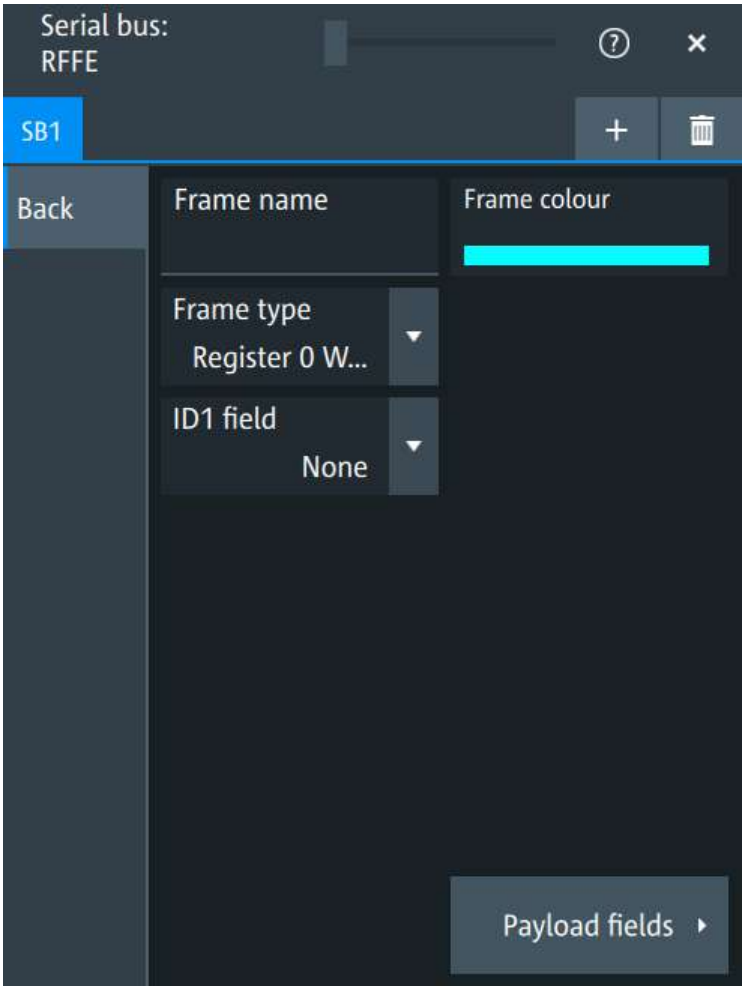
Access: "Menu" > "Apps" > "Protocol" tab > "RFFE" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
`SBUS<sb>:RFFE:FORMat:SELect` on page 1897

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:RFFE:FORMat:ADDFrame on page 1897

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:RFFE:FORMat:CLR on page 1898
SBUS<sb>:RFFE:FORMat:FCOunt? on page 1898

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:RFFE:FORMat:LOAD on page 1898

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:RFFE:FORMat:SAVE on page 1898

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:NAME on page 1899

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:COLor on page 1900

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:TYPE on page 1900

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID1Field on page 1900
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID2Field on page 1900
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID3Field on page 1900

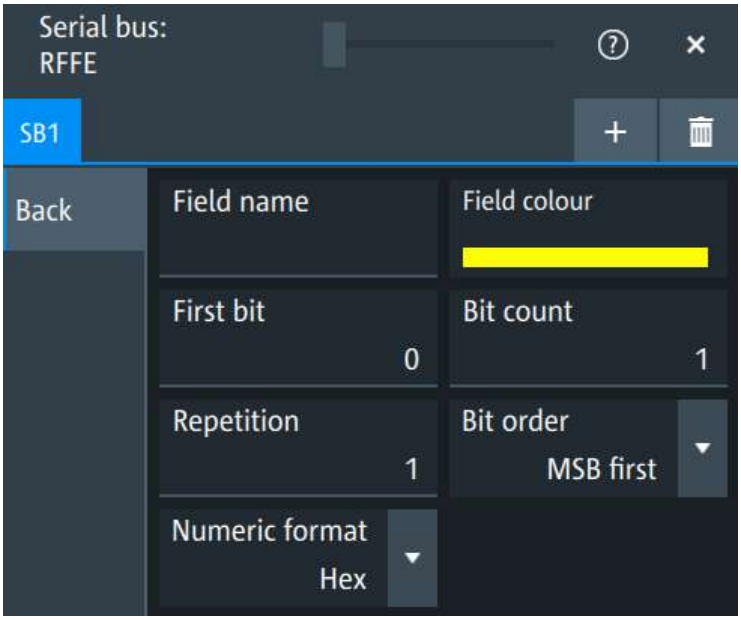
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID1Value on page 1901
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID2Value on page 1901
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID3Value on page 1901

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ADDField on page 1899

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1901

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1901

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1902

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.

- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  – Bit count = 2: truncated. Result: =00
  – Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  – Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  – Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1902

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
`SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:REPetition` on page 1902

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
`SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:BITorder` on page 1903

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:
- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  – "=1HL111000": valid (read as binary)
  – "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  – "=1010": valid (read as HEX, with a total of 16 bits)
  – "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)

– "=5X12": invalid
– "=1H33": invalid

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1903

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1904

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1904

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1905

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1905

**Parameter A,B**
Defines the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1905
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1905

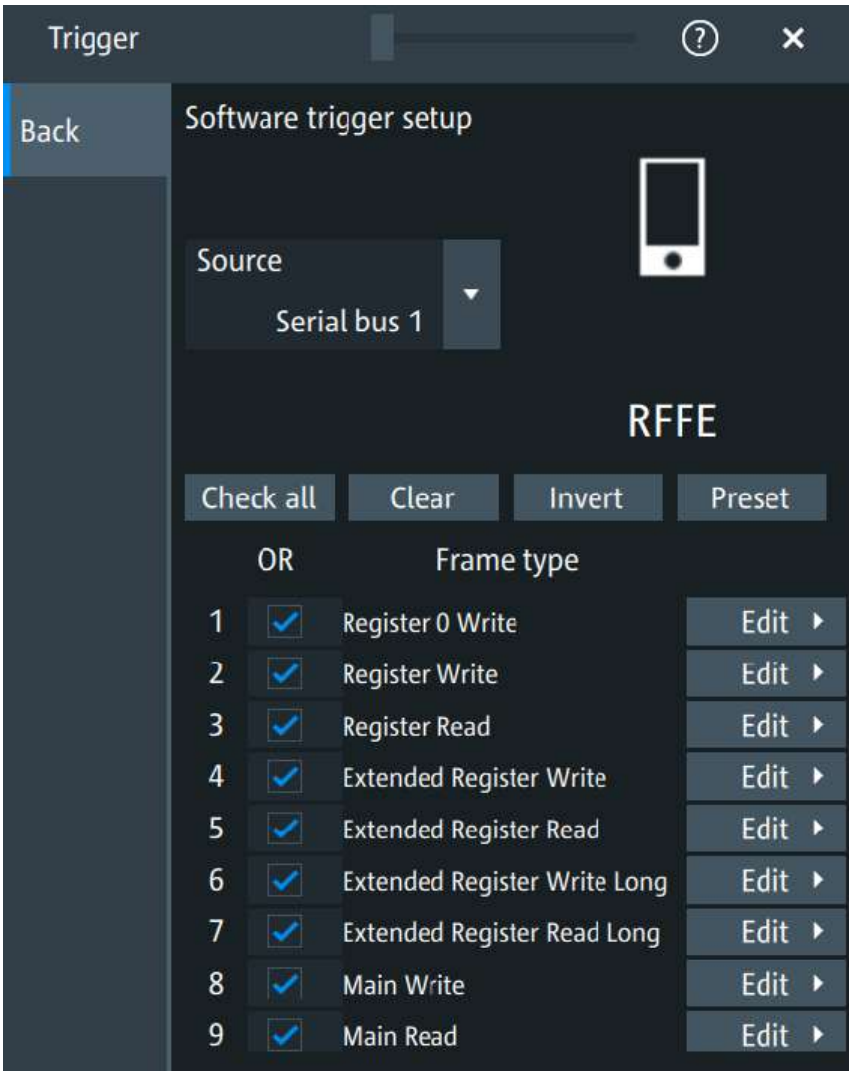
## 14.17.5 RFFE software trigger

### 14.17.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.17.5.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "RFFE" > "Trigger" tab > "Setup Software Trigger" > "Setup SB Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:RFFE:CHKall` on page 1907
`TRIGger:SBSW:RFFE:CLR` on page 1907
`TRIGger:SBSW:RFFE:INVert` on page 1907
`TRIGger:SBSW:RFFE:RST` on page 1907

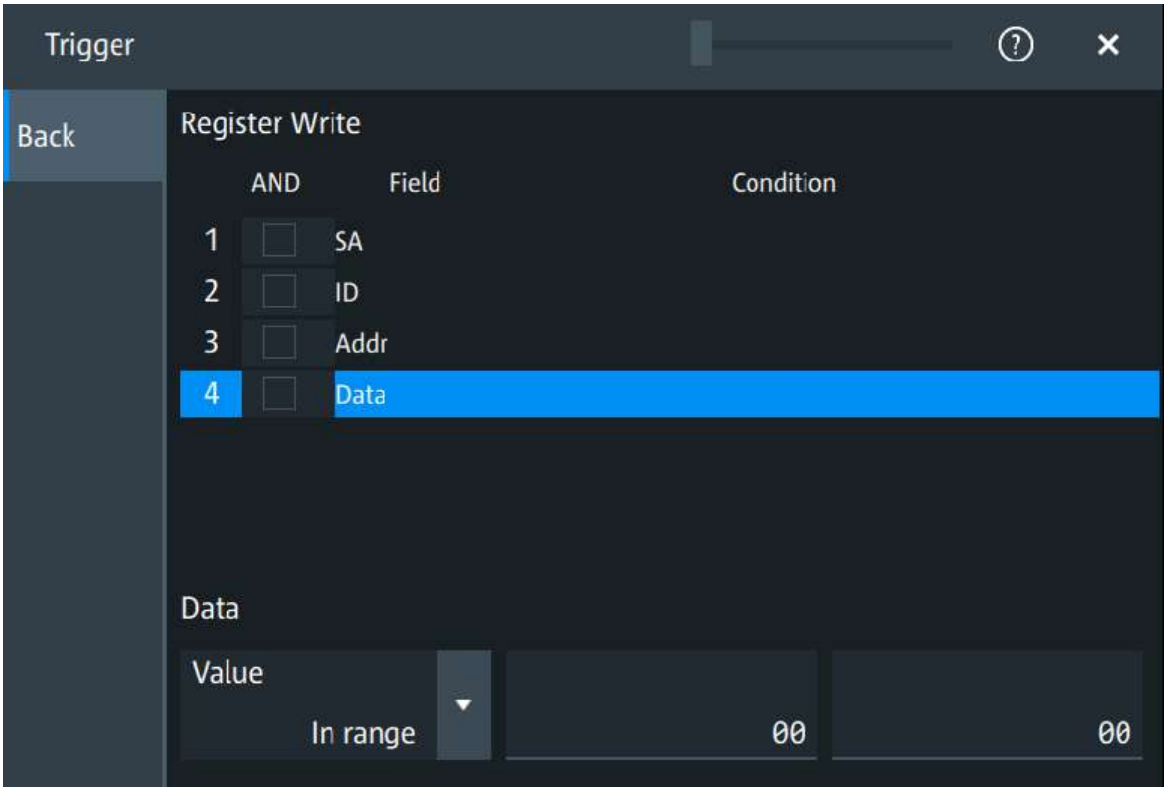
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: : "Register 0 Write", "Register Write", "Register Read", "Extended Register Write", "Extended Register Read", "Extended Register Write Long", "Extended Register Read Long", "Main Write", "Main Read".

Remote command:
`TRIGger:SBSW:RFFE:FRENable` on page 1907
`TRIGger:SBSW:RFFE:FRAMe<fr>:ENABle` on page 1907

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.




"Field" Enables the field type that you want to trigger on for the selected frame.
The available fields are: "SA", "ID", "Data", "Addr", "BC", "MID", "Data MSB", "Data LSB".

Remote command:
`TRIGger:SBSW:RFFE:FIENable` on page 1909
`TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:ENABle` on page 1909

"Condition" Displays the value condition for the selected field.

Remote command:
`TRIGger:SBSW:RFFE:BIT` on page 1908
`TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:BIT` on page 1908

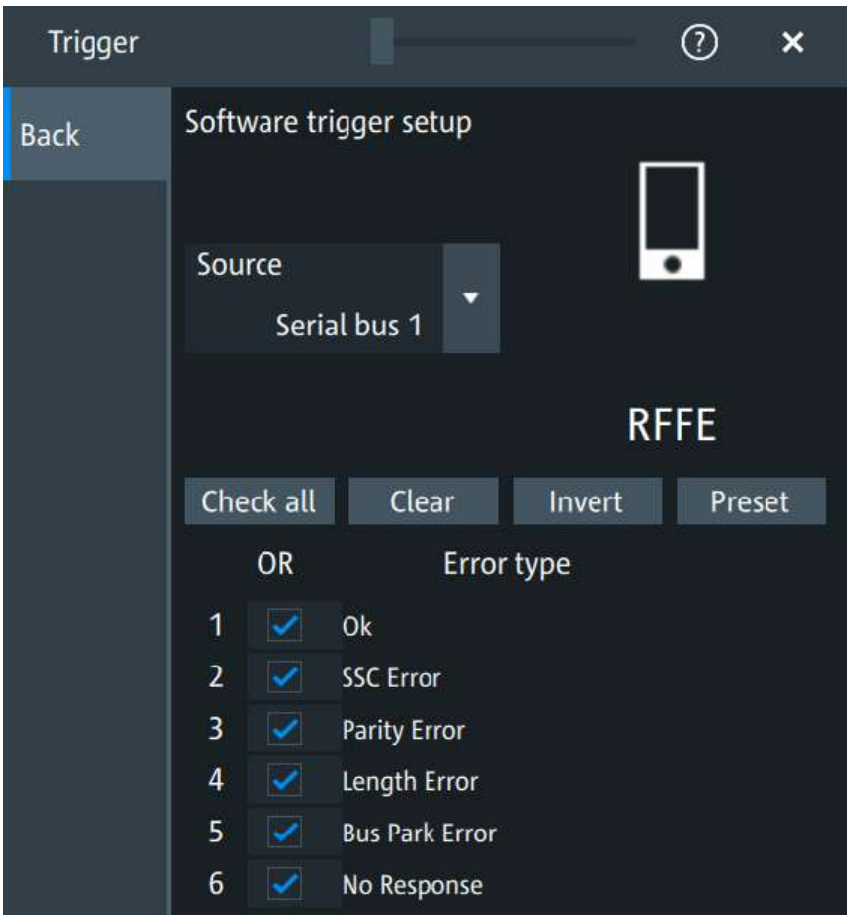
"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`TRIGger:SBSW:RFFE:DMAX` on page 1908
`TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DMAX` on page 1908
`TRIGger:SBSW:RFFE:DMIN` on page 1908
`TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DMIN` on page 1908
`TRIGger:SBSW:RFFE:DOPerator` on page 1909
`TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DOPerator` on page 1909

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:RFFE:IMAX on page 1910
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IMAX on page 1910
TRIGger:SBSW:RFFE:IMIN on page 1910
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IMIN on page 1910
TRIGger:SBSW:RFFE:IOPerator on page 1910
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IOPerator on page 1910

**Error type**
Enables triggering on the selected error type.




The following error types are available: "OK", "SSC Error", "Parity error", "Length error", "Bus park error", "No response".

Remote command:
TRIGger:SBSW:RFFE:ERENable on page 1911
TRIGger:SBSW:RFFE:ERRor<m>:ENABle on page 1911

### 14.17.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.17.7 RFFE decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

#### Decode results table

The "Decode results" table contains information about all decoded frames.

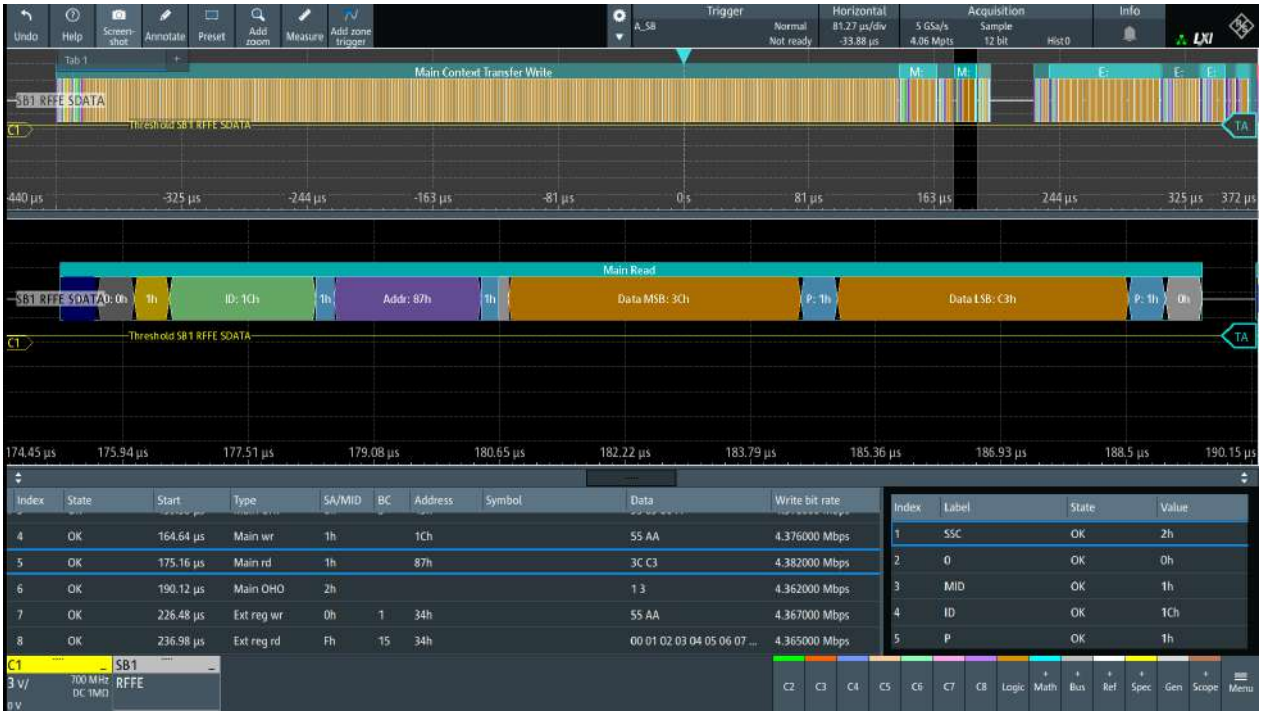



*Figure 14-51: Decoded RFFE signal*

The decode results table contains information about all decoded frames.

*Table 14-30: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Index of the decoded frame |
| State | Overall state of the frame |
| Start | Times of frame start |
| Type | The type of command |
| SA/MID | Address of the sub device |

| Column | Description |
|---|---|
| BC | Byte count |
| Address | Register address |
| Symbol | Symbolic label of the frame. Available only, if "Show symbols" is enabled in the "Display" tab and a symbols list is loaded. |
| Data | Values of the data bytes. Select the data format in the "Display" tab. |
| Write bit rate | Value of the write bit rate |
| Read bit rate | Value of the read bit rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

***Table 14-31: Content of the frame details table***

| Column | Description |
|---|---|
| Index | Number of the decoded field |
| Label | Name of the field |
| State | State of the field |
| Data | Value of the field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.18.5, "Decode results", on page 1912.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example of RFFE export file

```
Index,Start,Stop,State,Type,SA, BC,    Addr,    Write BR,    Read BR
1    -0.1000402    -0.0003965    OK    MSKW    5h    ---    2Eh    4376000    ---
2    -0.0003957    0.00013858    OK    MCTW    0h    255    3Ah    4375000    ---
3    0.00013938    0.00016396    OK    MCTR    0h    3      43h    4378000    2187000
4    0.00016464    0.00017436    OK    MWR     1h    ---    1Ch    4376000    ---
5    0.00017516    0.00018944    OK    MRD     1h    ---    87h    4382000    2190000
6    0.00019012    0.00019824    OK    MOHO    2h    ---    ---    4362000    2195000
7    0.00022648    0.00023618    OK    ERWR    0h    1      34h    4367000    ---
8    0.00023698    0.00030888    OK    ERRD    Fh    15     34h    4365000    2187000
```

```
Details frame 1
Index    Label    State    Value
1    'Mask'    'OK'    3Fh
2    'Data'    'OK'    15h

Details frame 3
Index    Label    State    Value
1    'Data: 1'    'OK'    55h
2    'Data: 2'    'OK'    83h
3    'Data: 3'    'OK'    00h
4    'Data: 4'    'OK'    FFh

Details frame 4
Index    Label    State    Value
1    'Data MSB'    'OK'    55h
2    'Data LSB'    'OK'    AAh
```

## 14.18 10BASE-T1S (automotive Ethernet, option R&S MXO3-K560)

10BASE-T1S technology provides a 10 Mbit/s, multidrop transmission over a single pair physical layer. It is defined in the IEEE standard 802.3cg-2019 specification.

**Requirements**

For performing 10BASE-T1S decode measurements, you need the following:

- MXO 3. One of the analog input channels is used for analysis.
- Option R&S MXO3-K560

10BASE-T1S does not support logical channels for decoding, because it requires two thresholds, which are not available with this option.

- 10BASE-T1S basics ..... 832
- 10BASE-T1S configuration ..... 833
- 10BASE-T1S filter ..... 838
- PPIFormat.PPIFrame.Name10BASE-T1S format configuration ..... 840
- 10BASE-T1S software trigger ..... 846
- Measure ..... 848
- 10BASE-T1S decode results ..... 849

### 14.18.1 10BASE-T1S basics

This option provides in-demand monitor and decode functions for 10BASE-T1S, where S stands for short reach. This standard, officially IEEE 802.3cg, attracted attention even before the specification was published in 2020. It was specified with automotive requirements in mind, as an alternative to CAN, CAN-FD or LIN. Target application is short-range data transmission with data rates up to 10 Mbit/s.

10BASE-T1S supports half-duplex and full-duplex communication, providing a point-to-point direct connection between two nodes. It is also suited for using a multidrop topology with up to eight nodes, connected on a single 25 m bus segment. Typical application examples are automotive interior lighting, parking radar, active suspension, motor management, and audio connections. It also supports networks of incompatible devices and different physical hardware.

Advantages over previously established 10 Mbit/s protocols include:

- Connecting all sensors over the same unshielded twisted pair cable, which saves interfaces, connectors, and cables, hence reducing weight, cost, and complexity.
- The simplified architecture, entirely Ethernet-based, eliminates any need for gateways between the networks. The unified design is also simpler to test and maintain.
- Maximized data rate: With Physical Layer Collision Avoidance (PLCA), only the node that currently has the right to transmit can send data. Nodes with no data to transmit will be given a reduced slot to send data, increasing the available bandwidth.

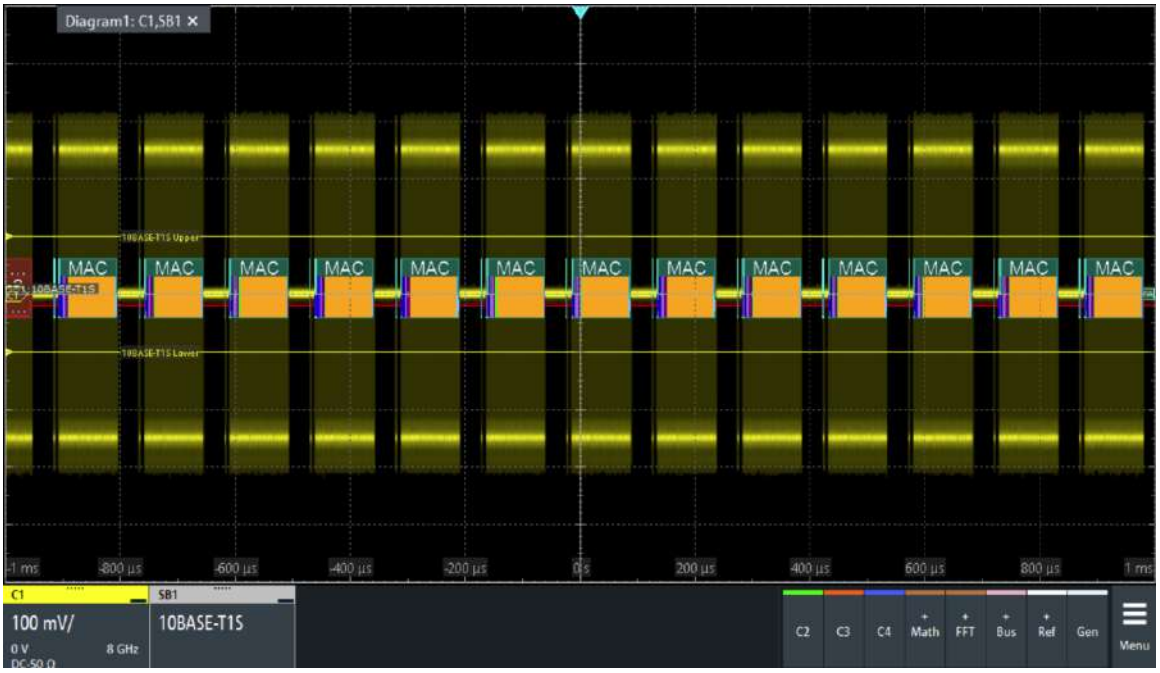
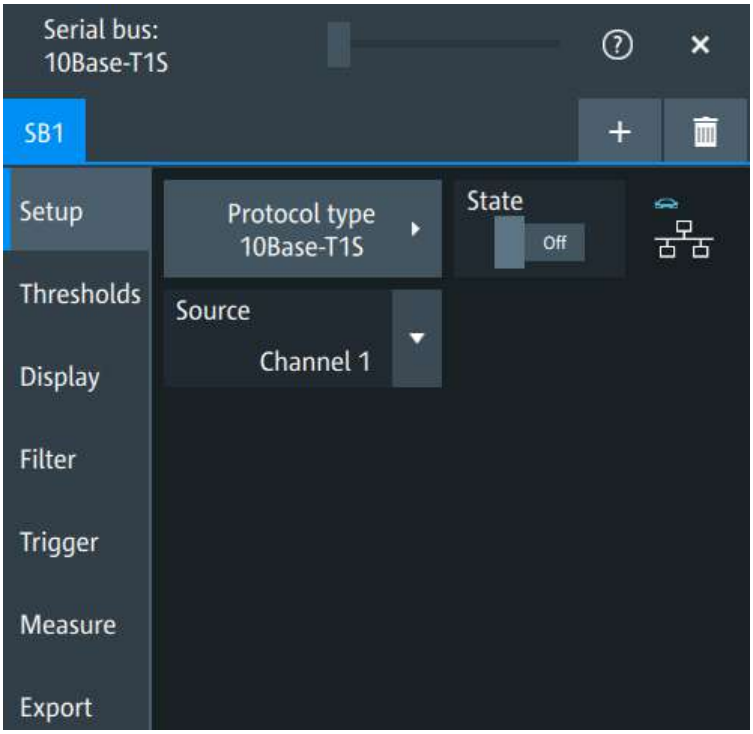



*Figure 14-52: Example of a decoded 10BASE-T1S signal*

## 14.18.2 10BASE-T1S configuration

### 14.18.2.1 10BASE-T1S configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "10BASE-T1S" > "Setup".




Make sure that the tab of the correct serial bus is selected.

**Protocol type**
Selects the protocol type to be decoded.

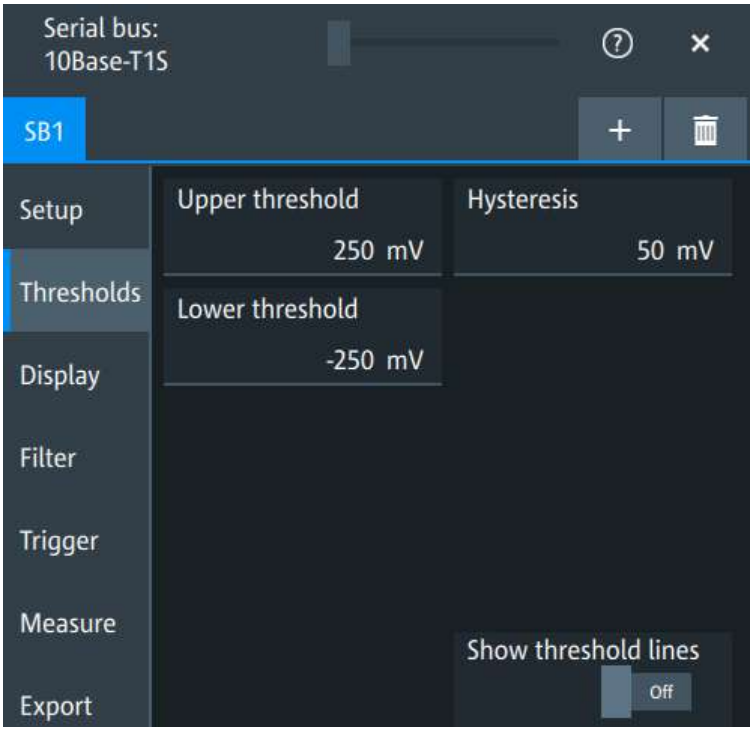
Remote command:
`SBUS<sb>:TYPE` on page 1362

**Source**
Selects the source channel for the 10BASE-T1S protocol. Only analog channels can be used.

Remote command:
`SBUS<sb>:TNOS:SOURce` on page 1921

### 14.18.2.2 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "10BASE-T1S" > "Thresholds".




**Threshold**
Sets the threshold for the data channel. Enter the value directly in the fields.

Additional to the threshold, you can also set a hysteresis.

Remote command:
`SBUS<sb>:TNOS:THReshold:LOWer` on page 1921
`SBUS<sb>:TNOS:THReshold:UPPer` on page 1922
`SBUS<sb>:TNOS:THReshold:HYSTeresis` on page 1921

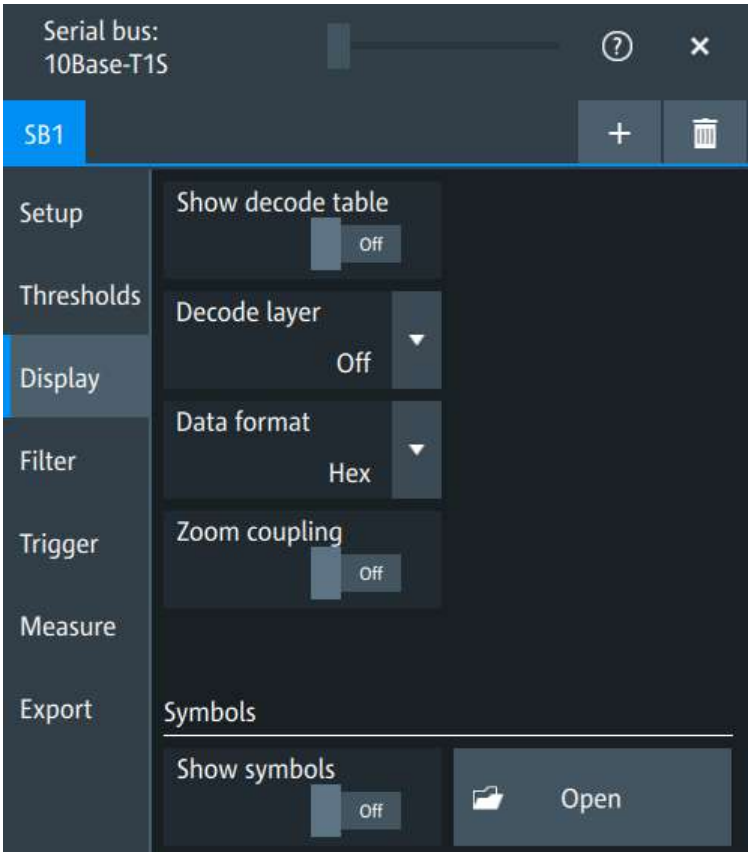
**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
`SBUS<sb>:THReshold` on page 1364

### 14.18.2.3 Display settings

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.




**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display:

| | |
|---|---|
| "Off" | Disables the display of the decode layer. |
| "Edges" | Enables the display of all edges. |
| "Bits" | Enables the display of the raw, encoded bits. |
| "Symbols" | Enables the display of the raw, encoded symbols. |
| "Descrambled bits" | Enables the display of the fully decoded and descrambled binary data, representing the original information being transmitted over the 10BASE-T1S link. |

**Data format**
See "Data format" on page 493.

**Zoom coupling**
See "Zoom coupling" on page 494.

**Show symbols**
You can load symbol lists and activate their usage for decoding. As a result, an additional "Symbol" column appears in the "Decode results" table, containing the symbolic label.

For more details, see Section 14.18.2.4, "10BASE-T1S symbols", on page 837.

### 14.18.2.4 10BASE-T1S symbols

Symbol lists are protocol-specific.

A 10BASE-T1S symbol file contains two values for each Ethernet header:

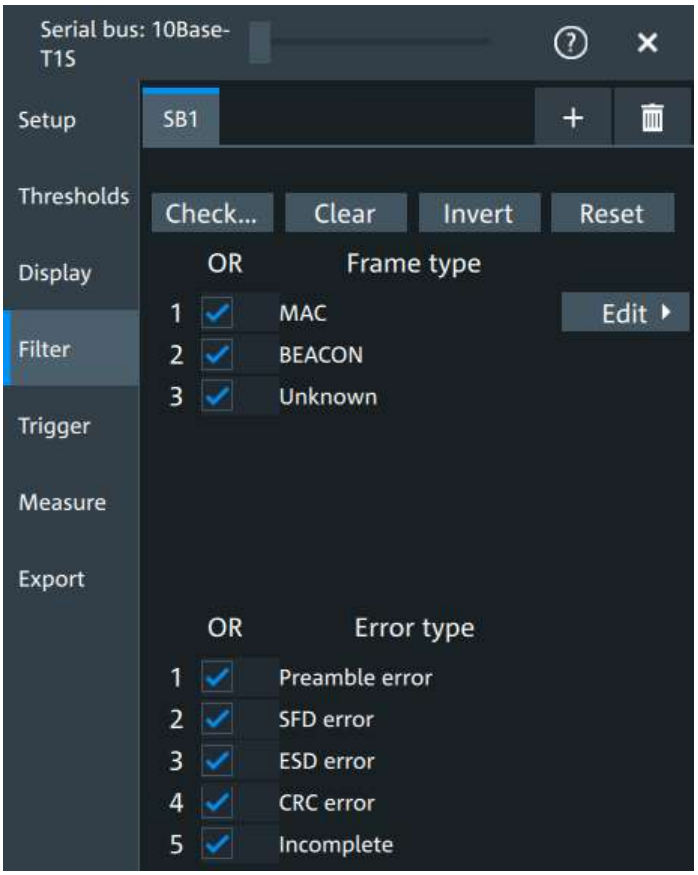
- Ethernet Header display
- Symbolic label: name of the address, specifying its function in the bus network.

```
############################################################
# Supported MAC Address Format
# xx:xx:xx  standard 24 bit manufactory header
# xx:xx:xx:xx:xx:xx/yy  support other length headers
# yy should be the header length in decimal
# yy should be between 24 - 48
############################################################
@PROTOCOL_NAME = ethernet
00:00:0C     Cisco
0:01:13     Olympus
0:01:14     KandaTsu
0:04:07     TopconPo
00:0B:64     KiebackP
00:1B:C5:06:C0:00/36     LuxconSy
00:1B:C5:06:D0:00/36     TesElect
00:1B:C5:06:E0:00/36     TwoDimen
00:1B:C5:06:F0:00/36     LlcEmzio
00:1F:BE     Shenzhen
00:21:8F     Avantgar
0.015625     GoliathS
0.015636574     D-Link
0.015648148     BaodingG
00:50:C2:5F:60:00/36     Cambridg
00:50:C2:5F:70:00/36     Metrolog
00:50:C2:5F:80:00/36     GrupoEpe
18:42:2F     AlcatelL
0.781273148     RiavaNet
18:46:17     SamsungE
18:8E:D5     TpVision
18:E7:F4     Apple
40:D8:55:1C:80:00/36     SensataT
40:D8:55:1C:90:00/36     Andy-L
40:D8:55:1C:A0:00/36     RigelEng
40:D8:55:1C:B0:00/36     MgSRL
40:D8:55:1C:D0:00/36     YxlonInt
```

```
40:D8:55:1C:E0:00/36      PeterHub
40:D8:55:1C:F0:00/36      OmnikNew
40:D8:55:1D:00:00/36      WebeasyB
FC:F8:B7      TronteqE
FC:FA:F7      Shanghai
FC:FE:77      HitachiR
FF:FF:FF:FF:FF:FF/48      BroadCast
```

### 14.18.3 10BASE-T1S filter

Access: "Menu" > "Apps" > "Protocol" tab > "10BASE-T1S" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
SBUS<sb>:TNOS:FILTer:CHKall on page 1923
SBUS<sb>:TNOS:FILTer:CLR on page 1923
SBUS<sb>:TNOS:FILTer:INVert on page 1924
SBUS<sb>:TNOS:FILTer:RST on page 1923

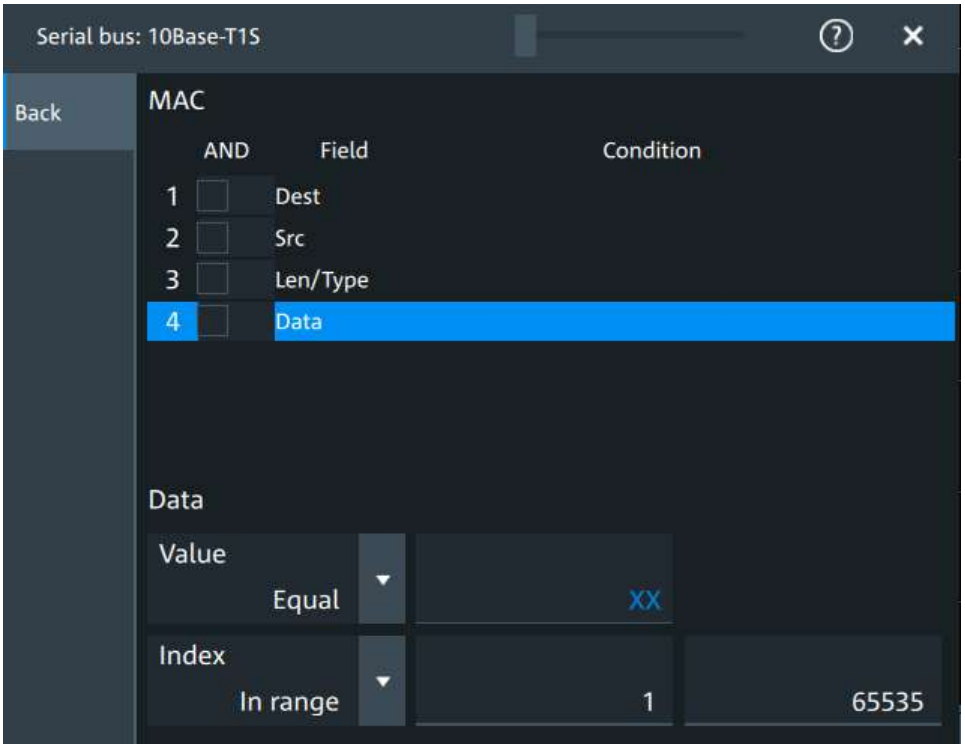
**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

Available frames are "MAC", "BEACON" and "Unknown".

Remote command:
SBUS<sb>:TNOS:FILTer:FRENable on page 1927
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:ENABle on page 1927

**Edit**
Opens a dialog to define the details of the selected frame.




"Field" Enables the field type that you want to filter on for the selected frame. The available fields are "Dest", "Src", "Len/Type" and "Data".

Remote command:
SBUS<sb>:TNOS:FILTer:FIENable on page 1926
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:ENABle on page 1926

"Condition" Displays the value condition for the selected field.

Remote command:
SBUS<sb>:TNOS:FILTer:BIT on page 1924
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:BIT on page 1924

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:TNOS:FILTer:DMAX on page 1924
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1924
SBUS<sb>:TNOS:FILTer:DMIN on page 1925
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1925
SBUS<sb>:TNOS:FILTer:DOPerator on page 1925
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1925

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:TNOS:FILTer:IMAX on page 1926
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1926
SBUS<sb>:TNOS:FILTer:IMIN on page 1926
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1926
SBUS<sb>:TNOS:FILTer:IOPerator on page 1927
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1927

**Error type**
Enables filtering on the selected error type.

Remote command:
SBUS<sb>:TNOS:FILTer:ERENable on page 1925
SBUS<sb>:TNOS:FILTer:ERRor<n>:ENABle on page 1925

### 14.18.4 PPIFormat.PPIFrame.Name10BASE-T1S format configuration

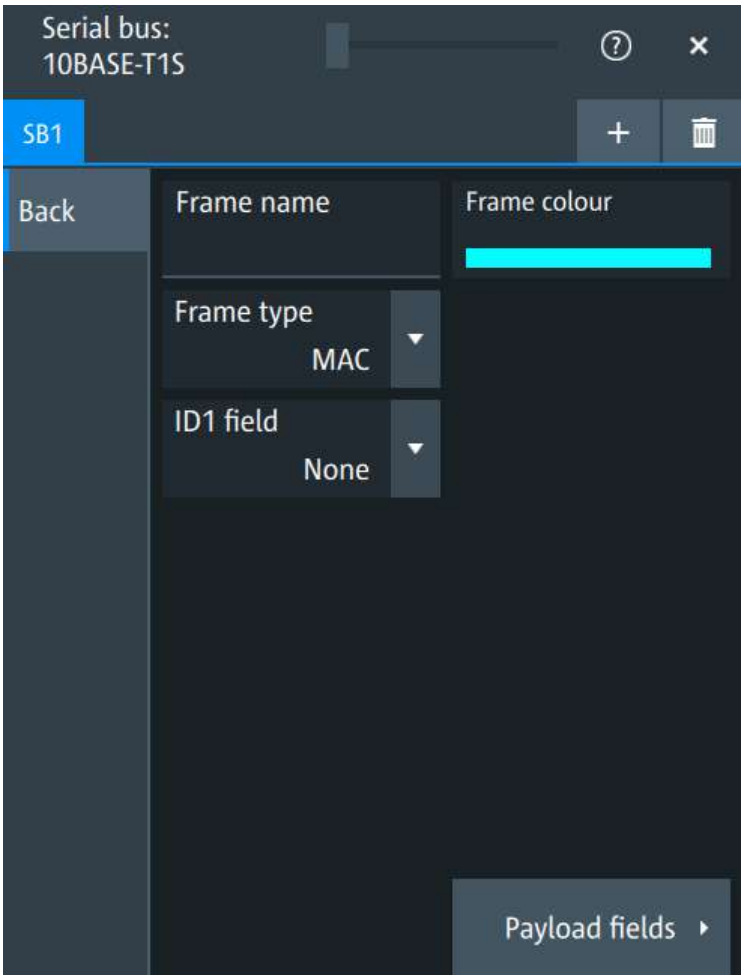
Access: "Menu" > "Apps" > "Protocol" tab > "10BASE-T1S" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
SBUS<sb>:TNOS:FORMat:SELect on page 1928

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:TNOS:FORMat:ADDFrame on page 1929

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:TNOS:FORMat:CLR on page 1929
SBUS<sb>:TNOS:FORMat:FCOunt? on page 1930

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:TNOS:FORMat:LOAD on page 1929

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:TNOS:FORMat:SAVE on page 1929

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:NAME on page 1930

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:COLor on page 1931

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:TYPE on page 1931

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID1Field on page 1931
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID2Field on page 1931
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID3Field on page 1931

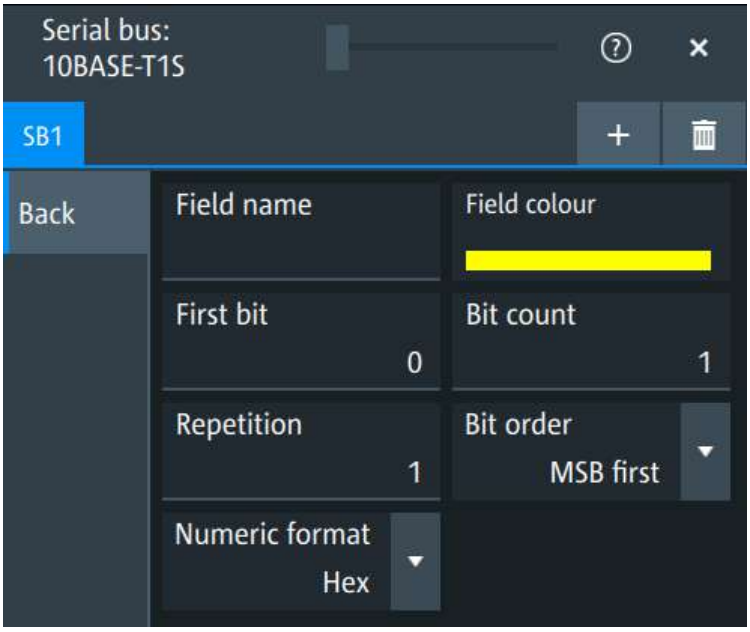
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID1Value on page 1932
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID2Value on page 1932
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID3Value on page 1932

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ADDField on page 1930

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1932

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1932

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1933

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.
- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  - Bit count = 2: truncated. Result: =00
  - Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  - Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  - Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1933

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
`SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:REPetition` on page 1933

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
`SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:BITorder` on page 1934

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:
- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.

- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  – "=1HL111000": valid (read as binary)
  – "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  – "=1010": valid (read as HEX, with a total of 16 bits)
  – "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)
  – "=5X12": invalid
  – "=1H33": invalid

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1934

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1935

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1935

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1936

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1936

**Parameter A,B**
Defines the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1936
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1936

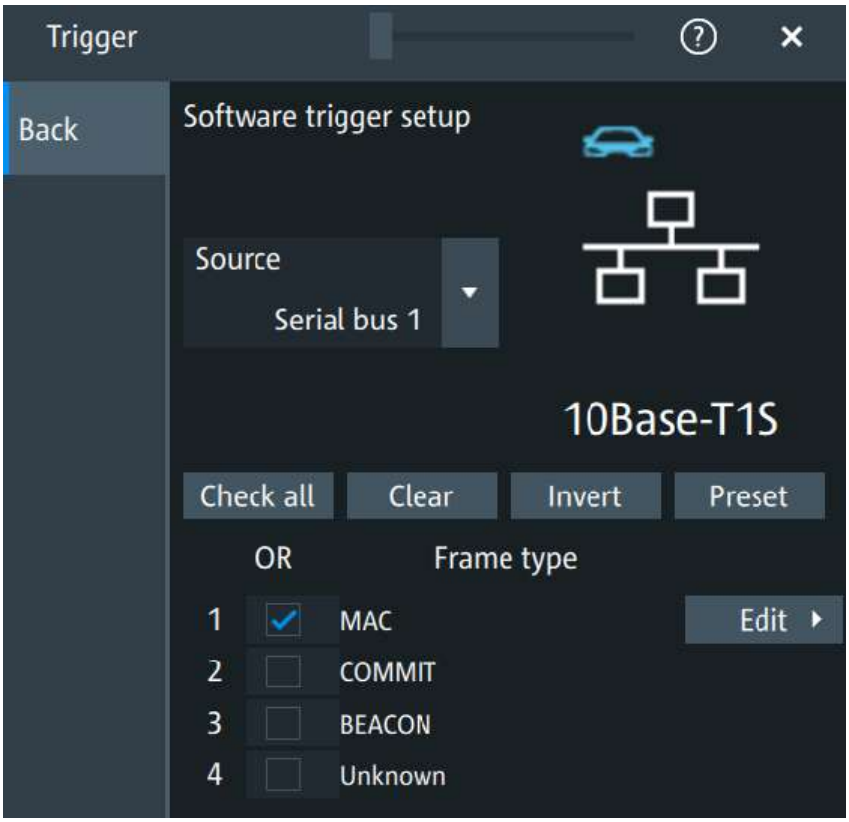
## 14.18.5 10BASE-T1S software trigger

### 14.18.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.18.5.2 Software trigger settings

Access: "Menu" > "Apps" > "Protocol" tab > "10BASE-T1S" > "Trigger" tab > "Setup Software Trigger"




In the "Software Trigger" tab, you can define the settings to trigger on the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
TRIGger:SBSW:TNOS:CHKall on page 1938
TRIGger:SBSW:TNOS:CLR on page 1938
TRIGger:SBSW:TNOS:INVert on page 1938
TRIGger:SBSW:TNOS:RST on page 1938

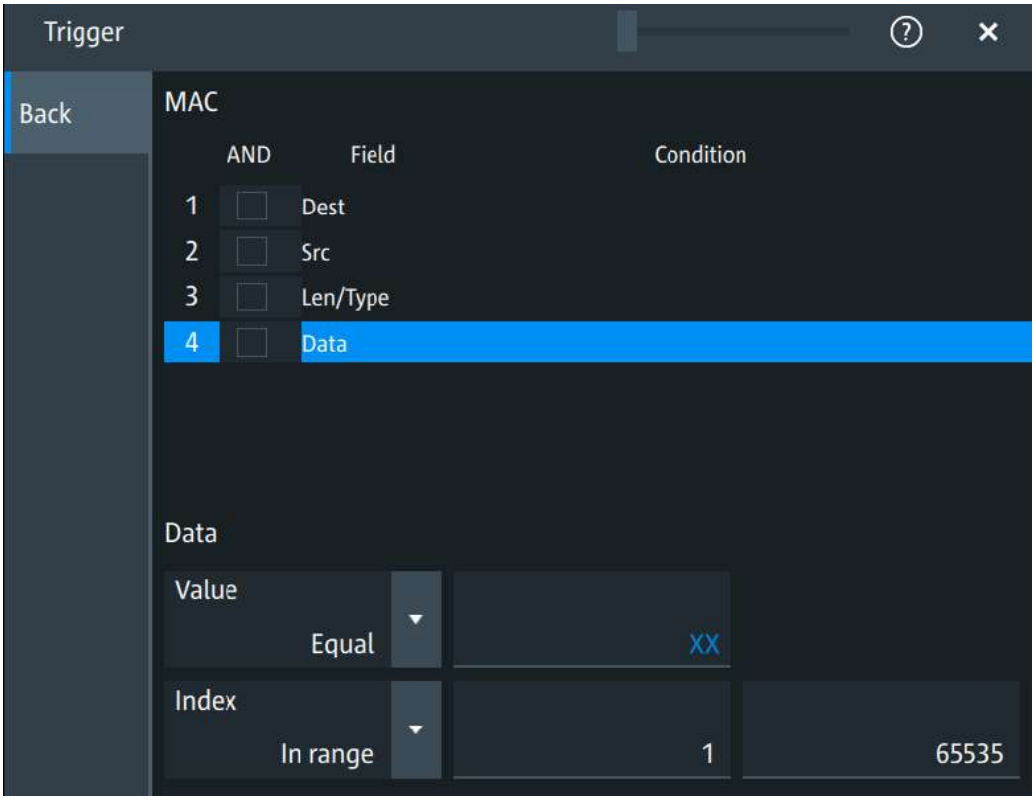
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: "MAC", "COMMIT", "BEACON", "Unknown".

Remote command:
TRIGger:SBSW:TNOS:FRENable on page 1938
TRIGger:SBSW:TNOS:FRAMe<fr>:ENABle on page 1938

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.



"Field" Enables the field type that you want to trigger on for the selected frame.
The following fields are available: "Dest", "Src", "Len/Type", "Data".

Remote command:
TRIGger:SBSW:TNOS:FIENable on page 1940
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:ENABle on page 1940

"Condition" Displays the value condition for the selected field.

Remote command:
TRIGger:SBSW:TNOS:BIT on page 1939
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:BIT on page 1939

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
TRIGger:SBSW:TNOS:DMAX on page 1939
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DMAX on page 1939
TRIGger:SBSW:TNOS:DMIN on page 1939
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DMIN on page 1939
TRIGger:SBSW:TNOS:DOPerator on page 1940
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DOPerator on page 1940

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:TNOS:IMAX on page 1941
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IMAX on page 1941
TRIGger:SBSW:TNOS:IMIN on page 1941
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IMIN on page 1941
TRIGger:SBSW:TNOS:IOPerator on page 1941
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IOPerator on page 1941

**Error type**
Enables triggering on the selected error type.

The following error types are available: "Preamble error", "SFD error", "ESD error", "CRC error".

Remote command:
TRIGger:SBSW:TNOS:ERENable on page 1942
TRIGger:SBSW:TNOS:ERRor<m>:ENABle on page 1942

### 14.18.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

## 14.18.7 10BASE-T1S decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

### Decode results table

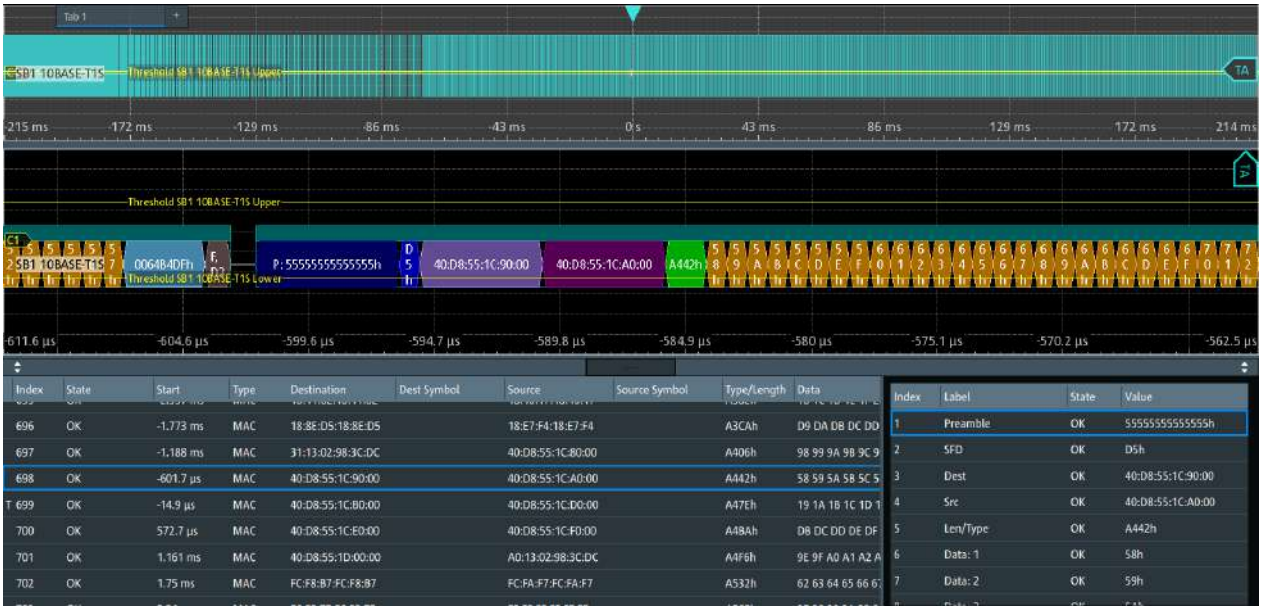



*Figure 14-53: Decoded 10BASE-T1S signal*

The decode results table contains information about all decoded frames.

*Table 14-32: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Frame count |
| State | Overall state of the frame: either OK or the relevant error condition (preamble, CRC or SFD) |
| Start | Start time of the frame |
| Type | Type of frame (e.g. Idle, MAC or data) |
| Destination | Destination address of the frame |
| Dest Symbol | Symbolic name of the destination defined in the symbol list. Available only, if "Show symbols" is enabled in the "Display" tab and a symbols list is loaded. |
| Source | Source address of the frame |
| Source Symbol | Symbolic name of the source defined in the symbol list. Available only, if "Show symbols" is enabled in the "Display" tab and a symbols list is loaded. |

| Column | Description |
|---|---|
| Type/Length | The sub-protocol (e.g. HTML, video, etc.) determines what meaning this field has. Since the content of this data area is not decoded, the interpretation of this field is ambivalent. It could either be the word type (specific for the sub-protocol) or the word length. |
| Data | Value of the data. Select the data format in the "Display" tab. |
| Bit rate | Value of the bit rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

***Table 14-33: Content of the frame details table***

| Column | Description |
|---|---|
| Index | Index of the decoded field |
| Label | Name of the field |
| State | State of the field (OK / not OK) |
| Value | Value of the field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.19.5, "Decode results", on page 1943.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

### Example 10BASE-T1S export file

```
Index,Start,Stop,State,Type,Destination,Dest Symbol,Source,Source Symbol,
  Type/Len,CRC,Bit rate
1,-0.214999,-0.214971,'OK','MAC','00:00:0C:12:14:65',,'00:01:13:AB:C1:9C',
  ,00E6h,00002507h,12500000
2,-0.21497,-0.214941,'EPRM','MAC','00:01:14:06:C0:00',,'00:04:07:C6:C0:00',
  ,0122h,00004A04h,12500000
3,-0.21494,-0.21491,'ESFD','MAC','00:13:02:98:3C:CC',,'00:0B:64:06:C0:12',
  ,015Eh,00006F01h,12500000
4,-0.214909,-0.214878,'EESD','MAC','00:1B:C5:06:C0:00',,'00:1B:C5:06:D0:00',
  ,019Ah,000093FEh,12500000
5,-0.214877,-0.214846,'ECRC','MAC','00:1B:C5:06:E0:00',,'00:1B:C5:06:F0:00',
  ,01D6h,0000B8FBh,12500000
```

```
Details frame 1
Index,State,Value
1,'OK',14h
2,'OK',15h
3,'OK',16h
4,'OK',17h
5,'OK',18h
6,'OK',19h
7,'OK',1Ah
8,'OK',1Bh

Details frame 2
Index,State,Value
1,'OK',1Ch
2,'OK',1Dh
3,'OK',1Eh
4,'OK',1Fh
5,'OK',20h
6,'OK',21h
7,'OK',22h
8,'OK',23h
9,'OK',24h
```

## 14.19 USB 1.1 (option R&S MXO3-K570)

R&S MXO3-K570 is a firmware option that enables the MXO 3 to analyze Universal Serial Bus (USB) protocols, by triggering and decoding them. The option is compatible with the standards USB 1.0, USB 1.1 and USB 2.0, but not with USB HSIC. It supports the data rates "Low Speed" (1.5 Mbit/s) and "Full Speed" (12 Mbit/s), both in USB 1.0 and USB 1.1, and "High Speed" (480 Mbit/s in USB 2.0).

For USB 2.0, refer to Section 14.20, "USB 2.0 (option R&S MXO3-K570)", on page 874.

- USB protocol basics....851
- USB 1.1 configuration....856
- USB 1.1 filter....860
- USB1.1 format configuration....863
- USB 1.1 trigger....869
- Measure....872
- USB 1.1 decode results....872

### 14.19.1 USB protocol basics

The USB protocol was developed, starting in 1996, by the nonprofit organization USB Implementers Forum, Inc. (USB-IF), formed by Compaq, Hewlett-Packard, Intel, Lucent Technologies, Microsoft, NEC, and Philips. The purpose was to provide a common "plug-and-play" solution to replace a multitude of interfaces for the communication

between computers and devices. It allows even unskilled users to connect many devices to a PC easily. USB was originally used for devices that feature low signaling rates (up to 1.5 Mbit/s), especially human interface devices like mouse, joystick or keyboard. With the release of USB 2.0 in April 2000, devices such as printers, cameras and mass storage media were enabled to exchange data at faster rates (up to 480 Mbit/s). Also, the high-speed protocol HSIC (High Speed Inter Chip) is used for the communication between on-board devices.

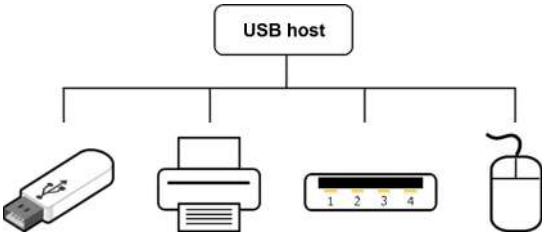



***Figure 14-54: USB Topology: mass storage device, printer, USB hub, and mouse (human interface device) connected to a USB host***

More information, including the USB specification, is available online within the web domain http://usb.org.

This section provides an overview of protocol characteristics, data transfer, packet structure, address and endpoint formats of USB and trigger possibilities.

### USB characteristics

The main characteristics of USB are:

- Four-wire design: USB requires a shielded cable containing four wires. Two of them, called D+ and D-, form a twisted pair (for low speed, they can be non-twisted). These data lines transmit differential data signals and single-ended signal states, both referenced to a third wire: the GND or ground. The fourth wire is called VBUS (voltage bus). It carries a nominal 5 V supply, which can be used to power a device.
- Host-to-device communication: in USBs "speak-when-spoken-to" protocol, communication is always initiated by the host. Consequently, there is no direct communication between USB devices, apart from a few exceptions.
- The addressing scheme can distinguish a maximum of 127 connected devices, because a packet's address field length is limited to 7 bits. USB devices have up to 16 OUT endpoints (from host to device) and up to 16 IN endpoints (from device to host).
- USB transactions consist of two or three packets: token, data, and typically handshake
- Packet type: a packet identifier (PID) is sent as a first byte within the packet and specifies the different packet types.
- NRZI (Non Return to Zero Inverted): a zero (0) is encoded as a transition of the physical level, whereas a one (1) has no transition, thus it is represented by a steady level.

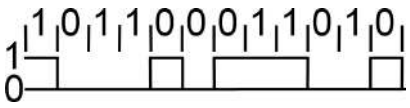



*Figure 14-55: Example of an NRZI sequence*

- Bit stuffing: a zero (0) is inserted after every 6 consecutive ones (111111). This insertion ensures sufficient transitions to keep the phase-locked loop (PLL) synchronized and locked onto the data stream.
- Little Endian scheme: both multiple bits and multiple bytes are transmitted with the least significant bit/byte (LSB) sent first, while the most significant bit/byte (MSB) is sent last.
- HSIC (high-speed inter-chip): an industry standard for USB chip-to-chip interconnection with a 2-signal (strobe, data) source synchronous serial interface, using 240 MHz DDR signaling to provide only high-speed (480 Mbit/s) data rate.

*Table 14-34: Simplified symbolic representation from the USB standard*

| Bus State | Protocol | Levels |
|---|---|---|
| Differential **"1"** | | D+ High, D- Low |
| Differential **"0"** | | D+ Low, D- High |
| Data **"J"** State | Low speed | D+ Low, D- High (differential **"0"**) |
| | Full speed | D+ High, D- Low (differential **"1"**) |
| Data **"K"** State | Low speed | D+ High, D- Low (differential **"1"**) |
| | Full speed | D+ Low, D- High (differential **"0"**) |

In the example in Table 14-34, "High" speed can be assumed to be like "Full" speed. For a complete overview of bus states and for an electrical definition of "High" and "Low" speed, refer to section 7 of the USB specification.

### Data transfer

In contrast to a conventional bus, USB is more like a network protocol, using target addresses and endpoints. However, USB features a primary on the bus, called the host. It transmits packets of data to all devices or hubs connected to the host (or connected to a device or hub, which is in turn connected to the host). All connected devices receive each data packet, but disregard it unless it carries the correct address. In reply, only the addressed device (one at a time) can send data upstream to the host.

The USB architecture supports four different kinds of data transfer: control, bulk, interrupt or isochronous transfer. For example, a "bulk OUT" transfer (from host to device) can look like this:

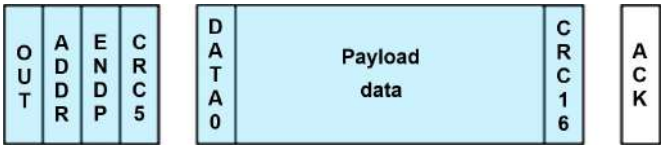



*Figure 14-56: Example of a bulk OUT transfer. Blue: host speaks, white: device speaks*

### Packet structure

All packets must start with a **SYNC** field, also called **SOP** (start of packet), which indicates data transmission. It consists of "KJ" pairs, followed by one "KK". At low speed and full speed (USB 1.x), it is 8 bits long, encoded as "KJKJKJKK". At high speed (USB 2.0 and HSIC), it is up to 32 bits long, encoded as "KJKJKJKJKJ...KK".

The SYNC field is used to synchronize the clock of the receiver with the clock of the transmitter. The final 2 bits ("KK") indicate where the PID field starts.

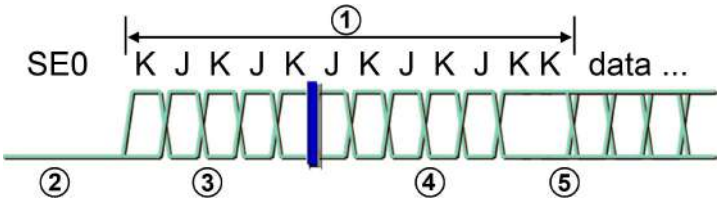



***Figure 14-57: Start of a high-speed packet: after an SE0, the packet starts with a SYNC field (or SOP), followed by the data packet***

SE0 = Single-ended zero, both D+ and D- wire are on low level
1 = High-speed SYNC field (or SOP), length up to 32 bits
2 = Receiver-squelched differential envelope, below 100 mV
3 = Differential envelope exceeds 150 mV, reception enabled with 4 symbol times
4 = Clock recovery must lock in time to detect the end of SYNC
5 = End of SYNC detected, receiver begins data recovery

After the SYNC (or SOP), USB data packets consist of (in this order):

- **PID** (packet identifier): specifies the type of packet. Examples are:
  - Token packets (IN, OUT, SETUP) are always the first packet in a USB transaction, containing the address of the device and its endpoint.
  - Token SOF packets (start of frame) are there to schedule a data transfer.
  - Handshake packets provide the status of the transaction.
  - Data packets are dedicated for the data transfer payload.
  - Various special packets provide special characteristics: for instance, SPLIT packets are used to communicate with low or full speed devices on a high-speed link.

  The PID's length is 8 bit, beginning (LSB) with the fields PID0, PID1, PID2 and PID3. Those 4 bits are repeated in PID4, PID5, PID6 and PID7, but in complementary form (for integrity checking). The PID is the only part of a packet not to be covered by the CRC (see below). Packets with invalid or non-supported PID fields are rejected. For valid PIDs, see Table 14-35.
- **ADDR** (address): contains the USB address of the target device. The length of this field is 7 bits, resulting in addresses 0 through 127 (with 0 reserved as the default address after reset).
- **ENDP** (endpoint): number of the endpoint. The length of this field is 4 bits, resulting in a maximum of 16 different endpoints.
- **Frame number**: this field with a length of 11 bits specifies the current frame number 0 to 2047. It is only sent in SOF packets.
- **Data field** (or "payload data"): the length of this field is not fixed, it can contain between 0 bytes and 1024 bytes.

- **CRC** (cyclic redundancy check): Finally, the CRC field ensures that the received data is correct. If an error is detected in the CRC calculation, the packet is rejected. Packets (in this case IN, OUT, SOF, SETUP, PING, SPLIT) are followed by a CRC5 (which is 5 bits long) over ADDR and ENDP (or the frame number field in the SOF packet). Data packets (DATA0, DATA1, MDATA, DATA2) are followed by a CRC16 (which is 16 bits long) over the data field. No CRC is used for handshake packets.
- **EOP** (end of package): the devices determine the length of the data field with the EOP signal, which indicates the end of a packet. In low and full speed, it consists of two SE0 bits, followed by a J state. For high speed, the current state is inverted and kept for 7-bit times, leading to a bit-stuff error that is used as an EOP indicator. This state corresponds to the bit sequence 01111111 ahead of NRZI. For the SOF token, the bit sequence 01111111111111111111111111111111111111111111 is transmitted (five NRZI bytes), which is used to recognize disconnects (since high-speed devices normally do not apply voltage to D+/D-). If no device is present, a higher voltage is measured. After that, the idle state (SE0) is occupied.

***Table 14-35: Valid PIDs for different packet types***

| Packet type | Name | PID value | Meaning |
|---|---|---|---|
| Token | OUT | 0xE1 | Starts data transfer towards a device |
| | IN | 0x69 | Starts data transfer towards the host |
| | SOF | 0xA5 | Indicates start of frame |
| | SETUP | 0x2D | Starts a setup transfer and sends information on this start to the device |
| Data | DATA0 | 0xC3 | Data packet with data-toggle bit 0 |
| | DATA1 | 0x4B | Data packet with data-toggle bit 1 |
| | DATA2 | 0x87 | Data packet for high speed IN isochronous transfers that require high bandwidth |
| | MDATA | 0x0F | Data packet for high-speed OUT isochronous transfers that require high bandwidth |
| Handshake | ACK | 0xD2 | Acknowledgment of a packet received without error |
| | NAK | 0x5A | Data not accepted, typically equivalent with some type of EAGAIN, meaning that the data needs to be resent later |
| | STALL | 0x1E | A severe error has occurred, the target endpoint cannot be addressed until it is explicitly cleared again |
| | NYET | 0x96 | Only used in high-speed transfers, meaning ACK, but in the next interval no data can be received. Therefore the host must apply a PING first. |
| Special | PRE | 0x3C | Starts a low speed transfer via a full speed bus |
| | ERR | 0x3C | Indicates an error in an SPLIT transaction (using the same PID as PRE, however, these indications cannot be mistaken for each other) |
| | SPLIT | 0x78 | Starts an SPLIT transaction (thus: a low speed or full speed transfer via a high-speed bus) |
| | PING | 0xB4 | Used for monitoring high-speed data flow |

- Token packets (IN, OUT, SETUP) and PING packets have the following format:

| SYNC | PID | ADDR | ENDP | CRC5 | EOP |
|---|---|---|---|---|---|

- Start of frame packets (SOF) have the following format:

| SYNC | PID | Frame number | CRC5 | EOP |
|---|---|---|---|---|

- Data packets have the following format:

| SYNC | PID | Data | CRC16 | EOP |
|---|---|---|---|---|

- Handshake packets have the following format:

| SYNC | PID | EOP |
|---|---|---|

- SPLIT packets have the following format:

| SYNC | PID | ADDR | SC | PORT | S | E | ET | CRC5 | EOP |
|---|---|---|---|---|---|---|---|---|---|

**Trigger possibilities**

Signals on the input channels can be triggered. The following trigger types are available:

- ANY Packet - Packet Sync: Triggering on the first rising slope after transmission of the packet Sync. Various lengths according to standard (in USB 2.0: 32 bit)
- Any token, OUT, IN, SOF, SETUP, AND-ing with user-defined PID check, address, endpoint, CRC5: For OUT, IN, SETUP, the endpoint and CRC5 follow from the bit order, therefore such patterns can be recognized.
- Data Selection: DATA0, DATA1, DATA2, MDATA, (for USB 1.x only: AND-ing with user-defined PID check, payload and CRC values)
- Handshake Packet Setup: Triggering on handshake packet, trigger with specific settings: ACK, NAK, NYET, STALL or ERR handshake packet
- Protocol Error: Triggering on PID/check error, CRC5 error, CRC16 (for USB 1.x only), frame length error (for USB 1.x only)
- Bus Event: Triggering on reset, resume, or suspend.

## 14.19.2 USB 1.1 configuration

If you need information on how to get started with configuring the USB 1.1 setup, see Section 14.19.2.2, "Configuring USB 1.1 signals", on page 858. Otherwise proceed with the configuration settings.

### 14.19.2.1 USB 1.1 configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "USB 1.1" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.

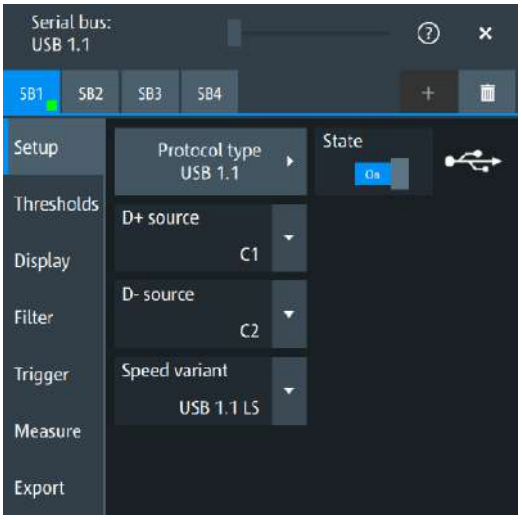



*Figure 14-58: Serial bus protocol configuration dialog for USB 1.1 LS or FS*

See also: Section 14.1.1, "Setup - general settings", on page 489.

**Protocol type**
Selects the protocol type to be decoded.

Remote command:
`SBUS<sb>:TYPE` on page 1362

**State**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
`SBUS<sb>[:STATe]` on page 1362

**D+ source, D- source**
Defines the source settings for the D+ and the D- data signals for USB 1.1 low speed (LS) or full speed (FS) protocols.

Usually, the source is one of the analog channels. Reference and math waveforms are available as sources if the trigger source is one of the analog channels but not the serial bus.

Remote command:
`SBUS<sb>:USBone:DPLus:SOURce` on page 1951
`SBUS<sb>:USBone:DMINus:SOURce` on page 1952

**Speed variant**
Selects the low or full speed variant of USB:

"USB 1.1 LS"
Selects the 1.5 Mbit/s low speed (LS) protocol variant of USB 1.1.

"USB 1.1 FS"
Selects the 12 Mbit/s full speed (FS) protocol variant of USB 1.1.

**Note:** Instead of selecting the speed variant in this "Setup" dialog, you can select it in the protocol selection table:




To access the protocol selection table, select "Menu" > "Apps" > "Protocol", or use the "Protocol type" button in the "Setup" dialog.

Remote command:
`SBUS<sb>:USBone:MODE` on page 1952

### 14.19.2.2 Configuring USB 1.1 signals

For configuration, assign the lines to the input channels and define the active states and the logical thresholds.

For details on configuration settings, see Section 14.19.2.1, "USB 1.1 configuration settings", on page 856.

**Serial Bus Setup**

1. Tap the "+ Bus" activator in the bottom right of the screen.
2. Select the tab of the bus that you want to set up, for example "SB1".
3. Tap the "Protocol type" button. Select "Protocol type" = "USB 1.1 LS" or "USB 1.1 FS".
4. Select the waveform for the "D+ source" and "D- source" lines.
5. In the "Thresholds" tab, specify the logical thresholds: "D+ threshold" and "D- threshold".
   Specify also hysteresis settings: "D+ hysteresis" and "D- hysteresis".
6. In the "Display" tab, enable "Show decode table".

### 14.19.2.3 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "USB 1.1" > "Thresholds" tab

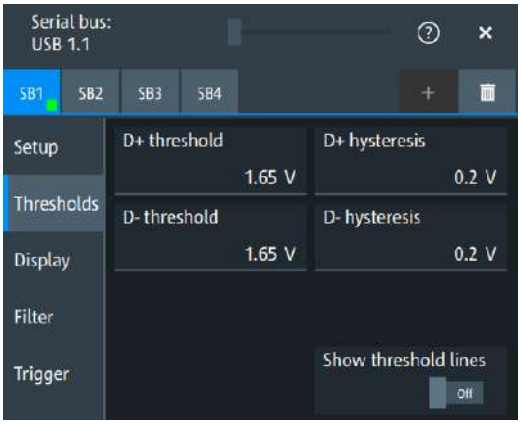



*Figure 14-59: Serial bus threshold configuration dialog for USB 1.1*

**Threshold and hysteresis settings**
Specifies the following values:

**D+ threshold ← Threshold and hysteresis settings**
Sets the threshold value for the digitization of the signal on the D+ data line.

For transformation of high or low levels into J or K states of the bus, refer to Table 14-34. The J and K states are transformed into bits according to the rules of non-return-to-zero inverted (NRZI) encoding.

Remote command:
`SBUS<sb>:USBone:DPLus:THReshold` on page 1952

**D- threshold ← Threshold and hysteresis settings**
Sets the threshold value for the digitization of the signal on the D- data line.

See also "D+ threshold" on page 859.

Remote command:
`SBUS<sb>:USBone:DMINus:THReshold` on page 1953

**D+ hysteresis ← Threshold and hysteresis settings**
Sets the hysteresis value for the D+ data line.

Remote command:
`SBUS<sb>:USBone:DPLus:HYSTeresis` on page 1953

**D- hysteresis ← Threshold and hysteresis settings**
Sets the hysteresis value for the D- data line.

Remote command:
`SBUS<sb>:USBone:DMINus:HYSTeresis` on page 1953

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
SBUS<sb>:THReshold on page 1364

#### 14.19.2.4 Display settings

Access: "Menu" > "Apps" > "Protocol" tab > "USB 1.1" > "Display" tab

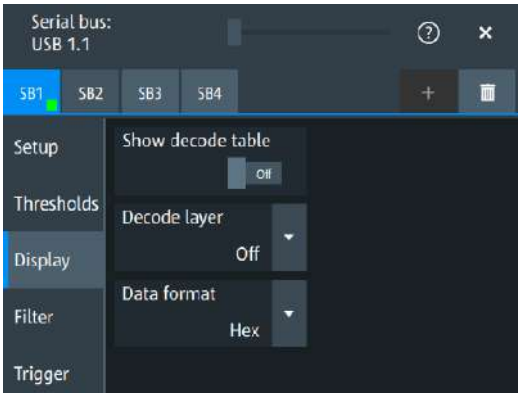



*Figure 14-60: Serial bus display configuration dialog for USB 1.1*

For details about the display settings, see Section 14.1.3.1, "Display settings", on page 492.

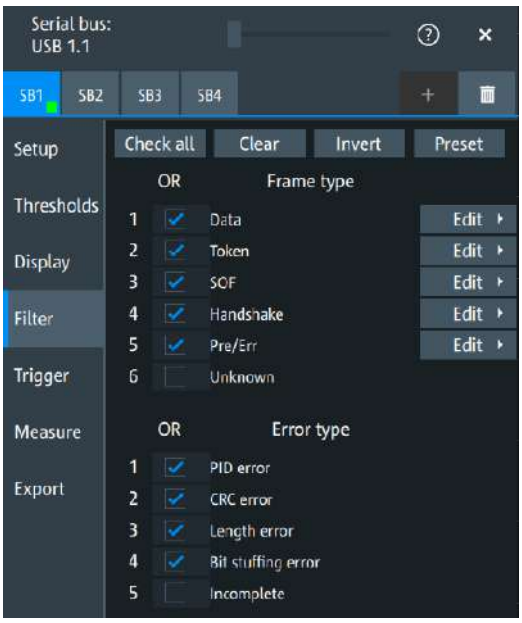
**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

| | |
|---|---|
| "Off" | Disables the display of the decode layer. |
| "Edges" | Enables the display of all edges. |
| "States" | "Enables the display of J and K states according to Table 14-34 in Section 14.19.1, "USB protocol basics", on page 851. |
| "Bits" | Enables the display of the raw, binary encoded symbols, consisting of the states "0" and "1". |
| "Words" | Enables the display of words, which are the numerical values of the individual fields that compose each packet, like PID, ADDR, DATA or CRC. |

### 14.19.3 USB 1.1 filter

Access: "Menu" > "Apps" > "Protocol" tab > "USB 1.1" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
`SBUS<sb>:USBone:FILTer:CHKall` on page 1955
`SBUS<sb>:USBone:FILTer:CLR` on page 1955
`SBUS<sb>:USBone:FILTer:INVert` on page 1955
`SBUS<sb>:USBone:FILTer:RST` on page 1955

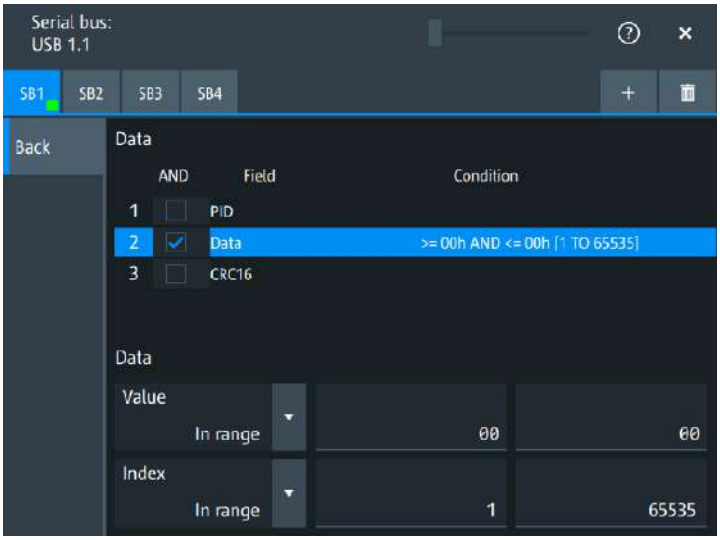
**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

Available frames are "Data", "Token", "SOF", "Handshake", "Pre/Err" and "Unknown".

Remote command:
`SBUS<sb>:USBone:FILTer:FRENable` on page 1955
`SBUS<sb>:USBone:FILTer:FRAMe<fr>:ENABle` on page 1955

**Edit**
Opens a dialog to define the details of the selected frame.




"Field" Enables the field type that you want to filter on for the selected frame. The available fields are "PID", "Data", "CRC5", "CRC16", "Addr", "Endp" and "Frame No.".

Remote command:
`SBUS<sb>:USBone:FILTer:FIENable` on page 1959
`SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:ENABle` on page 1959

"Condition" Displays the value condition for the selected field.

Remote command:
`SBUS<sb>:USBone:FILTer:BIT` on page 1959
`SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:BIT` on page 1959

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`SBUS<sb>:USBone:FILTer:DMAX` on page 1956
`SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DMAX` on page 1956
`SBUS<sb>:USBone:FILTer:DMIN` on page 1956
`SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DMIN` on page 1956
`SBUS<sb>:USBone:FILTer:DOPerator` on page 1957
`SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DOPerator` on page 1957

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:USBone:FILTer:IMAX on page 1957
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1957
SBUS<sb>:USBone:FILTer:IMIN on page 1958
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1958
SBUS<sb>:USBone:FILTer:IOPerator on page 1958
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1958

**Error type**
Enables filtering on the selected error type.

The available error types are "PID error" (packet identifier), "CRC error" (cyclic redundancy check), "Length error", "Bit stuffing error" and "Incomplete".

Remote command:
SBUS<sb>:USBone:FILTer:ERENable on page 1957
SBUS<sb>:USBone:FILTer:ERRor<n>:ENABle on page 1957

### 14.19.4 USB1.1 format configuration

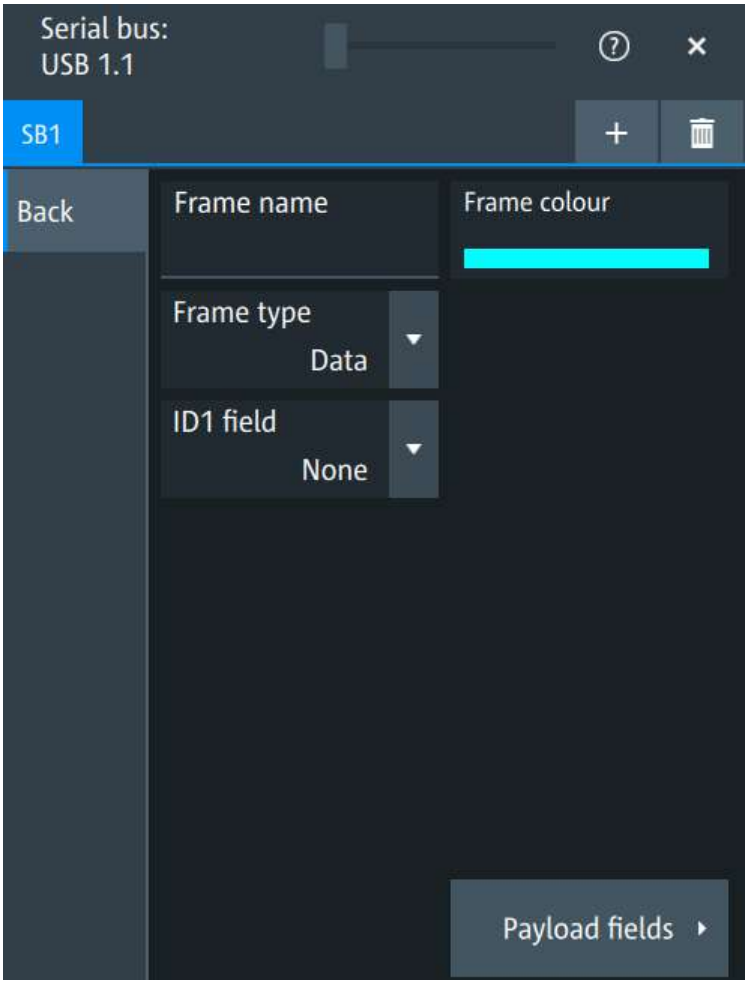
Access: "Menu" > "Apps" > "Protocol" tab > "USB 1.1" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
SBUS<sb>:USBone:FORMat:SELect on page 1960

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
SBUS<sb>:USBone:FORMat:ADDFrame on page 1960

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:USBone:FORMat:CLR on page 1961
SBUS<sb>:USBone:FORMat:FCOunt? on page 1961

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:USBone:FORMat:LOAD on page 1961

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:USBone:FORMat:SAVE on page 1961

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:NAME on page 1962

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:COLor on page 1962

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:TYPE on page 1963

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID1Field on page 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID2Field on page 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID3Field on page 1963

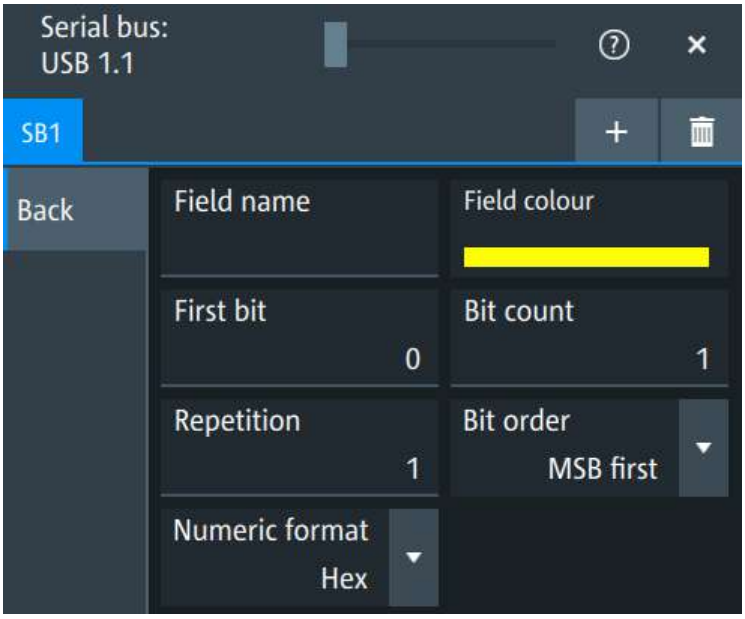
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID1Value on page 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID2Value on page 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID3Value on page 1963

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ADDField on page 1962

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1964

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1964

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1964

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.

- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  – Bit count = 2: truncated. Result: =00
  – Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  – Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  – Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
`SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:BITCount` on page 1965

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
`SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:REPetition` on page 1965

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
`SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:BITorder` on page 1965

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.
- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:
- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  – "=1HL111000": valid (read as binary)
  – "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  – "=1010": valid (read as HEX, with a total of 16 bits)
  – "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)

– "=5X12": invalid
– "=1H33": invalid

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FORMat on page 1966

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ADDValue on page 1967

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue on page 1967

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel on page 1967

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:UNIT on page 1968

**Parameter A,B**
Defines the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:PARA on page 1968
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:PARB on page 1968

## 14.19.5 USB 1.1 trigger

### 14.19.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.19.5.2 Software trigger setup

Access: "Menu" > "Apps" > "Protocol" tab > "USB 1.1" > "Trigger" > "Setup Software Trigger" tab

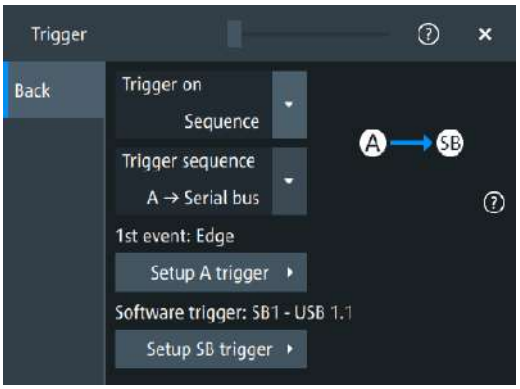



*Figure 14-61: Serial bus Trigger setup dialog for USB 1.1*

Make sure that:

- In the "Trigger" tab (Figure 14-58):
  - The correct protocol type and speed variant are selected.
  - The correct channels are assigned to the corresponding data source of the serial bus.
- In the "Trigger" tab (Figure 14-61):
  - The "Trigger on" = "Sequence"
  - The "Trigger sequence" = "A → Serial bus"
  - The 1st event is set according to your requirements.
- In the "Setup SB Trigger" tab (Figure 14-62):
  - The trigger "Source" is set to the correct serial bus, for example "Serial bus 1".

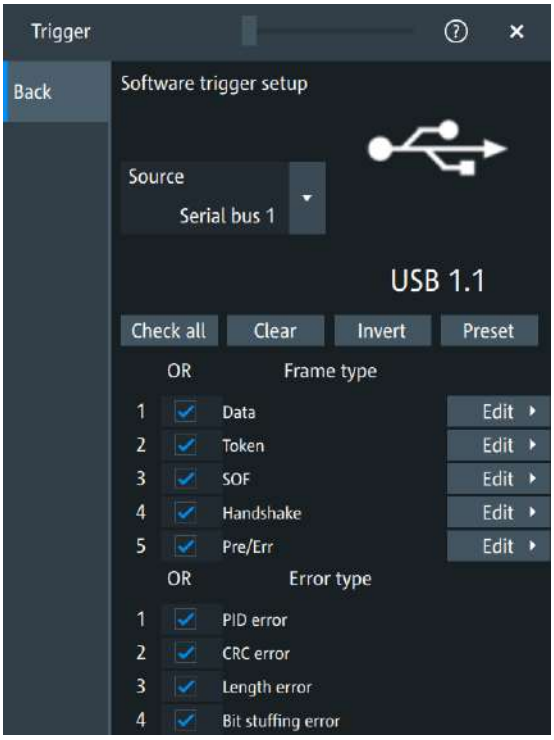



*Figure 14-62: Serial bus Software trigger setup dialog for USB 1.1*

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:USBone:CHKall` on page 1969
`TRIGger:SBSW:USBone:CLR` on page 1970
`TRIGger:SBSW:USBone:INVert` on page 1970
`TRIGger:SBSW:USBone:RST` on page 1970

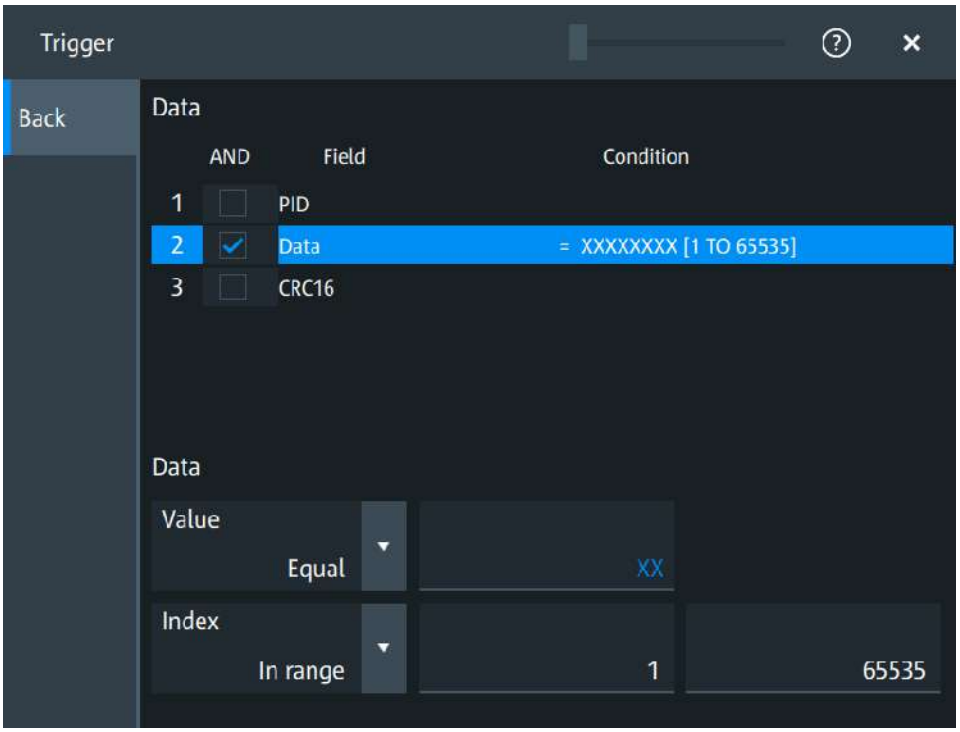
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: "Data", "Token", "SOF", "Handshake" and "Pre/Err".

Remote command:
`TRIGger:SBSW:USBone:FRENable` on page 1970
`TRIGger:SBSW:USBone:FRAMe<fr>:ENABle` on page 1970

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.




"Field" Enables the field type that you want to trigger on for the selected frame.
The available fields are "PID", "Data", "CRC5", "CRC16", "Addr", "Endp" and "Frame No.".

Remote command:
`TRIGger:SBSW:USBone:FIENable` on page 1972
`TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:ENABle` on page 1972

"Condition" Displays the value condition for the selected field.

Remote command:
`TRIGger:SBSW:USBone:BIT` on page 1970
`TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:BIT` on page 1970

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
TRIGger:SBSW:USBone:DMAX on page 1971
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DMAX on page 1971
TRIGger:SBSW:USBone:DMIN on page 1971
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DMIN on page 1971
TRIGger:SBSW:USBone:DOPerator on page 1972
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DOPerator on page 1972

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
TRIGger:SBSW:USBone:IMAX on page 1972
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IMAX on page 1972
TRIGger:SBSW:USBone:IMIN on page 1973
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IMIN on page 1973
TRIGger:SBSW:USBone:IOPerator on page 1973
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IOPerator on page 1973

**Error type**
Enables triggering on the selected error type.

The available error types are "PID error" (packet identifier), "CRC error" (cyclic redundancy check), "Length error" and "Bit stuffing error".

Remote command:
TRIGger:SBSW:USBone:ERENable on page 1973
TRIGger:SBSW:USBone:ERRor<m>:ENABle on page 1973

### 14.19.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.19.7 USB 1.1 decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

### Decode results table

The "Decode results" table contains information about all decoded frames.

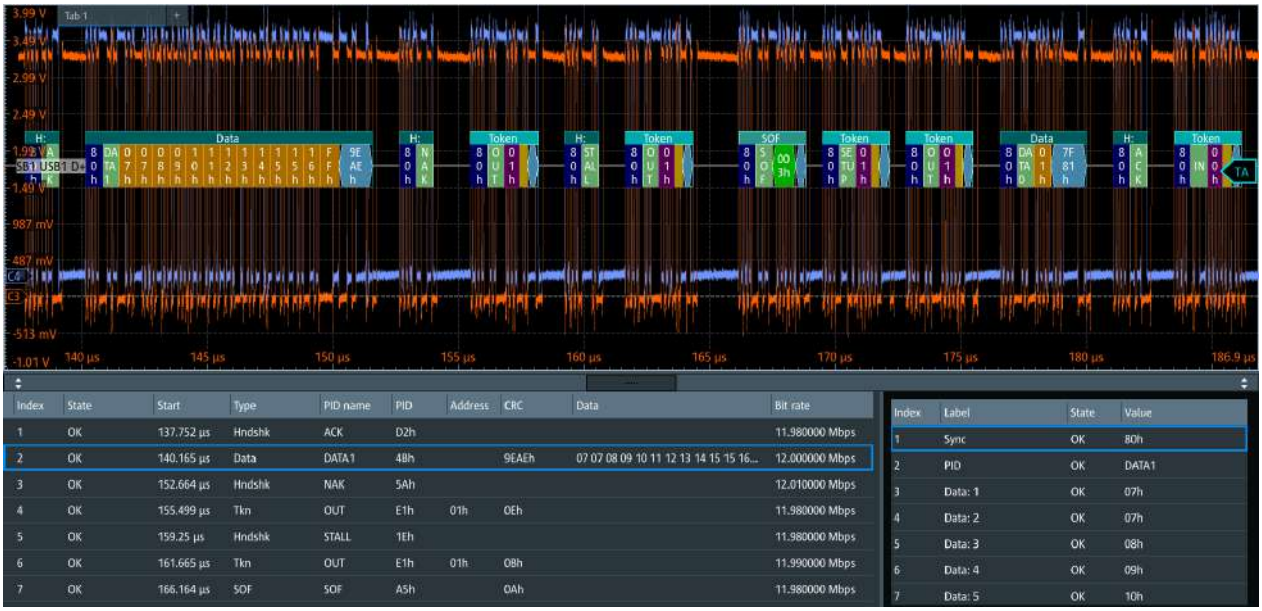



*Figure 14-63: Example of a decoded USB 1.1 signal*

The decode results table contains information about all decoded frames.

*Table 14-36: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Index of the decoded frame |
| State | Overall state of the frame |
| Start | Times of frame start |
| Type | Type of the frame |
| PID name | Name of the packet identifier |
| PID | Value of the packet identifier |
| Address | Address of the frame |
| CRC | CRC value (cyclic redundancy check) |
| Data | Payload data bytes |
| Bit rate | Value of the bit rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

*Table 14-37: Content of the frame details table*

| Column | Description |
|---|---|
| Index | Number of the decoded field |
| Label | Name of the field |

| Column | Description |
|---|---|
| State | State of the field (OK / Not OK) |
| Value | Value of the field |

**Data format**

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

**Remote commands**

Remote commands to retrieve decode results are described in Section 18.18.20.5, "Decode results", on page 1974.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

## 14.20 USB 2.0 (option R&S MXO3-K570)

R&S MXO3-K570 is a firmware option that enables the MXO 3 to analyze Universal Serial Bus (USB) protocols, by triggering and decoding them. The option is compatible with the standards USB 1.0, USB 1.1 and USB 2.0, but not with USB HSIC (High-Speed Inter-Chip, 480 Mbit/s). It supports the data rates "Low Speed" (1.5 Mbit/s) and "Full Speed" (12 Mbit/s), both in USB 1.0 and USB 1.1, and "High Speed" (480 Mbit/s in USB 2.0).

For USB 1.1, refer to Section 14.19, "USB 1.1 (option R&S MXO3-K570)", on page 851.

- About the USB protocol....874
- USB 2.0 configuration....875
- USB 2.0 filter....878
- USB 2.0 format configuration....881
- USB 2.0 trigger....886
- Measure....890
- USB 2.0 decode results....890

### 14.20.1 About the USB protocol

Refer to Section 14.19.1, "USB protocol basics", on page 851 in the USB 1.1 (option R&S MXO3-K570) section.

## 14.20.2 USB 2.0 configuration

If you need information on how to get started with configuring the USB 2.0 setup, see Section 14.20.2.2, "Configuring USB 2.0 signals", on page 876. Otherwise proceed with the configuration settings.

### 14.20.2.1 USB 2.0 configuration settings

Access: "Menu" > "Apps" > "Protocol" tab > "USB 2.0" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.

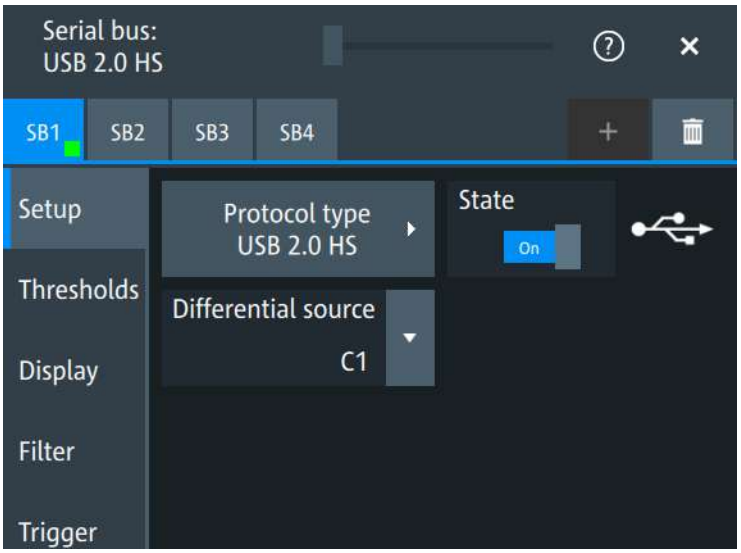



*Figure 14-64: Serial bus protocol configuration dialog for USB 2.0 HS*

See also: Section 14.1.1, "Setup - general settings", on page 489.

**Protocol type**
Selects the protocol type to be decoded.

Remote command:
`SBUS<sb>:TYPE` on page 1362

**State**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
`SBUS<sb>[:STATe]` on page 1362

**Differential source**
Selects the differential signal source of the data line for the USB 2.0 high speed (HS) protocol.

Usually, the source is one of the analog channels. Reference and math waveforms are available as sources if the trigger source is one of the analog channels but not the serial bus.

Remote command:
`SBUS<sb>:USBHspeed:SOURce` on page 1982

### 14.20.2.2 Configuring USB 2.0 signals

For configuration, assign the lines to the input channels and define the active states and the logical thresholds.

For details on configuration settings, see Section 14.20.2.1, "USB 2.0 configuration settings", on page 875.

**Serial Bus Setup**

1. Tap the "+ Bus" activator in the bottom right of the screen.
2. Select the tab of the bus that you want to set up, for example "SB1".
3. Tap the "Protocol type" button. Select the protocol: "USB 2.0 HS".
4. Select the waveform for the "Differential" source.
5. In the "Thresholds" tab, specify the upper and lower logical thresholds and the hysteresis.
6. In the "Display" tab, enable "Show decode table".

### 14.20.2.3 Threshold settings

Access: "Menu" > "Apps" > "Protocol" tab > "USB 2.0" > "Thresholds" tab




*Figure 14-65: Serial bus threshold configuration dialog for USB 2.0*

**Upper threshold**
Sets the upper threshold value for the digitization of the data line's differential signal.

For transformation of high or low levels into bus states, refer to Table 14-34. The bus states are transformed into bits according to the rules of non-return-to-zero inverted (NRZI) encoding.

For transformation of high or low levels into J or K states of the bus, refer to Table 14-34. The J and K states are transformed to bits according to the rules of non-return-to-zero inverted (NRZI) encoding.

Remote command:
`SBUS<sb>:USBHspeed:THReshold:UPPer` on page 1983

**Lower threshold**
Sets the lower threshold value for the digitization of the data line's differential signal.

See also "Upper threshold" on page 876.

Remote command:
`SBUS<sb>:USBHspeed:THReshold:LOWer` on page 1983

**Hysteresis**
Sets the hysteresis value for the data line's differential signal.

Remote command:
`SBUS<sb>:USBHspeed:THReshold:HYSTeresis` on page 1983

**Show threshold lines**
If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

Remote command:
`SBUS<sb>:THReshold` on page 1364

### 14.20.2.4 Display settings

Access: "Menu" > "Apps" > "Protocol" tab > "USB 2.0" > "Display" tab

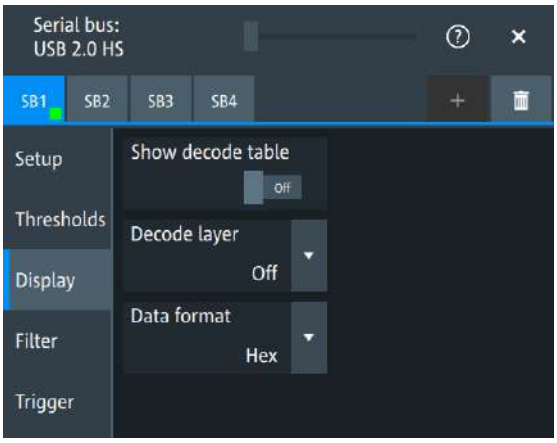



*Figure 14-66: Serial bus display configuration dialog for USB 2.0*

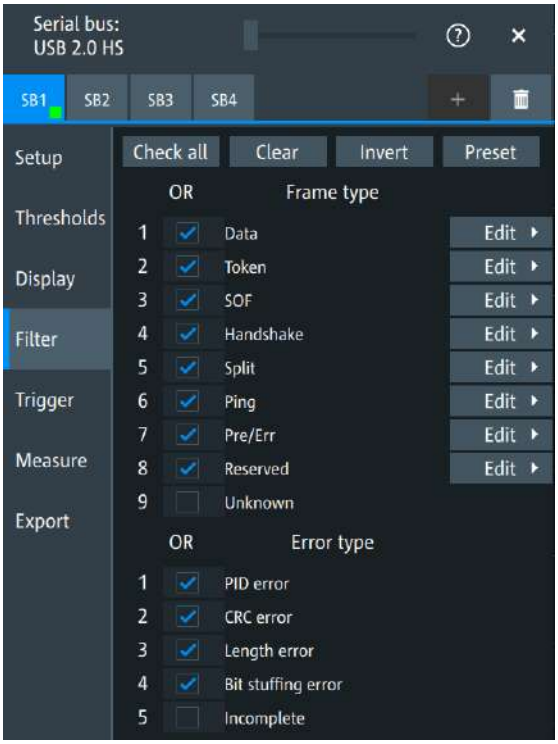
**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

| | |
|---|---|
| "Off" | Disables the display of the decode layer. |
| "Edges" | Enables the display of all edges. |
| "States" | "Enables the display of J and K states according to Table 14-34 in Section 14.19.1, "USB protocol basics", on page 851. |
| "Bits" | Enables the display of the raw, binary encoded symbols, consisting of the states "0" and "1". |
| "Words" | Enables the display of words, which are the numerical values of the individual fields that compose each packet, like PID, ADDR, DATA or CRC. |

## 14.20.3 USB 2.0 filter

Access: "Menu" > "Apps" > "Protocol" tab > "USB 2.0" > "Filter" tab




In the "Filter" tab, you can define the settings to display only the frames that match the selected filter conditions.

**Check all, Clear, Invert, Preset**
The following settings help you select the frames and errors you want to filter for. Only the frames and error types that match the selected filter conditions are displayed.

"Check all" Enables the filter for all available frames and error types.

"Clear" Disables the filter for all available frames and error types.

"Invert" Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types.

Remote command:
`SBUS<sb>:USBHspeed:FILTer:CHKall` on page 1985
`SBUS<sb>:USBHspeed:FILTer:CLR` on page 1985
`SBUS<sb>:USBHspeed:FILTer:INVert` on page 1985
`SBUS<sb>:USBHspeed:FILTer:RST` on page 1985

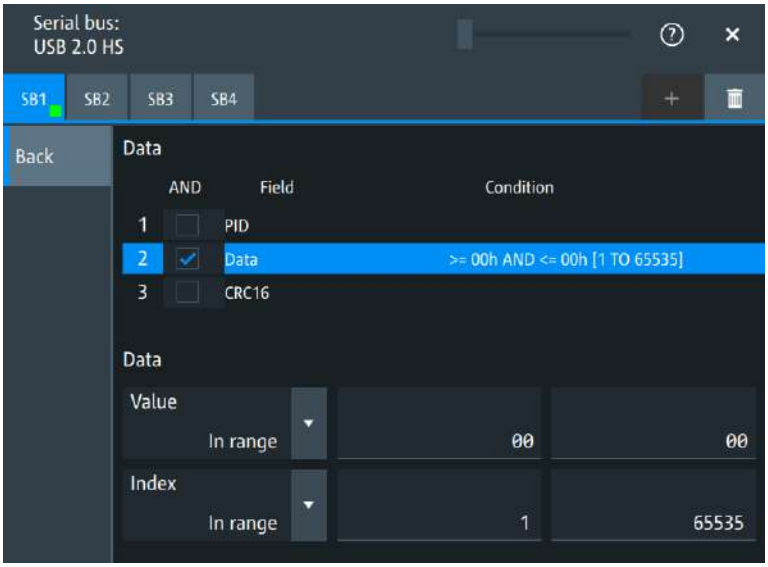
**Frame type**
Selects the frame type that you want to display. You can filter all enabled frame types simultaneously.

For selected frame types, you can also specify conditions for the value of the fields in the "Edit" dialog.

Available frames are "Data", "Token", "SOF", "Handshake", "Split", "Ping", "Pre/Err", "Reserved" and "Unknown".

Remote command:
`SBUS<sb>:USBHspeed:FILTer:FRENable` on page 1985
`SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:ENABle` on page 1985

**Edit**
Opens a dialog to define the details of the selected frame.



"Field" Enables the field type that you want to filter on for the selected frame. The available fields are "PID", "Data", "CRC5", "CRC16", "Addr", "Endp", "SC", "Port", "S", "E/U", "ET" and "Frame No.".

Remote command:
SBUS<sb>:USBHspeed:FILTer:FIENable on page 1989
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:ENABle on page 1989

"Condition" Displays the value condition for the selected field.

Remote command:
SBUS<sb>:USBHspeed:FILTer:BIT on page 1989
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:BIT on page 1989

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
SBUS<sb>:USBHspeed:FILTer:DMAX on page 1986
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DMAX on page 1986
SBUS<sb>:USBHspeed:FILTer:DMIN on page 1986
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DMIN on page 1986
SBUS<sb>:USBHspeed:FILTer:DOPerator on page 1987
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DOPerator on page 1987

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
SBUS<sb>:USBHspeed:FILTer:IMAX on page 1987
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IMAX on page 1987
SBUS<sb>:USBHspeed:FILTer:IMIN on page 1988
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IMIN on page 1988
SBUS<sb>:USBHspeed:FILTer:IOPerator on page 1988
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IOPerator on page 1988

**Error type**
Enables filtering on the selected error type.

The available error types are "PID error" (packet identifier), "CRC error" (cyclic redundancy check), "Length error", "Bit stuffing error" and "Incomplete".

Remote command:
SBUS<sb>:USBHspeed:FILTer:ERENable on page 1987
SBUS<sb>:USBHspeed:FILTer:ERRor<n>:ENABle on page 1987

### 14.20.4 USB 2.0 format configuration

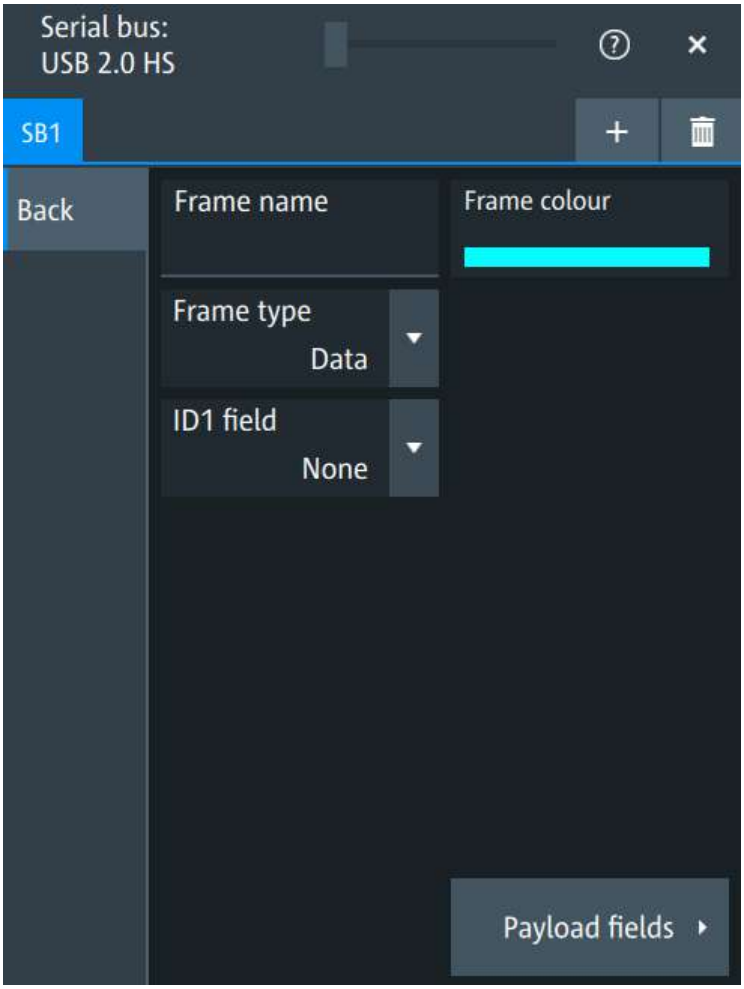
Access: "Menu" > "Apps" > "Protocol" tab > "USB 2.0" > "Format".

See Section 14.1.5, "Format", on page 497.

**Show format**
Enables the format configuration and displays the frame formats in the honey-comb view and results. If enabled, the frame formats are also selectable for triggering and filtering.

Remote command:
`SBUS<sb>:USBHspeed:FORMat:SELect` on page 1990

**Add frame**
Creates a frame format and adds it to the end of the frame list. A dialog opens to specify the frame details. The number of frame formats is limited to 100.




Remote command:
`SBUS<sb>:USBHspeed:FORMat:ADDFrame` on page 1990

**Edit list**
Enables editing the frame list. If enabled, you can delete a frame format by clicking its "Delete" icon. In a list with several entries, you can move frame formats up or down by clicking the arrow buttons. To exit the editing mode, click "Edit list" again.

Remote command:
SBUS<sb>:USBHspeed:FORMat:CLR on page 1991
SBUS<sb>:USBHspeed:FORMat:FCOunt? on page 1991

**Open**
Opens a dialog for loading an existing list of frame formats in XML file format.

Remote command:
SBUS<sb>:USBHspeed:FORMat:LOAD on page 1991

**Save as**
Opens a dialog for saving your current list of frame formats to an XML file. Saved frame formats can support efficient and convenient working.

Remote command:
SBUS<sb>:USBHspeed:FORMat:SAVE on page 1991

**Frame name**
Specifies an arbitrary name to label the currently selected frame format. Ideally, use speaking names according to the specifications of the applicable protocol standard. Frame names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:NAME on page 1992

**Frame color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:COLor on page 1992

**Frame type**
Select an original frame type.

Remote command:
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:TYPE on page 1993

**ID<n> field**
Select the first to third identifying field types.

Remote command:
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID1Field on page 1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID2Field on page 1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID3Field on page 1993

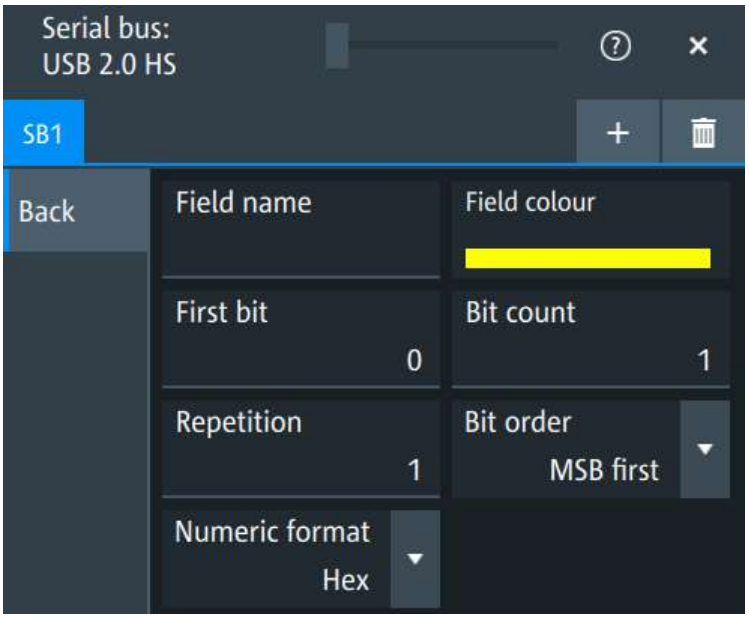
**ID<n> value**
Select the value of the corresponding field type.

Remote command:
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID1Value on page 1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID2Value on page 1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID3Value on page 1993

**Payload fields**
Opens the payload field configuration dialog. The format configuration of payload fields is optional and replaces the payload data of the identified frame.

**Add field**
Creates a new field description for the currently selected frame format and adds it to the end of the field list with color, bit position (order of transmission) and name. Brackets indicate how often bits will be repeated, [] indicates that this is an open array.

A dialog opens to specify the field details. The number of field descriptions is limited to 100.




Remote command:
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ADDField on page 1992

**Field name**
Specifies an arbitrary name for the currently selected field description. Ideally, use speaking names.

Field names do not have to be unique and are only intended to support you.

Remote command:
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:NAME on page 1994

**Field color**
Opens a dialog to select the color representation of different frames or fields in the honeycomb display. Assigning specific colors can help you to interpret the decode results more easily. You can select a predefined color or specify a "User-defined color".

Remote command:
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:COLor on page 1994

**First bit**
In order of transmission, sets the first bit of this field in the payload.

Remote command:
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FSTBit on page 1994

**Bit count**
Sets the number of bits in the selected field, i.e., the field's length. Selecting the value opens a keypad showing allowed minimum and maximum values. You can set "Min", "Max", "Reset" or adjust with the "Inc" or "Dec" buttons by 1 or 10 (step size dependent). Use "CE" to restore the value used before the keypad opened.

If the "Condition" pattern length does not match the "Bit count":
- Too long, it is truncated.
- Too short, it is left-padded with 0.

Both truncation and padding happen on the left side of the condition and follow the configured "Bit order".

Examples for "Condition" "=100"):
- "Bit order" is "MSB":
  - Bit count = 2: truncated. Result: =00
  - Bit count = 4: padded. Result: =0100
- "Bit order" is LSB first (equivalent MSB-first representation is =001):
  - Bit count = 2: truncated. Result: =00 (LSB first), =00 (MSB first)
  - Bit count = 4: padded. Result: =0100 (LSB first), =0010 (MSB first)

Remote command:
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:BITCount on page 1995

**Repetition**
Defines the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

Remote command:
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:REPetition
on page 1995

**Bit order**
Selects, in which order the bits of a field's "Condition" value are evaluated:

- "LSB first": little endian, least significant bit first
- "MSB first": big endian, most significant bit first

Remote command:
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:BITorder on page 1995

**Numeric format**
Selects how the "Condition" value is interpreted:

- Symbolic: Assigns a symbolic translation on the data fields. Only available for custom protocols.

- Value list: Assigns a custom name to each hexadecimal value; any value without an assigned value is shown as "Undefined".
- A*X+B: Assigns a linear function on the data fields.
- For details on the other numeric formats, see "Format-specific information" on page 513.

The following rules apply:

- If the condition contains at least one of "H", "h", "L", "l", "X", "x", and all other characters are only "1" and/or "0", the value is always treated as binary, regardless of the selected numeric format.
- "x" and "X" are wildcards and are supported only in binary format or when the string is auto-interpreted as binary per the rule above.
  Examples for format set to "HEX":
  – "=1HL111000": valid (read as binary)
  – "=0x10101": valid (read as binary; "x" acts as wildcard, not a hex prefix)
  – "=1010": valid (read as HEX, with a total of 16 bits)
  – "=0x5A": valid (read as HEX "5A", because "0x" is a valid hex prefix; entering 5A alone is recommended)
  – "=5X12": invalid
  – "=1H33": invalid

Remote command:
`SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FORMat` on page 1996

**Add value**
Creates a new hexagonal value for customization.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
`SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ADDValue` on page 1997

**Value**
Defines the hexagonal value to convert it into a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
`SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue` on page 1997

**Label**
Defines the label to convert the corresponding hexagonal value to a labeled state.

This setting is only available, for "Numeric format" = "Value list".

Remote command:
`SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel` on page 1997

**Unit**
Defines a unit for the linear function.

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
`SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:UNIT` on page 1998

**Parameter A,B**
Defines the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

This setting is only available, for "Numeric format" = "A*X+B".

Remote command:
`SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:PARA` on page 1998
`SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:PARB` on page 1998

## 14.20.5 USB 2.0 trigger

### 14.20.5.1 Setup A trigger

Before setting up the software trigger for the protocols, a trigger A event needs to be set up.

### 14.20.5.2 Software trigger setup

Access: "Menu" > "Apps" > "Protocol" tab > "USB 2.0" > "Trigger" > "Setup Software Trigger" tab

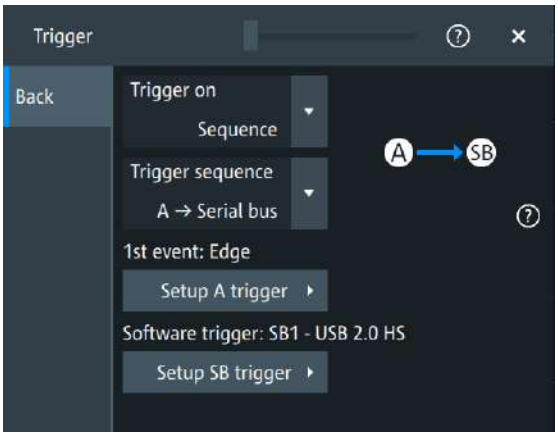



*Figure 14-67: Serial bus Trigger setup dialog for USB 2.0*

Make sure that:

- In the "Trigger" tab (Figure 14-64):
  - The correct protocol type is selected.
  - The correct channels are assigned to the corresponding data source of the serial bus.
- In the "Trigger" tab (Figure 14-67):
  - The "Trigger on" = "Sequence"
  - The "Trigger sequence" = "A → Serial bus"
  - The 1st event is set according to your requirements.
- In the "Setup SB Trigger" tab (Figure 14-68):
  - The trigger "Source" is set to the correct serial bus, for example "Serial bus 1".

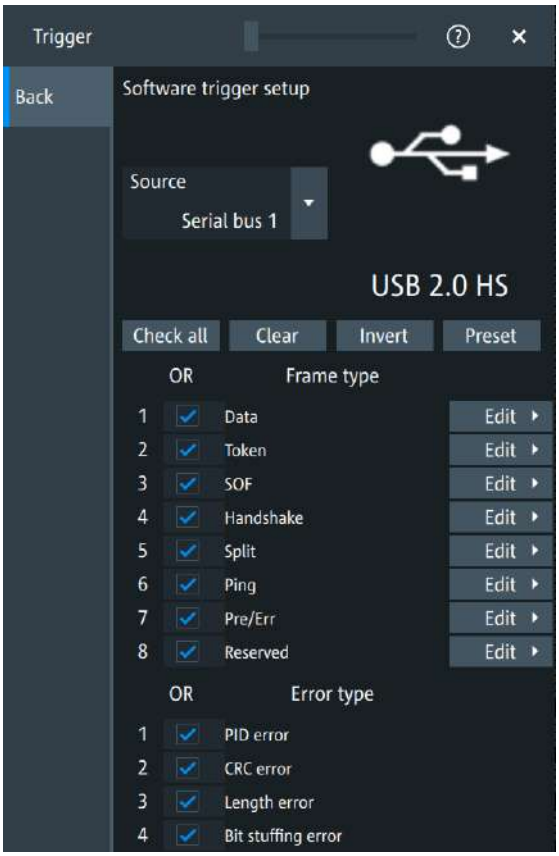



*Figure 14-68: Serial bus Software trigger setup dialog for USB 2.0*

**Check all, Clear, Invert, Preset**

The following settings help you select the frames and errors you want to trigger on.

"Check all" Enables the software trigger for all available frames and error types.

"Clear" Disables the software trigger for all available frames and error types.

"Invert" Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

"Preset" Presets the state of the selected frames and error types for the software trigger.

Remote command:
`TRIGger:SBSW:USBHspeed:CHKall` on page 2000
`TRIGger:SBSW:USBHspeed:CLR` on page 2000
`TRIGger:SBSW:USBHspeed:INVert` on page 2000
`TRIGger:SBSW:USBHspeed:RST` on page 2000

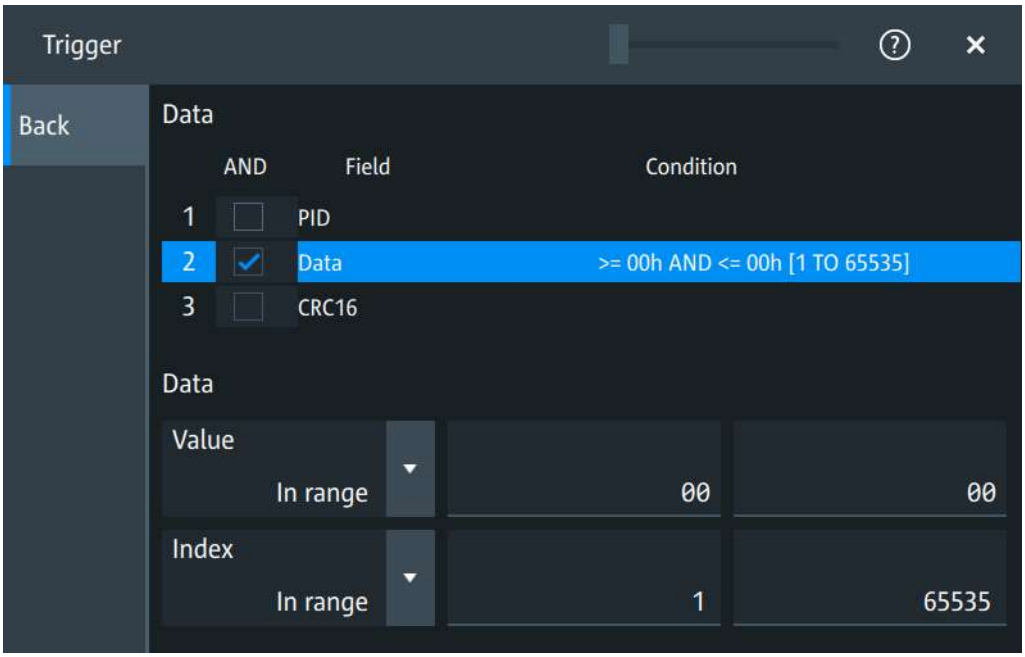
**Frame type**
Selects the frame type that you want to trigger on. You can trigger on all enabled frame types simultaneously.

For each frame type, you can also specify conditions for the value of the fields in the "Edit" dialog.

The following frames are available: "Data", "Token", "SOF", "Handshake", "Split", "Ping", "Pre/Err" and "Reserved".

Remote command:
`TRIGger:SBSW:USBHspeed:FRENable` on page 2000
`TRIGger:SBSW:USBHspeed:FRAMe<fr>:ENABle` on page 2000

**Edit**
Some trigger types have additional settings that can be defined. In this case, the "Set details" button appears next to "Type" the function. Open it for a detailed definition of the trigger conditions.



"Field" Enables the field type that you want to trigger on for the selected frame.
The available fields are "PID", "Data", "CRC5", "CRC16", "Addr", "Endp", "SC", "Port", "S", "E/U", "ET" and "Frame No.".

Remote command:
`TRIGger:SBSW:USBHspeed:FIENable` on page 2002
`TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:ENABle` on page 2002

"Condition" Displays the value condition for the selected field.

Remote command:
`TRIGger:SBSW:USBHspeed:BIT` on page 2001
`TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:BIT` on page 2001

"Value" The data setup consists of a comparison condition and one or two data patterns.

Remote command:
`TRIGger:SBSW:USBHspeed:DMAX` on page 2001
`TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DMAX` on page 2001
`TRIGger:SBSW:USBHspeed:DMIN` on page 2001
`TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DMIN` on page 2001
`TRIGger:SBSW:USBHspeed:DOPerator` on page 2002
`TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DOPerator` on page 2002

"Index" The index setup consists of a comparison condition and one or two index values.

Remote command:
`TRIGger:SBSW:USBHspeed:IMAX` on page 2002
`TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:IMAX` on page 2002
`TRIGger:SBSW:USBHspeed:IMIN` on page 2003
`TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:IMIN` on page 2003
`TRIGger:SBSW:USBHspeed:IOPerator` on page 2003
`TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:IOPerator` on page 2003

**Error type**
Enables triggering on the selected error type.

The available error types are "PID error" (packet identifier), "CRC error" (cyclic redundancy check), "Length error" and "Bit stuffing error".

Remote command:
`TRIGger:SBSW:USBHspeed:ERENable` on page 2004
`TRIGger:SBSW:USBHspeed:ERRor<m>:ENABle` on page 2004

### 14.20.6 Measure

The "Setup Measurement" button opens the dialog for automatic measurements in the "Protocol" category for the selected serial bus.

See Section 10.2.7.2, "Settings for protocol measurements", on page 300.

### 14.20.7 USB 2.0 decode results

To enable the decoding, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

#### Decode results table

The "Decode results" table contains information about all decoded frames.

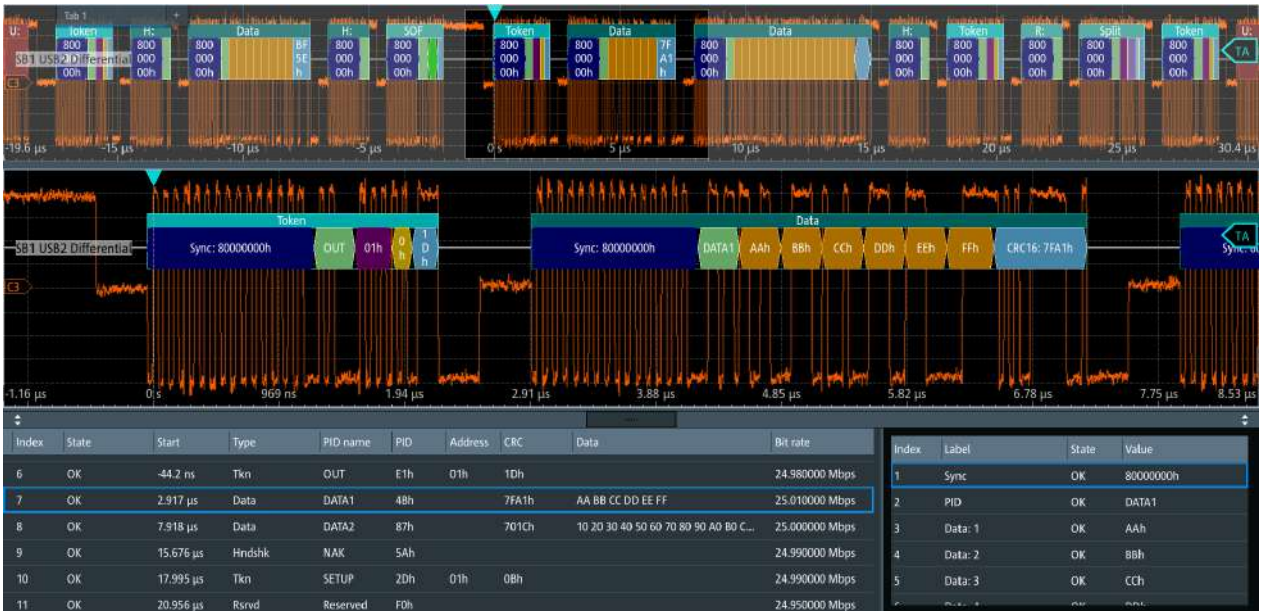



*Figure 14-69: Example of a decoded USB 2.0 signal*

The decode results table contains information about all decoded frames.

*Table 14-38: Content of the decode results table*

| Column | Description |
|---|---|
| Index | Index of the decoded frame |
| State | Overall state of the frame |
| Start | Times of frame start |
| Type | Type of the frame |
| PID name | Name of the packet identifier |
| PID | Value of the packet identifier |
| Address | Address of the frame |

| Column | Description |
|---|---|
| CRC | CRC value (cyclic redundancy check) |
| Data | Payload data bytes |
| Bit rate | Value of the bit rate |

When you select a frame in the decode results table, its field content is shown in the details table to the right.

***Table 14-39: Content of the frame details table***

| Column | Description |
|---|---|
| Index | Number of the decoded field |
| Label | Name of the field |
| State | State of the field (OK / Not OK) |
| Value | Value of the field |

### Data format

You can select various data formats for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:

- `SBUS<sb>:FORMat` on page 1363

### Remote commands

Remote commands to retrieve decode results are described in Section 18.18.21, "USB 2.0 (option R&S MXO3-K570)", on page 1982.

You can also export the decode results. For details see Section 14.1.9.4, "Exporting decode results", on page 512.

# 15 Logic analysis on digital channels

The Mixed Signal Option R&S MXO3-B1 adds logic analyzer functions to the classical oscilloscope functions. Using the MSO option, you can analyze and debug embedded systems with mixed-signal designs that use analog signals and correlated digital signals simultaneously.

The Mixed Signal Option provides 16 digital channels grouped in two logic probes (pods) with 8 channels each. The instrument ensures that analog and digital waveforms are time-aligned and synchronized so that critical timing interactions between analog and digital signals can be displayed and tested. The automatic alignment compensates the skew between the probe connectors of the analog channels and the probe boxes of the digital channels.

**Trigger**

You can trigger on digital channels. Digital channels are available as trigger source if you trigger on single events, but not in any trigger sequence. The level is the threshold of the digital signal, which is configured in "Logic" > "Setup". All other trigger settings are configured in the "Trigger" dialog.

The following trigger types are available for digital trigger sources:

- Edge
- Width
- Timeout
- State
- Pattern

**Data export**

You can export the data of digital channels to waveform files, and you can use remote commands to retrieve and transmit digital data for further analysis.

## 15.1 Logic configuration

Access: "Menu" > "Logic" > "Setup" tab

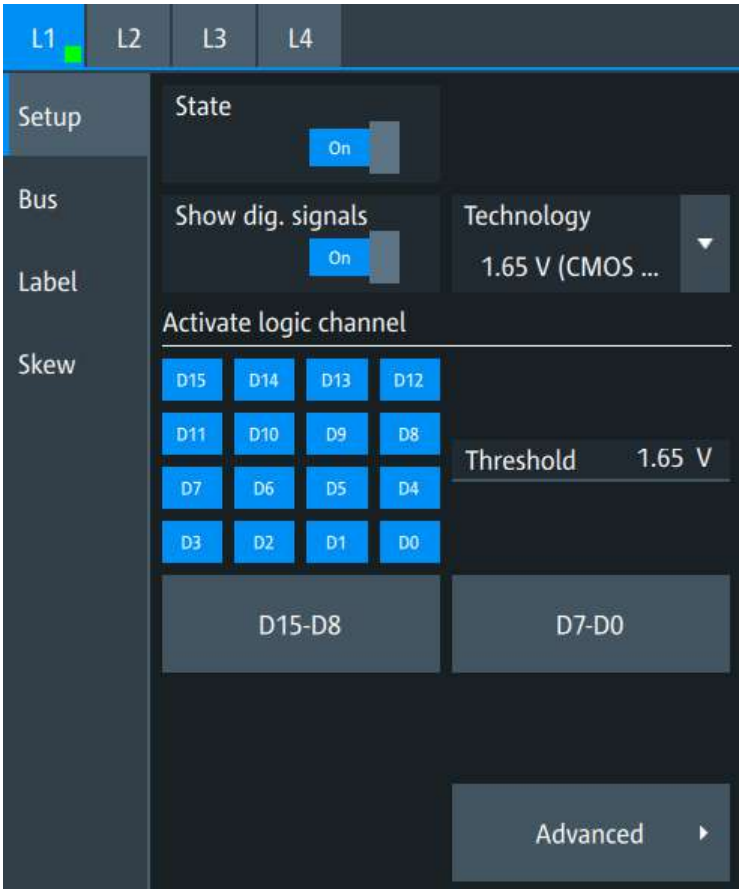
You can display digital channels individually, and you can group them and display as a logic group. 4 logic groups are available. Each digital channel can be assigned to one *active* logic only. Conflicting buses are disabled automatically.

For clocked buses, you can display the decoded data in a result box.

If you have configured several logic groups and you want to modify the settings, make sure that the tab of the correct logic is selected. Disable the logic before you change the settings.

### 15.1.1 Setup

Access: "Menu" > "Logic" > "Setup" tab




**State**
Enables the selected logic group. The corresponding signal icon appears on the signal bar.

If another active bus already uses the same digital channel, the instrument disables the other bus and shows a message.

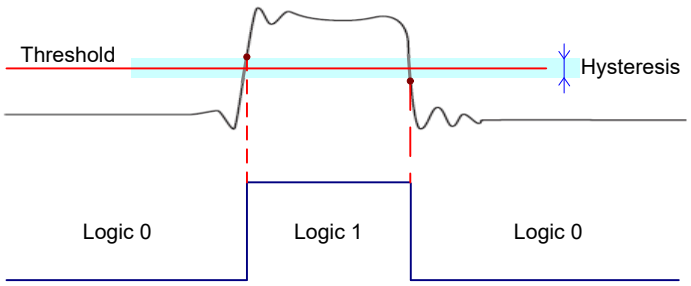
Remote command:
`PBUS<pb>:STATe` on page 2024

**Show dig. signals**
If enabled, the selected digital channels are shown in the diagram. Each channel is displayed as a logic signal.

Remote command:
`PBUS<pb>:DISPlay:SHDI` on page 2022

**Technology, Threshold**
Sets the logical threshold. For each sample, the instrument compares the input voltage with the threshold value. If the input voltage is above the threshold, the signal state "1" is stored. Otherwise, the signal state "0" is stored if the input voltage is below the threshold.

To avoid the change of signal states due to noise, a hysteresis is considered.




By default, the same threshold and hysteresis value are used for all digital channels and all logic buses: "Couple thresholds" is enabled.

You can also set different thresholds for the individual channel groups. As long as the buses are disabled, you can set different thresholds for each bus. Active buses use the same threshold and hysteresis values, the settings of the last activated bus take effect.

The range of threshold levels and the minimum voltage swing is given in the specifications document.

"Threshold" Enter the value directly in the field.

"Technology" Selects the threshold voltage for various types of integrated circuits from a list and applies it to all digital channels. The value is set to "Manual" if a user-defined threshold was entered directly.

Remote command:
`PBUS<pb>:TECHnology` on page 2025
`PBUS<pb>:THReshold<n>` on page 2025

**Active logic channel**
Set the state of the individual digital channels.

Remote command:
`PBUS<pb>:BIT<n>[:STATe]` on page 2019
`DIGital<m>:STATe` on page 2016

**Enable/Disable D7-D0,D15-D8**
The buttons select or deselect all digital channels of a pod at once.

**Advanced**
Opens a dialog for advanced threshold and hysteresis settings. You can define them in addition to the "Threshold" and "Technology".

**Level coupling ← Advanced**
Sets the threshold and the hysteresis for all digital channels and all buses to the same value.

If level coupling is off, you can set the thresholds individually for the channel groups D0 to D3, D4 to D7, D8 to D11 and D12 to D15 in the "Setup" tab. The hysteresis for each channel group is set in the "Advanced" dialog.

Remote command:
PBUS<pb>:THCoupling on page 2025

**Hysteresis ← Advanced**
Defines the size of the hysteresis for the respective channels.

"Small" The instrument sets a small value suitable for the signal and its settings. Use this setting for clean signals.

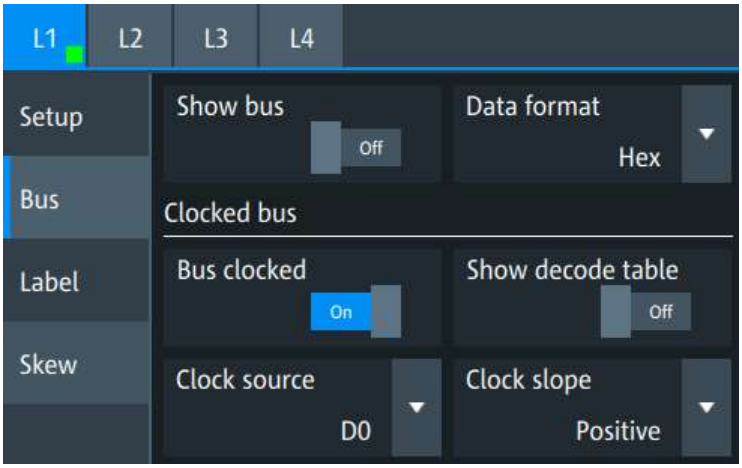
"Medium" The instrument sets the maximum value that is possible and useful for the signal and its settings. Use this setting for noisy signals.

"Large" Sets different hysteresis values for falling and rising edges to avoid an undefined state of the trigger system. Use this setting for very noisy signals.

Remote command:
PBUS<pb>:HYSTeresis<n> on page 2023

## 15.1.2 Bus

Access: "Menu" > "Logic" > "Bus" tab




**Show bus**
If enabled, the resulting bus signal and bus values are displayed in the diagram.

Remote command:
PBUS<pb>:DISPlay:SHBU on page 2022

**Data format**
Sets the data format of bus values, which are displayed in the decode table and on the comb bus display.

Available formats are: Hex, octal, binary, ASCII, string, signed, unsigned, symbolic, and auto.

Signed and Unsigned are integer data types with a maximum 16-bit length. Unsigned is used for positive integers. Signed is used for positive and negative integers.

If the target file format is BIN, you can save only signed and unsigned binary data. The data format "Signed" writes signed data; all other formats are saved as unsigned binary data.

Remote command:
`PBUS<pb>:DATA:FORMat` on page 1148
`PBUS<pb>:DATA:HEADer?` on page 1148
`PBUS<pb>:DATA[:VALues]?` on page 1148

**Clocked bus**
If a bus is a clocked bus, one of the digital channels serves as the clock of the bus.

For an unclocked bus, the logical state of the bus is determined for each sample. For a clocked bus, the logical state is determined only at the specified clock edges.

**Bus clocked ← Clocked bus**
Enable this option for a clocked bus.

Remote command:
`PBUS<pb>:CLON` on page 2020

**Show decode table ← Clocked bus**
If enabled, a result table is shown with decoded values and corresponding points in time of the bus signal. Each clock edge corresponds to one row in the table.

The decode table is only available for clocked buses to check the data words.

Remote command:
`PBUS<pb>:DECTable:SHOW` on page 2021
`PBUS<pb>:DECTable:COUNt?` on page 2021
`PBUS<pb>:DECTable:DATA?` on page 2021

**Clock source ← Clocked bus**
Selects the digital channel used as clock.

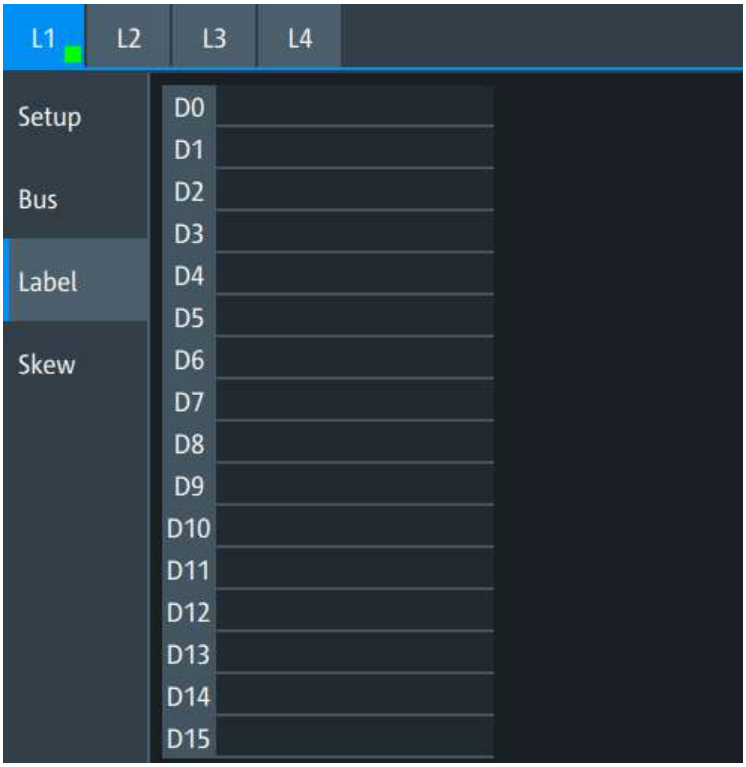
Remote command:
`PBUS<pb>:CLOCk` on page 2020

**Clock slope ← Clocked bus**
Selects the slope of the clock signal at which all digital channels of the bus are analyzed.

Remote command:
`PBUS<pb>:CLSLope` on page 2020

### 15.1.3 Label settings

Access: "Menu" > "Logic" > "Label" tab




In this tab, you can enter a name for each digital channel. The name is displayed in the diagram.

PBUS<pb>:BIT<n>:LABel on page 2018

### 15.1.4 Skew settings

Access: "Menu" > "Logic" > "Skew" tab




**D0-D15**
Sets an individual delay for each digital channel to time-align it with other digital channels.

The skew value compensates delays that are known from the circuit specifics or caused by the different length of cables. The skew between the probe boxes of the digital channels and the probe connectors of the analog channels is automatically aligned by the instrument.

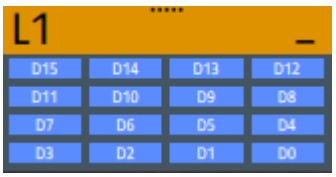
Remote command:
`PBUS<pb>:BIT<n>:SKEW` on page 2019

**Skew**
Sets a general delay for all digital channels.

Remote command:
`PBUS<pb>:SKEW` on page 2024

## 15.2 Display

Each logic group is shown in a separate diagram, and the diagrams can be minimized and arranged as usual.




You can adjust the display of the logic bus signals and the individual digital channels to optimize the analysis of bus data:

- Select the digital channels to be displayed.
  Drag the logic signals and the digital channels to the optimal position.
- Adjust the size and position of the logic signal: Tap the Lx signal or its signal icon, and use the [Position] and [Scale] knobs.
- Adjust the size and position of active digital channels: Tap one of the Dx signals, and use the [Position] and [Scale] knobs.
- Show the result table of the decoded clocked bus signal.

Remote commands:

- `PBUS<pb>:DIGSignals:POSition` on page 2022
- `PBUS<pb>:DIGSignals:SCALe` on page 2022
- `PBUS<pb>:POSition` on page 2023
- `PBUS<pb>:SCALe` on page 2024

To access and analyze one or more specific acquisitions, you can use the "History" in the common way.

Furthermore, you can zoom in digital signals and bus signal in the same way as in analog waveforms.

### 15.2.1 Logic bus - decode table

Decoding is available for clocked buses.

The decode table shows the decoded data words of the bus signal and the corresponding time. Each clock edge corresponds to one row in the table. Beside the table, you can select the data format of the bus values.




The results can be saved to a `.csv` or `.html` file.

# 16 Waveform generator (option R&S MXO3-B6)

The MXO 3 includes a 50 MHz waveform generator. The waveform generator can output simple functions, modulated sine waveform, arbitrary waveforms and sweep waveforms.

The instrument preset does not affect the generator settings. The generator has its own preset ("Default setup"). When the instrument is shut down, the waveform generator settings are stored. At the instrument start, these settings are restored but the generator is switched off. When a saveset of instrument settings is recalled, the generator is also switched off but the generator settings remain unchanged.

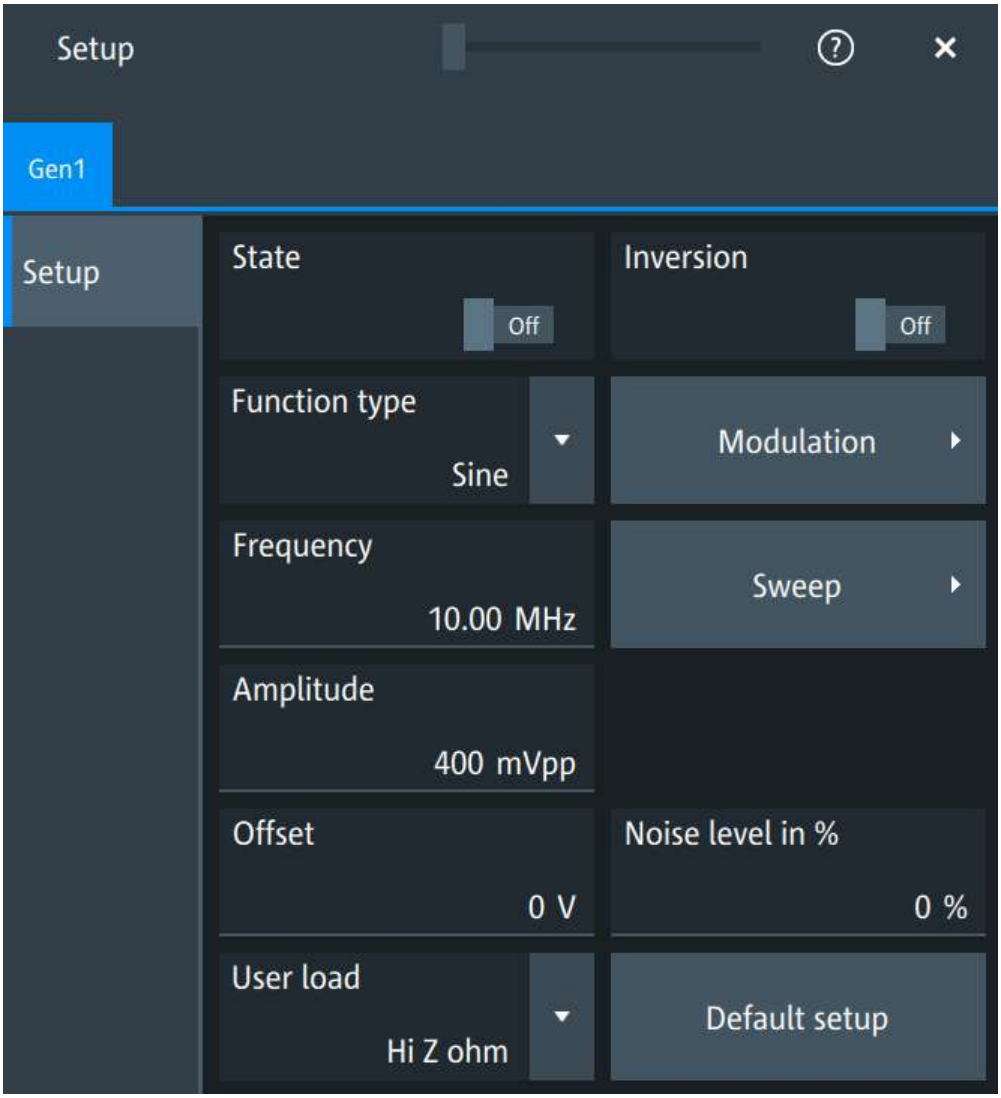
If you often switch on and off a generator, add a generator icon to the toolbar ("Gen1"). Tap the icon to switch on or off the generator.

## 16.1 Setup of the waveform generator

Access: "Gen" signal activator > "Setup" tab.

### 16.1.1 General settings

The "Setup" tab provides general settings like enabling the waveform generator, and setting that depend on the selected function type.




**State**
Enables the function generator.

If you often switch on and off a generator, use the generator icon on the toolbar ("Gen1").

Remote command:
`WGENerator<wg>[:ENABle]` on page 2032

**Inversion**
Inverts the waveform at the offset level.

Remote command:
`WGENerator<wg>:VOLTage:INVersion` on page 2032

**Function type**
Selects the type of waveform to be generated.

For all waveforms, you can set:

- Frequency
- Amplitude, Peak amplitude
- Offset
- Noise level in % or DC level
- User load

| | |
|---|---|
| "Sine" | Generates a sine wave.<br>Modulation settings and Sweep settings are available. |
| "Square" | Generates a square wave. You can set the Duty cycle.<br>Modulation settings is available. |
| "Ramp" | Generates a ramp signal. You can set the Symmetry. |
| "DC" | Generates a direct current (DC) signal. You can set the DC level. |
| "Pulse" | Generates a pulse signal. Additional settings are the Period and the Pulse width.<br>Modulation settings is available. |
| "Cardinal sine" | Generates a cardinal sine wave. No additional settings are needed. |
| "Cardiac" | Generates a cardiac signal. No additional settings are needed. |
| "Gauss" | Generates a Gaussian signal. No additional settings are needed. |
| "Lorentz" | Generates a Lorentz signal. No additional settings are needed. |
| "Haversine" | Generates a haversine function signal (great-circle distance between two points on a sphere). No additional settings are needed. |
| "Exp. rise" | Generates an exponential rise signal. No additional settings are needed. |
| "Exp. fall" | Generates an exponential fall signal. No additional settings are needed. |
| "Arbitrary" | Generates an arbitrary waveform, which is copied from an existing waveform, or loaded from a file. See Section 16.1.4, "Arbitrary waveforms", on page 908. |

Remote command:
`WGENerator<wg>:FUNCtion[:SELect]` on page 2029

**Frequency**
Sets the frequency of the waveform.

The available frequency range depends on the selected "Function type", see Frequency range of the function generator waveforms.

***Table 16-1: Frequency range of the function generator waveforms***

| "Function type" | Min frequency | Max frequency |
|---|---|---|
| "Sine" | 0.001 Hz | 50 MHz |
| "Square" | 0.001 Hz | 30 MHz |
| "Ramp" | 0.001 Hz | 1 MHz |
| "DC" | - | - |
| "Pulse" | 0.001 Hz | 20 MHz |
| "Cardinal sine" | 0.001 Hz | 5 MHz |

| "Function type" | Min frequency | Max frequency |
|---|---|---|
| "Cardiac" | 0.001 Hz | 1 MHz |
| "Gauss" | 0.001 Hz | 25 MHz |
| "Lorentz" | 0.001 Hz | 10 MHz |
| "Haversine" | 0.001 Hz | 50 MHz |
| "Exp.rise" | 0.001 Hz | 10 MHz |
| "Ep. fall" | 0.001 Hz | 10 MHz |

Remote command:
`WGENerator<wg>:FREQuency` on page 2028

**Amplitude, Peak amplitude**
Sets the amplitude of the waveform. If AM modulation is active, this value defines the peak amplitude of the modulated signal.

Remote command:
`WGENerator<wg>:VOLTage[:VPP]` on page 2032

**Carrier amplitude**
Sets the mean peak-to-peak voltage of the modulated signal if AM modulation is active.

The setting is only available, if "Modulation" is on and set to "AM".

Remote command:
`WGENerator<wg>:MODulation:AM:CAMPlitude` on page 2030

**Offset**
Sets the vertical offset of the generated waveform.

Remote command:
`WGENerator<wg>:VOLTage:OFFSet` on page 2032

**Arbitrary Setup**
Opens a menu to configure the arbitrary waveform.

See Section 16.1.4, "Arbitrary waveforms", on page 908.

**Sweep**
Opens a menu to configure the sweep.

See Section 16.1.3, "Sweep settings", on page 907.

**Modulation**
Opens a menu to configure the modulation.

See Section 16.1.2, "Modulation settings", on page 905.

**Noise level in %**
Sets the level of the noise in percentage of the set "Amplitude" output of the signal.

Remote command:
WGENerator<wg>:MODulation:NLPCent on page 2038

**User load**
Select the user load, the load of the DUT at its connection.

You can select either a "50Ω" or a "High-Z" (high input impedance) load.

Remote command:
WGENerator<wg>:OUTPut[:LOAD] on page 2029

**Symmetry**
Sets the symmetry of a ramp waveform, the percentage of time the waveform is rising. By changing the symmetry of the ramp, you can create, for example, triangular waveforms.

50% defines symmetric triangles. Values <50% define triangles with a steeper rising edge leaned to the left. Values >50% define triangles with a steeper falling edge leaned to the right.

Remote command:
WGENerator<wg>:FUNCtion:RAMP[:SYMMetry] on page 2028

**Duty cycle**
Sets the duty cycle for the pulse function.

The duty cycle expresses for what percentage of the period, the signal state is high.

Remote command:
WGENerator<wg>:FUNCtion[:SQUare]:DCYCle on page 2029

**Pulse width**
Sets the pulse width, the pulse duration of the generated pulse waveform.

Remote command:
WGENerator<wg>:FUNCtion:PULSe[:WIDTh] on page 2028

**Default setup**
Presets the generator to a default setup. The default includes the following settings:

- "Function type" = "Sine"
- "Frequency" = "1 MHz"
- "Amplitude" = "1 Vpp"

Remote command:
WGENerator<wg>:PRESet on page 2030

**DC level**
Available for "Function type" = DC.

Sets the level for the DC signal.

Remote command:
WGENerator<wg>:VOLTage:DCLevel on page 2031

**Noise level for DC**
Available for "Function type" = "DC".

Sets the level of the noise for the DC signal.

Remote command:
WGENerator<wg>:MODulation:NDCLevel on page 2038

**Period**
Available for "Function type" = "Pulse".

Sets the period of the pulse waveform.

Remote command:
WGENerator<wg>:PERiod on page 2030

## 16.1.2 Modulation settings

Modulation is available only for sine and square waveforms.

Access: "Gen" signal activator > "Setup" tab > "Function type" = "Sine" or "Square" > "Modulation".

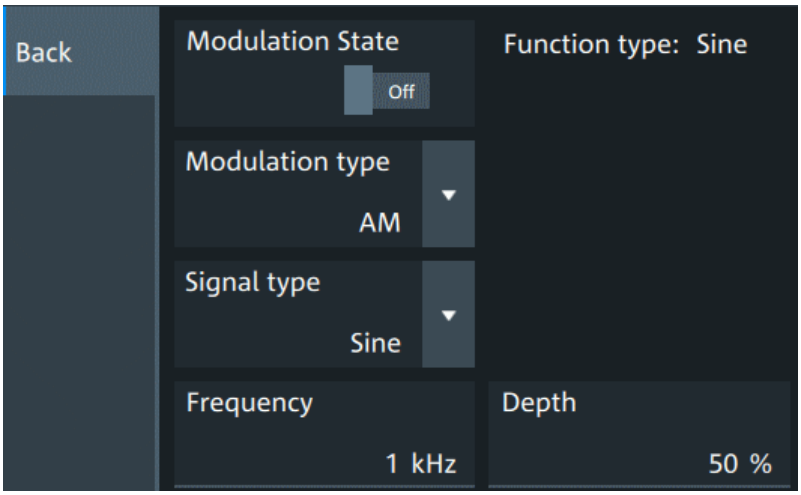



*Figure 16-1: Modulation setup*

**Modulation state**
Enables or disables modulation.

Modulation is available only for "Function type" = "Sine"/"Square".

Remote command:
WGENerator<wg>:MODulation[:STATe] on page 2033

**Modulation type**
Selects the modulation type, which defines how the carrier signal is modified.

"AM" Amplitude modulation. The amplitude of the carrier signal is varied according to the modulation signal. Available for "Function type" = "Sine".

| | |
|---|---|
| "FM" | Frequency modulation. The frequency of the carrier signal is varied according to the modulation signal. Available for "Function type" = "Sine". |
| "PWM" | Pulse width modulation. The time for which the signal is in a high state is varied according to the modulation signal. Available for "Function type" = "Square". |
| "FSK" | Frequency shift keying (FSK) modulation. The signal frequency switches between "Frequency 1" and "Frequency 2" at a "FSK rate". Available for "Function type" = "Sine". |

Remote command:
`WGENerator<wg>:MODulation:TYPE` on page 2040

**Signal type**
Selects the type of the modulating signal for AM, FM or PWM modulation.

Remote command:
`WGENerator<wg>:MODulation:AM[:FUNCtion]` on page 2035
`WGENerator<wg>:MODulation:FM[:FUNCtion]` on page 2037
`WGENerator<wg>:MODulation:PWM[:FUNCtion]` on page 2040

**Frequency**
Sets the frequency of the modulating waveform for AM/FM/PWM modulation.

Remote command:
`WGENerator<wg>:MODulation:AM:FREQuency` on page 2034
`WGENerator<wg>:MODulation:FM:FREQuency` on page 2036
`WGENerator<wg>:MODulation:PWM:FREQuency` on page 2039

**Depth**
Sets the modulation depth, the percentage of the amplitude range that is used for AM modulation.

Remote command:
`WGENerator<wg>:MODulation:AM:DEPTh` on page 2034

**Duty cycle**
Sets the duty cycle for a square waveform. The duty cycle expresses for what percentage fraction of the period, the waveform is active, i.e. the signal state is high.

Remote command:
`WGENerator<wg>:MODulation:AM:DCYCle` on page 2034
`WGENerator<wg>:MODulation:FM:DCYCle` on page 2036
`WGENerator<wg>:MODulation:PWM:DCYCle` on page 2039

**Symmetry**
Sets the symmetry for the ramp modulation waveform, the percentage of time that the waveform is rising.

Remote command:
`WGENerator<wg>:MODulation:AM:SYMMetry` on page 2035
`WGENerator<wg>:MODulation:FM:SYMMetry` on page 2037
`WGENerator<wg>:MODulation:PWM:SYMMetry` on page 2040

**Deviation**
Sets the frequency deviation, the maximum difference between the FM modulated signal and the carrier signal.

Remote command:
`WGENerator<wg>:MODulation:FM:DEViation` on page 2036

**Modulation depth**
Sets the modulation depth, the percentage of the amplitude range that is used for PWM modulation.

Remote command:
`WGENerator<wg>:MODulation:PWM:DEPTh` on page 2039

**Frequency 1/Frequency 2**
Sets the frequency of the first /second signal in FSK modulated signal.

Remote command:
`WGENerator<wg>:MODulation:FSK:FONE` on page 2037
`WGENerator<wg>:MODulation:FSK:FTWO` on page 2037

**FSK rate**
Sets the frequency at which signal switches between "Frequency 1" and "Frequency 2".

Remote command:
`WGENerator<wg>:MODulation:FSK[:RATE]` on page 2038

### 16.1.3 Sweep settings

Sweep is available only for sine waveforms.

Access: "Gen" signal activator > "Setup" tab > "Function type" = "Sine" > "Sweep".

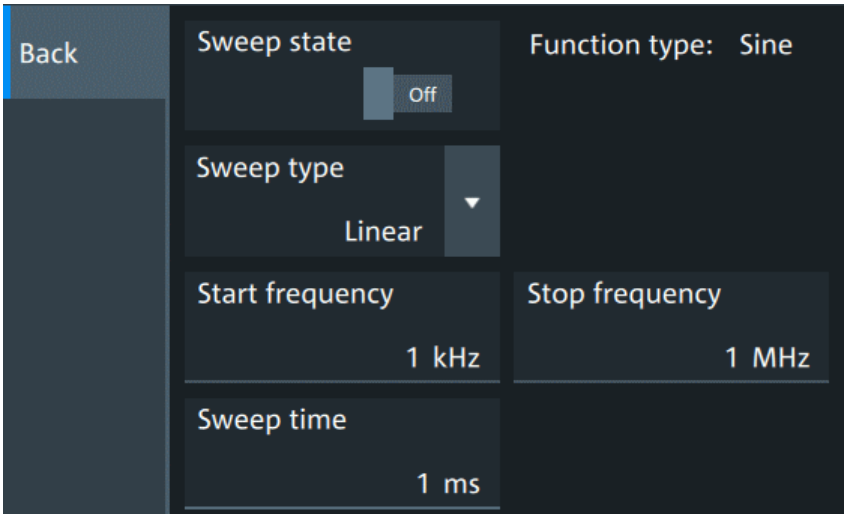



*Figure 16-2: Sweep setup*

In the sweep mode, the MXO 3 generates a signal whose frequency gradually changes from the "Start frequency" to the "Stop frequency" for a certain "Sweep time".

For a step-by-step description of the sweep setup, see Section 16.2.3, "Configuring a sine sweep waveform", on page 914.

**Sweep state**
Enables or disables the sweeping.

Remote command:
`WGENerator<wg>:SWEep[:STATe]` on page 2043

**Sweep type**
Sets the type of the sweep, a linear or logarithmic change of the frequency.

Remote command:
`WGENerator<wg>:SWEep:TYPE` on page 2043

**Start frequency**
Sets the start frequency of the sweep signal.

Remote command:
`WGENerator<wg>:SWEep:FSTart` on page 2044

**Stop frequency**
Sets the stop frequency of the sweep signal.

Remote command:
`WGENerator<wg>:SWEep:FEND` on page 2044

**Sweep time**
Sets the duration of the sweep.

Remote command:
`WGENerator<wg>:SWEep:TIME` on page 2044

## 16.1.4 Arbitrary waveforms

The arbitrary waveform generator allows you to output a user-defined waveform for testing your devices. An arbitrary waveform is loaded from a file. You can load CSV files that follow a defined structure, or saved reference waveforms.

See also: Section 16.2.4, "Configuring an arbitrary waveform", on page 914.

### 16.1.4.1 Arbitrary waveform files

You can load CSV files that have a Rohde & Schwarz WaveGen format.

**Content and format of the Rohde & Schwarz arbitrary waveform CSV files**

The waveform generator format can contain the following values:

- Rate

- Time value
- Voltage value

If all the values are defined, the file format is as follows:

```
Rate 5000000            //Sample rate of the arbitrary waveform.
0.000000E+000,-5.995    //Time value 1, Voltage value 1
1.237011E-005,-6.0      //Time value 2, Voltage value 2
.......
```

In this case, the rate is reflected in the "Sample Rate" field of the user interface. The total number of Time/Voltage values is reflected in the "Samples" of the user interface. Anything written after `//` is ignored as a comment.

You can define only some of the values. According to what you define, the file format looks different and is handled differently:

- With specified *Rate*:
  Time values are ignored. You can specify just rate and voltage values as below:

```
Rate 5000000               //Sample rate of the arbitrary waveform
-5.995                     //Voltage value 1
-6.0                       //Voltage value 2
.......
```

- Without specified *Rate* and without specified *Time* values:
  The last user-defined sample rate is used to calculate the waveform. You can change the "Sample rate" in the user interface. The voltage values are then played with this sample rate.

```
-5.995                //Voltage value 1
-6.0                   //Voltage value 2
.......
```

- Without specified *Rate* and with specified *Time*:
  The timing information of the first 2 time values is used to calculate the sample rate.

**Example:**

Consider the following file:

```
0.000000E+000,-5.995    //Time value 1, Voltage value 1
1.237011E-005,-6.0      //Time value 2, Voltage value 2
```

The sample rate is:

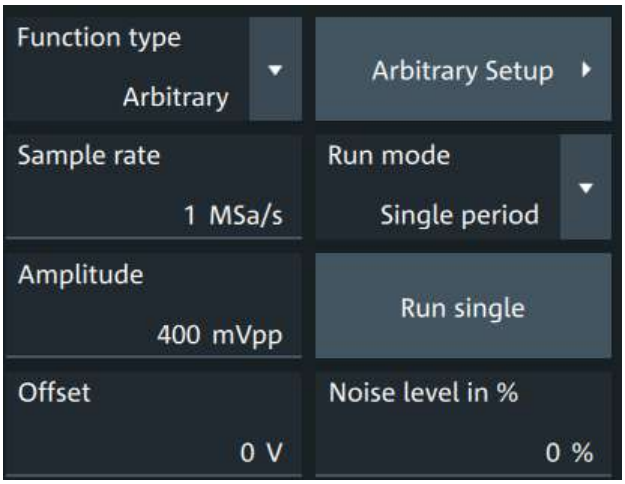
*Sample rate = 1/ Time between first two samples* =1 / 1.237011E-005 = 80.840KSampe/sec

### 16.1.4.2 Arbitrary settings

Access: "Gen" signal activator> "Gen1" or "Gen2" > "Setup" tab > "Function type" = "Arbitrary"

Access: "Gen" signal activator > "Setup" tab > "Function type" = "Arbitrary".

The following settings in the "Setup" tab and the "Arbitrary Setup" dialog are specific for generated arbitrary waveforms.




Run mode....910
Run single....910
Arbitrary Setup....910
Arb wfm source....911
Sample rate....911
Number of samples....911
Open....911

**Run mode**
Selects the duration for which the signal of the arbitrary generator is output after the trigger event. You can choose between a "Continuous" and "Single period" duration.

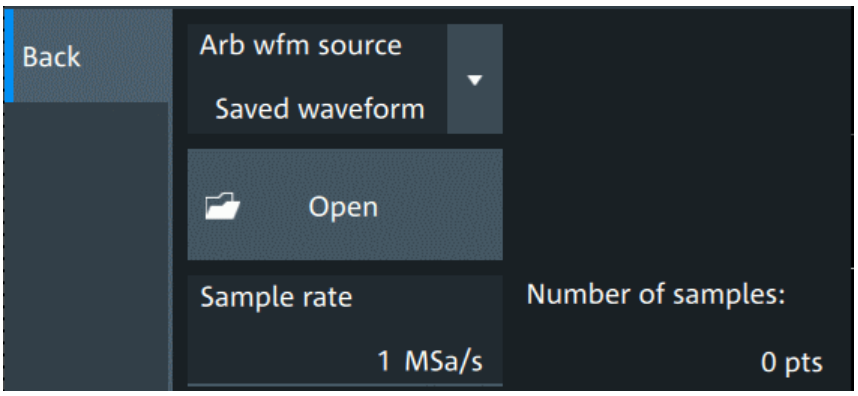
Remote command:
`WGENerator<wg>:ARBGen:RUNMode` on page 2041

**Run single**
Generates a single period of the arbitrary waveform if "Run mode" is set to "Single period".

If you load the arbitrary waveform from a file, first load the file, and then connect the DUT. See also: "Open" on page 911.

Remote command:
`WGENerator<wg>:ARBGen:RUNSingle` on page 2042

**Arbitrary Setup**
Opens a dialog to define the arbitrary waveform.




**Arb wfm source**
Selects the source of the arbitrary waveform.

You can load CSV files that follow a defined structure, or saved reference waveforms.

Remote command:
`WGENerator<wg>:ARBGen[:SOURce]` on page 2043

**Sample rate**
Sets the sample rate for the arbitrary waveform.

If the sample rate is given in the arbitrary waveform file (CSV or REF file), the instrument displays the value here, and you can change it. If no sample rate is available in the data, you can set the required sample rate.

Remote command:
`WGENerator<wg>:ARBGen:SRATe` on page 2042

**Number of samples**
Displays the number of samples in the loaded waveform.

Remote command:
`WGENerator<wg>:ARBGen:SAMPles?` on page 2042

**Open**
Opens a dialog to select the saved waveform, and loads the file. The setting is available if "Arb wfm source" is set to "Saved waveform".

You can load CSV files, or REF waveforms (saved reference waveforms). See also: Section 16.1.4.1, "Arbitrary waveform files", on page 908.

When the file is loaded, the output voltage is set to the value of the first sample in the file. Therefore, if you want to output single periods of the arbitrary waveform, first load the waveform file, and then connect the DUT.

Remote command:
`WGENerator<wg>:ARBGen:NAME` on page 2041
`WGENerator<wg>:ARBGen:OPEN` on page 2041

## 16.2 Configuring the waveform generator

This section explains step by step how to configure the waveform generator.

- Configuring a function waveform....912
- Configuring a modulation waveform....912
- Configuring a sine sweep waveform....914
- Configuring an arbitrary waveform....914

### 16.2.1 Configuring a function waveform

1. Tap the "Gen" signal activator.
2. In the "Setup" tab, select the "Function type" that you want to generate, e.g. "Sine".
3. Depending on the selected "Function type", configure the settings of the waveform like "Frequency" and "Amplitude".
4. If necessary, change the "User load" settings, or add "Noise level in %" to the waveform.
5. Set "State" = "On" to output the waveform at the output connector of the waveform generator.

### 16.2.2 Configuring a modulation waveform

**Generating an AM modulated waveform**

1. Tap the "Gen" signal activator.
2. In the "Setup" tab, set the "Function type" = "Sine".
3. Press "Modulation" to open the modulation dialog.
4. Set "Modulation type" = "AM".
5. Set the "Carrier frequency" and the "Carrier period".
6. Tap "Modulation signal", and select the required waveform.
7. Depending on the selected "Modulation signal", configure the settings of the waveform like "Modulation freq" and "Modulation depth".
8. Set "Modulation state" = "On" to enable the modulation.
9. Press "Back" to return to the "Setup" tab.
10. Set "State" = "On" to output the waveform at the output connector of the waveform generator.

**Generating an FM modulated waveform**

1. Tap the "Gen" signal activator.
2. In the "Setup" tab, set the "Function type" = "Sine".
3. Press "Modulation" to open the modulation dialog.
4. Set "Modulation type" = "FM".
5. Select the "Signal type".
6. Set the "Frequency" and the "Deviation".
7. Set "Modulation state" = "On" to enable the modulation.
8. Press "Back" to return to the "Setup" tab.
9. Set "State" = "On" to output the waveform at the output connector of the waveform generator.

**Generating an FSK modulated waveform**

1. Tap the "Gen" signal activator.
2. In the "Setup" tab, set the "Function type" = "Sine".
3. Press "Modulation" to open the modulation dialog.
4. Set "Modulation type" = "FSK".
5. Set the "Frequency 1", "Frequency 2" and the "FSK rate".
6. Set "Modulation state" = "On" to enable the modulation.
7. Press "Back" to return to the "Setup" tab.
8. Set "State" = "On" to output the waveform at the output connector of the waveform generator.

**Generating a PWM modulated waveform**

1. Tap the "Gen" signal activator.
2. In the "Setup" tab, set the "Function type" = "Square".
3. Press "Modulation" to open the modulation dialog.
4. Set "Modulation type" = "PWM".
5. Tap "Signal type", and select the required waveform.
6. Configure the settings of the waveform like "Frequency" and "Modulation depth".
7. Set "Modulation state" = "On" to enable the modulation.
8. Press "Back" to return to the "Setup" tab.
9. Set "State" = "On" to output the waveform at the output connector of the waveform generator.

### 16.2.3 Configuring a sine sweep waveform

1. Tap the "Gen" signal activator.
2. In the "Setup" tab, set "Function type" = "Sine".
3. Press "Sweep" to open the sweep dialog.
4. Select the "Sweep type".
5. Set the "Start frequency", the "Stop frequency" and the "Sweep time".
6. Set "Sweep state" = "On" to enable the sweep.
7. Press "Back" to return to the "Setup" tab.
8. Set "State" = "On" to output the waveform at the output connector of the waveform generator.

### 16.2.4 Configuring an arbitrary waveform

For a description of the settings, see Section 16.1.4, "Arbitrary waveforms", on page 908.

**Generating an arbitrary waveform from a saved file**

If you want to output single periods of the arbitrary waveform, first set up the generator, and then connect the DUT.

1. Tap the "Gen" signal activator.
2. In the "Setup" tab, set "Function type" = "Arbitrary".
3. Select "Arbitrary Setup" to open the arbitrary dialog.
4. Set the "Arb wfm source" to "Saved waveform".
5. Set the "Sample rate" if it is not defined in the file data.
6. Check "Number of samples".
7. Select "Back" to return to the "Setup" tab.
8. Select the "Run mode".
9. If "Run mode" = "Repetitive", set "State" = "On" to output the waveform at the output connector of the waveform generator.
10. If "Run mode" = "Single period":
    a) Connect the DUT.
    b) Set "State" = "On" to enable the output of the waveform generator.
    c) Tap "Run single" to output one waveform.

# 17 Network operation

The operating system is the basis of the instrument's firmware, it provides basic functions such as logon, password protection, virus protection, and connection to a network.

In addition to working with the MXO 3 directly with the touchscreen and the keys, you can also operate the instrument from a remote PC. Various methods of remote operation and control in a LAN network are supported:

- Using the web interface
- Using a VNC client
- Remote control with SCPI commands

You can also connect and synchronize two MXO oscilloscopes together and thus extend the number of channels. If you use R&S ScopeStudio on a computer, you can connect and synchronize two oscilloscopes to the software.

- Connecting the instrument to the network (LAN) ..... 915
- Web interface ..... 917
- Remote operation with VNC client ..... 923
- Remote control ..... 923
- Remote control - status reporting system ..... 927

## 17.1 Connecting the instrument to the network (LAN)

**Network environment**

Before connecting the product to a LAN, consider the following:

- Install the latest firmware to reduce security risks.
- For internet or remote access, use secured connections if applicable.
- Ensure that the network settings comply with the security policies of your company. Contact your local system administrator or IT department before connecting your product to your company LAN.
- When connected to the LAN, the product may potentially be accessed from the internet, which may be a security risk. For example, attackers might misuse or damage the product.

### 17.1.1 Connecting the instrument to the network

There are two methods to establish a LAN connection to the instrument:

- A non-dedicated network (Ethernet) connection from the instrument to an existing network.
- A dedicated network connection (Point-to-point connection) between the instrument and a single computer.

**To connect a LAN cable**

► Connect an RJ-45 cable to the LAN connector on the rear panel, and to the LAN.

For supported LAN interfaces, refer to the specifications document.

### 17.1.2 Assigning the IP address

Depending on the network capacities, the TCP/IP address information for the instrument can be obtained in different ways.

- If the network supports the Dynamic Host Configuration Protocol (DHCP), the address is assigned automatically.
- If the network does not support DHCP, or if the instrument is set to use manual configuration, the addresses must be set manually.

By default, the instrument is configured to use DHCP and obtains all address information automatically. Thus it is safe to establish a physical connection to the LAN without any previous instrument configuration.

**To assign the IP address manually on the instrument**

1. **NOTICE!** Connection errors can affect the entire network. If your network does not support DHCP, or if you choose to disable dynamic TCP/IP configuration, you must assign valid address information before connecting the instrument to the LAN.

   Contact your network administrator to obtain a valid IP address.
2. Open "Menu" > "Settings".
3. Select "System" > "Network" > "Advanced".
4. Disable "Automatic IP (DHCP)".
5. Enter the address information as obtained from the network administrator.

### 17.1.3 Using host names

In a LAN that uses a DNS server (Domain Name System server), each computer and instrument in the LAN can be accessed via an unambiguous host name (or computer name) instead of the IP address. The DNS server translates the host name to the IP address. Host names are useful when a DHCP server is used, as a new IP address can be assigned each time the instrument is restarted.

Each instrument is delivered with an assigned host name, but you can change this name.

The default host name is the device name, a non-case-sensitive string with the following syntax: <instrument_model>-<serial_number>.

You can find the current host name, the model and the serial number in the "Device summary", when you tap the Rohde & Schwarz logo in the upper right corner of the screen. See also: Section 4.13, "Information and notifications", on page 79.

The serial number can also be found on the rear panel of the instrument. It is the third part of the device ID printed on the barcode sticker:




**To change the host name**

1. Open the "Menu".
2. Select "Settings" > "System" > "Network".
3. In "Host name", enter the new host name.
4. Confirm the entry.
5. Reboot the instrument.

## 17.2 Web interface

If the MXO 3 is connected to a computer via LAN, you can operate the instrument from the computer. No additional tools are required, you need only a web browser.

### 17.2.1 Settings on the MXO 3

The "Network" tab of the "System" dialog box provides network information and settings. See Section 5.1.2, "Network settings", on page 85.

### 17.2.2 Web browser

The instrument's web interface works with all W3C compliant browsers.

**To open the instrument's web browser**

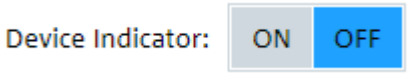
1. Open a browser on a computer connected to the instrument via LAN.
2. Type the instrument's host name or IP address in the address field of the browser on your PC, e.g. "http://11.111.11.222".

   The instrument homepage opens.

#### 17.2.2.1 Instrument homepage

The instrument homepage displays the device information in read-only format, including DNS host names, IP address, and the VISA resource strings.

At the bottom of the homepage, you find the "Device Indicator". Activate the indicator to find the instrument. In the instrument GUI, above the waveform diagram, the connection status icon starts blinking.




The navigation menu of the instrument's browser interface has the following items.




- "LAN configuration" opens the menu with configuration pages.
- "Web control" emulates the front panel and shows the instrument display. You see a live image of the instrument, and you can operate the instrument remotely. You can use the keys, the knobs and the menus in the same way as directly on the instrument.
- "File manager" provides access to the user data that is saved on the instrument, upload and download.

### 17.2.2.2 LAN configuration

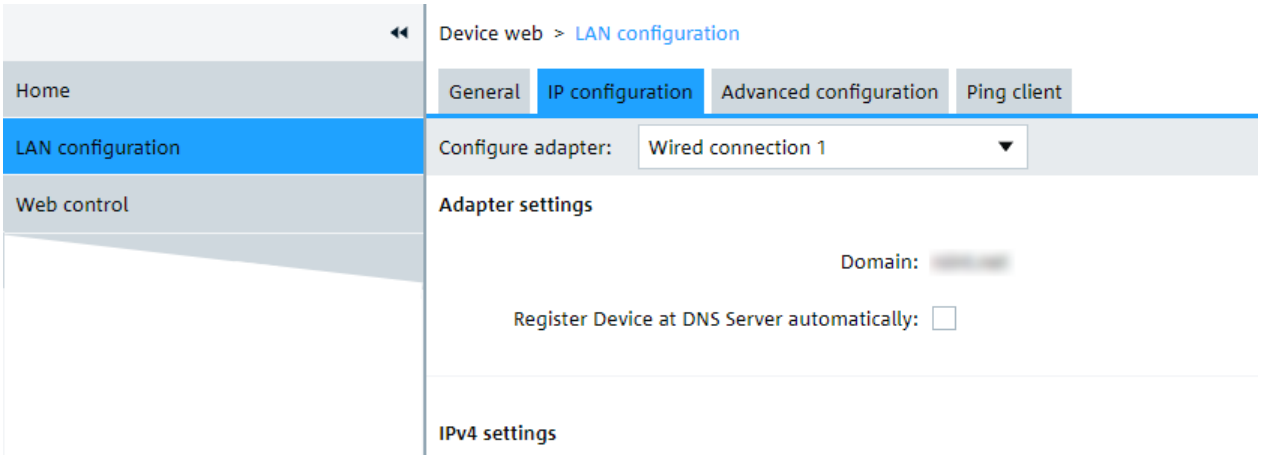
The LAN configuration consists of several parts:

- "General"
- "IP configuration" provides all mandatory LAN parameters.
- "Advanced configuration" provides LAN settings that are not declared mandatory by the standard.
- "Ping client" provides the ping utility to verify the connection between the instrument and other devices.

Changing the LAN parameters requires the "Web password" password, which is set on the instrument.

#### IP configuration

The "LAN Configuration > IP configuration" webpage displays all mandatory LAN parameters and allows their modification. Changing the LAN parameters requires the "Web password" password, which is set on the instrument.



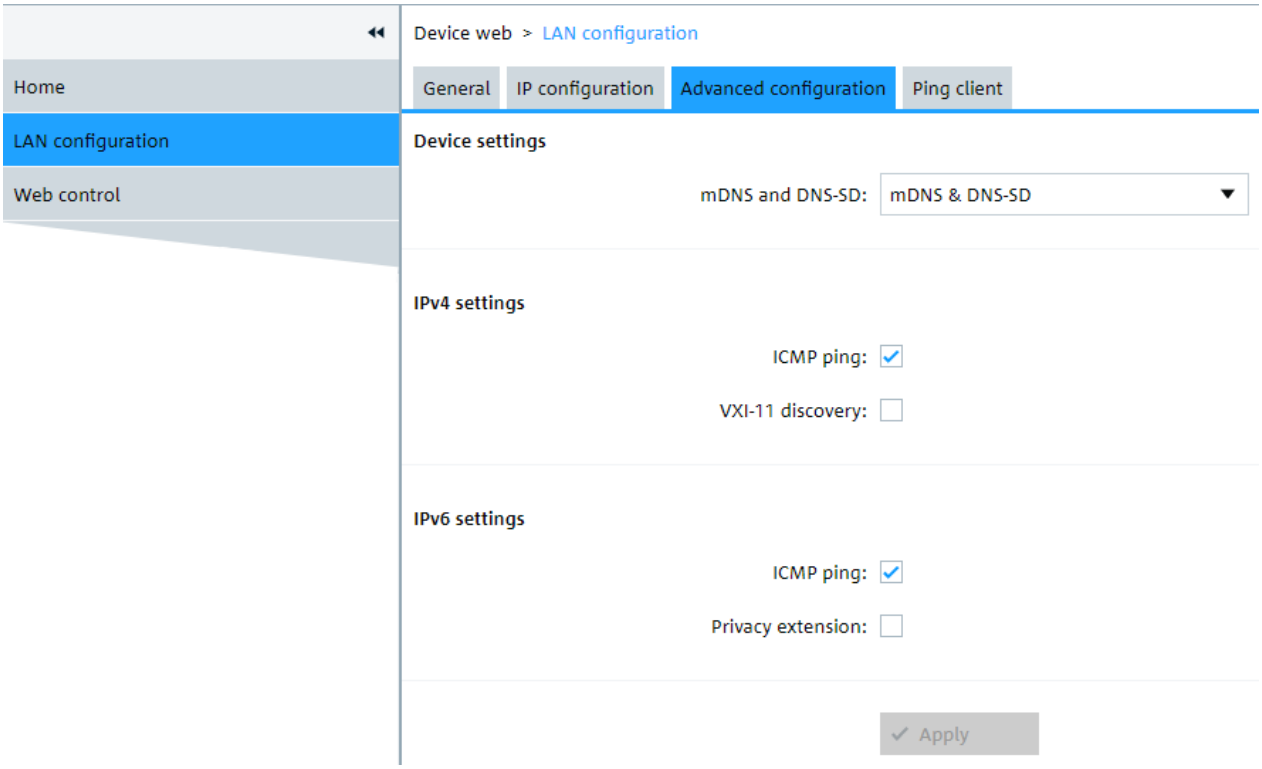

The "IP address mode" controls how the IP address for the instrument gets assigned. For the manual configuration mode, at least the static IP address, subnet mask, and default gateway are used to configure the LAN. The automatic configuration mode uses a DHCP server or Dynamic Link Local Addressing to obtain the instrument IP address.

The same settings are available on the instrument under "Menu" > "Settings" > "System" > "Network" > "Advanced".

### Advanced configuration

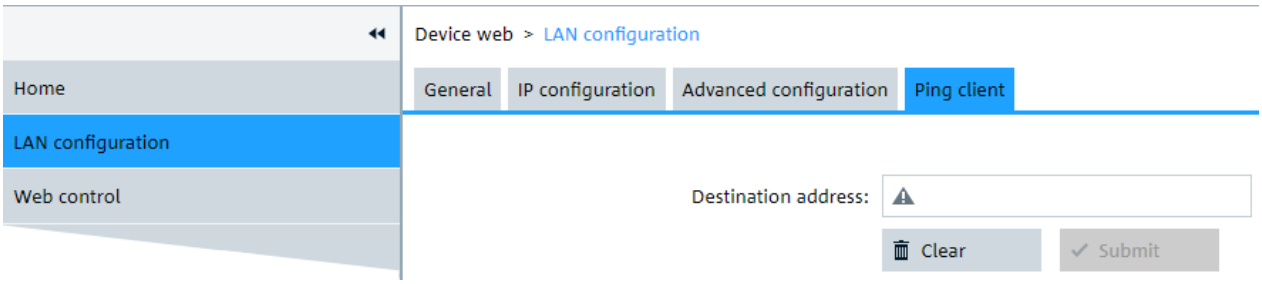
The "LAN Configuration > Advanced configuration" parameters are used as follows:

- mDNS and DNS-SD are two additional protocols: Multicast DNS and DNS Service Discovery. They are used for device communication in zero configuration networks working without DNS and DHCP.
- "ICMP ping" must be enabled to use the ping utility.
- "VXI-11" is the protocol that is used to detect the instrument in the LAN.




### Ping client

Ping is a utility that verifies the connection between the instrument and another device. The ping command uses the `ICMP` echo request and echo reply packets to determine whether the LAN connection is functional. Ping is useful for diagnosing IP network or router failures. The ping utility is not password-protected.




To initiate a ping between the compliant instrument and a second connected device:

1. Enable "ICMP ping" on the "Advanced configuration" page (enabled by default).
2. On the "Ping client" page, enter the IP address of the second device **without the ping command and without any further parameters** into the "Destination address" field (e.g. *11.113.11.203*).
3. Click "Submit".

### 17.2.2.3 Web control

"Web Control" emulates the front panel and shows the instrument display. You see a live image of the instrument, and you can operate the instrument remotely. You can use the keys, the knobs and the menus in the same way as directly on the instrument.

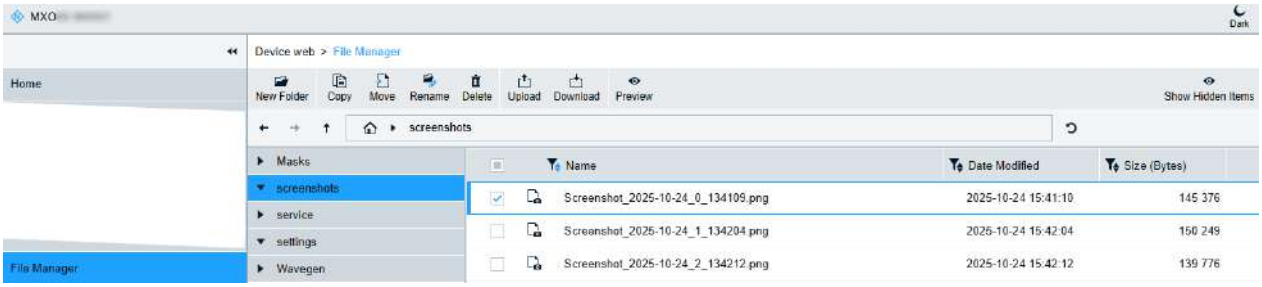



In the upper right corner above the emulated display, you see some options:

- "Front panel" hides or shows the front panel.
- "Scaling" scales the diagram to show it completely. If disabled, the diagram is shown in its original size.
- "Read-only" disables operating, only viewing is possible.
- The camera icon saves a screenshot.
- The arrows icon hides or shows the menu.

### 17.2.2.4 File manager

The file manager on the instrument's webpage provides access to the user data that is stored on the instrument, or on a USB flash device connected to the instrument.




In the browser's file manager, you can organize the data:

- Create folders.
- Copy files within the instrument's file system, or from and to a USB flash device. Multiple files selection is possible.
- Move files within the instrument's file system, or from and to a USB flash device. Multiple files selection is possible.
- Rename files.
- Delete files.
- Upload files from the operating computer to the instrument. Multiple files can be uploaded at once.
- Download files from the instrument to the operating computer. Multiple files selection is possible.
- Preview file content of text files with maximum size of 500 kbyte.

**Notes on USB flash devices**

If no USB flash device is connected to the MXO 3, the "USB" folder is empty but still listed. Do NOT upload files to this main "USB" folder because data will be lost. Upload files only to the subfolders of connected USB flash devices.

After attaching a USB flash device, creating a screenshot or other actions, the file manager might not display the last change:

► Click the "Refresh" icon on the right of the address field to update the display.

The "Eject" button for USB flash devices has no function when operating the instrument remotely.

### 17.2.2.5 Mapping the instrument's file system as a network drive

Using web based distributed authoring and versioning (WebDAV), you can access and exchange files over the internet: move, cut, copy and paste files from the instrument to the computer and vice versa. To prepare file exchange, map the instrument's file system as a network drive in the file explorer of the operating computer:

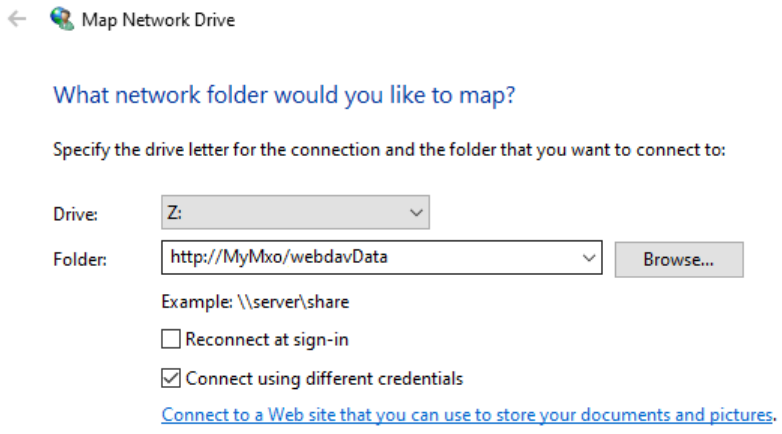
1. Check if the connection status icon in the upper right corner of the instrument's display is green.

   If it is green, the instrument is connected to the LAN.

2. Open the file explorer on the operating computer.
3. In the left list, right-click "Network" or the entry of the operating computer, and select "Map network drive".
4. Enter the path to the instrument's file system:
   `http://<hostname>/webdavData`.

   You can find the host name in the "Device summary", which is shown when you tap the Rohde & Schwarz logo in the upper right corner of the instrument's screen. The default host name is <instrument_model>-<serial_number>.

5. Activate "Connect using different credentials".



6. Click "Finish".
7. Leave the "User name" and "Password" empty. Click "OK".

If there are problems with the connection, check if the "WebClient" service is running: "Task Manager" > "Services" tab. If the "WebClient" service is not running, start it.

## 17.3 Remote operation with VNC client

If the MXO 3 is connected to a computer via LAN, you can operate the instrument from the computer using a VNC client. Therefore, install a VNC client on the computer.

To operate the instrument via VNC client:

1. Start the VNC client.
2. Enter the hostname (device name) of the instrument. You find the name in the instrument information on the oscilloscope. Tap the Rohde & Schwarz logo to show the information.
3. Connect to the instrument.

   You see the display of the oscilloscope and can use the menus, dialogs, SmartGrid and all operating means of the display.

## 17.4 Remote control

Remote control automates the operation of the MXO 3 using SCPI commands, scripts and programs.

For general information on remote control of Rohde & Schwarz products via SCPI, refer to www.rohde-schwarz.com/rc-via-scpi.

The following sections describe the specific basics of remote control. Definitions specified in the SCPI standard are not provided.

SCPI commands are listed and described in Section 18, "Remote control commands", on page 937.

- Remote control interfaces and protocols.....923
- Starting and stopping remote control.....926

### 17.4.1 Remote control interfaces and protocols

The MXO 3 supports the following interfaces for remote control:

- LAN interface with HiSLIP protocol, VXI-11 protocol or raw socket.
  See Section 17.4.1.1, "LAN interface", on page 924.

- USB TMC interface (USB Test & Measurement Class), allows test instruments to be connected to and controlled by a computer via a standard USB cable. A VISA installation is required.
  See Section 17.4.1.2, "USB interface", on page 926.

#### 17.4.1.1 LAN interface

To be integrated in a LAN, the instrument is equipped with a LAN interface, consisting of a connector, a network interface card and protocols. The interface details are given in the specifications.

For remote control via a network, the PC and the instrument must be connected via the LAN interface to a common network with TCP/IP network protocol. They are connected using a commercial RJ-45 cable (shielded or unshielded twisted-pair category 5 or better). The TCP/IP network protocol and the associated network services are preconfigured on the instrument. Software for instrument control and the VISA program library must be installed on the controller.

The MXO 3 supports the following ways of LAN communication:

- HiSLIP protocol (high-speed LAN interface protocol): recommended protocol for remote control via LAN, successor of VXI-11.
- VXI-11 protocol: first LAN-based protocol, developed for test and measurement instruments communication as a replacement for the expensive GPIB equipment.
- Raw socket: universal I/O protocol, usually over the port 5025. No need for VISA installation. Communication without control channel, hence there is no support for events like Service Request (SRQ) or Device Clear (DCL).

**HiSLIP protocol**

The high-speed LAN instrument protocol (HiSLIP) was defined by the IVI Foundation and adopted by the LXI Consortium as the recommended LAN protocol.

The HiSLIP protocol uses two TCP sockets for a single connection - one for fast data transfer, the other for non-sequential control commands (e.g. `Device Clear` or `SRQ`).

HiSLIP has the following characteristics:

- High performance as with raw socket network connections.
- Compatible IEEE 488.2 support for message exchange protocol, device clear, serial poll, remote/local, trigger, and service request.
- It uses a single IANA-registered port (4880), which simplifies the configuration of firewalls.
- It supports simultaneous access of multiple users by providing versatile locking mechanisms.
- Usable for IPv6 or IPv4 networks.

Note that HiSLIP data is sent to the device using the "fire and forget" method with immediate return. A successful return of a VISA operation such as `viWrite()` does not ensure that the instrument has finished or started the requested command. It only indicates that the command has been delivered to the TCP/IP buffers.

### VXI-11 protocol

The VXI-11 standard is based on the open network computing remote procedure call (ONC RPC) protocol, which in turn relies on TCP/IP at the network/transport layer. The TCP/IP network protocol and the associated network services are preconfigured.

TCP/IP ensures connection-oriented communication, where the order of the exchanged messages is adhered to and interrupted links are identified. With this protocol, messages cannot be lost.

### Raw socket mode

With raw socket communication, the instrument communicates directly with the TCP transport layer. A VISA library is not required, but VISA offers an option to open a raw socket session. Service requests and status byte polling are not supported in the raw socket mode.

TCP/IP raw socket connections for SCPI commands use port 5025. Terminate all commands with a line feed (LF).

### IP address (VISA resource string) for LAN connections

Only the IP address or a valid DNS host name is required to set up the connection. The host address is part of the "VISA resource string" used by programs to identify and control the target application. The VISA resource string has the form:

- **HiSLIP protocol**
  `TCPIP::<host address>::hislip0[,<port>][::INSTR]`
  *Examples*:
  IP address is *192.1.2.3*, resource string: `TCPIP::192.1.2.3::hislip0`
  Instrument name is *RSRT1*, resource string: `TCPIP::RSRT1::hislip0`.
  DNS host name is *MXO 3-123456*, resource string:
  `TCPIP::MXO 3-123456::hislip0`.
- **VXI-11 protocol**
  `TCPIP::<host address>[::inst0][::INSTR]`
  *Examples*
  IP address is *192.1.2.3*, resource string: `TCPIP::192.1.2.3`
  Instrument name is *RSRT1*, resource string: `TCPIP::RSRT1`.
  DNS host name is *MXO 3-123456*, resource string: `TCPIP::MXO 3-123456`.
- **Raw socket communication**
  `TCPIP::<host address>::<port>::SOCKET`
  *Example*
  TCPIP::192.1.2.3::5025::SOCKET

Where:

- `host address` identifies the MXO 3 in the network, usually the IP address. If the LAN is supported by a DNS server, the host name can be used instead of the IP address. The DNS server (Domain Name System server) translates the host name to the IP address.
- `hislip0` indicates the HiSLIP protocol.

- `inst0` is the default LAN device name. VISA supports several devices running on the instrument. On MXO 3, only one device is configured, so the LAN device name can be omitted.
- `INSTR` specifies a VISA resource of the type INSTR. By default, the VISA resource name control is set to the INSTR class.
- `port` is the port number, for raw socket communication, port 5025 is used.

#### 17.4.1.2 USB interface

For remote control using USB TMC, the controller PC and the instrument are connected over the USB type B interface.

A VISA installation is required, because it contains the necessary USB drivers to be able to connect and communicate with the instrument.

**USB resource string**

`USB::<vendor ID>::<product ID>::<serial number>[::INSTR]`

- <vendor ID> is the manufacturer ID for Rohde & Schwarz.
- <product ID> is the product identification of the instrument.
- <serial number> is the individual serial number of the instrument.
- [::INSTR] indicates the instrument resource class (optional).

### 17.4.2 Starting and stopping remote control

#### 17.4.2.1 Starting a remote control session

When you switch on the instrument or start the firmware, it is always in manual operation state ("local" state).

► To start the remote control:

- Send a command from the controller.
- VXI-11 protocol (LAN or USB interface): Use `&GTR` interface message.

While remote control is active, the settings are optimized for maximum measurement speed; the display is switched off.

Operation via the front panel is disabled.

On the screen, two buttons appear in the upper left corner: "Local" and "View".

#### 17.4.2.2 Using the display during remote control

You can observe the screen while a remote control script is executed. Displaying the results of script execution is helpful for program test purposes but tends to slow down

the measurement. Therefore it is recommended that you switch off the display in real measurement applications where a tested program script is to be executed repeatedly.

► To switch on the display, do one of the following:

- Tap the "View" button in the upper left corner of the screen.
- Use the `SYSTem:DISPlay:UPDate ON` command.

► To switch off the display, do one of the following:

- Tap the "View" button again.
- Use the `SYSTem:DISPlay:UPDate OFF` command.

#### 17.4.2.3 Returning to manual operation

The MXO 3 switches back to manual operation when the remote connection is closed. Besides, you can return to manual operation manually or via remote control.

► To return to manual operation:

- Tap the "Local" button in the upper left corner of the screen.
- VXI-11 protocol: Use `&GTL` interface message.

## 17.5 Remote control - status reporting system

The status reporting system stores all information on the current operating state of the instrument, and on errors that have occurred. This information is stored in the status registers and in the error queue. Both can be queried via `STATus...` commands.

### 17.5.1 Hierarchy of status registers

As shown in the following figure, the status information is of hierarchical structure.

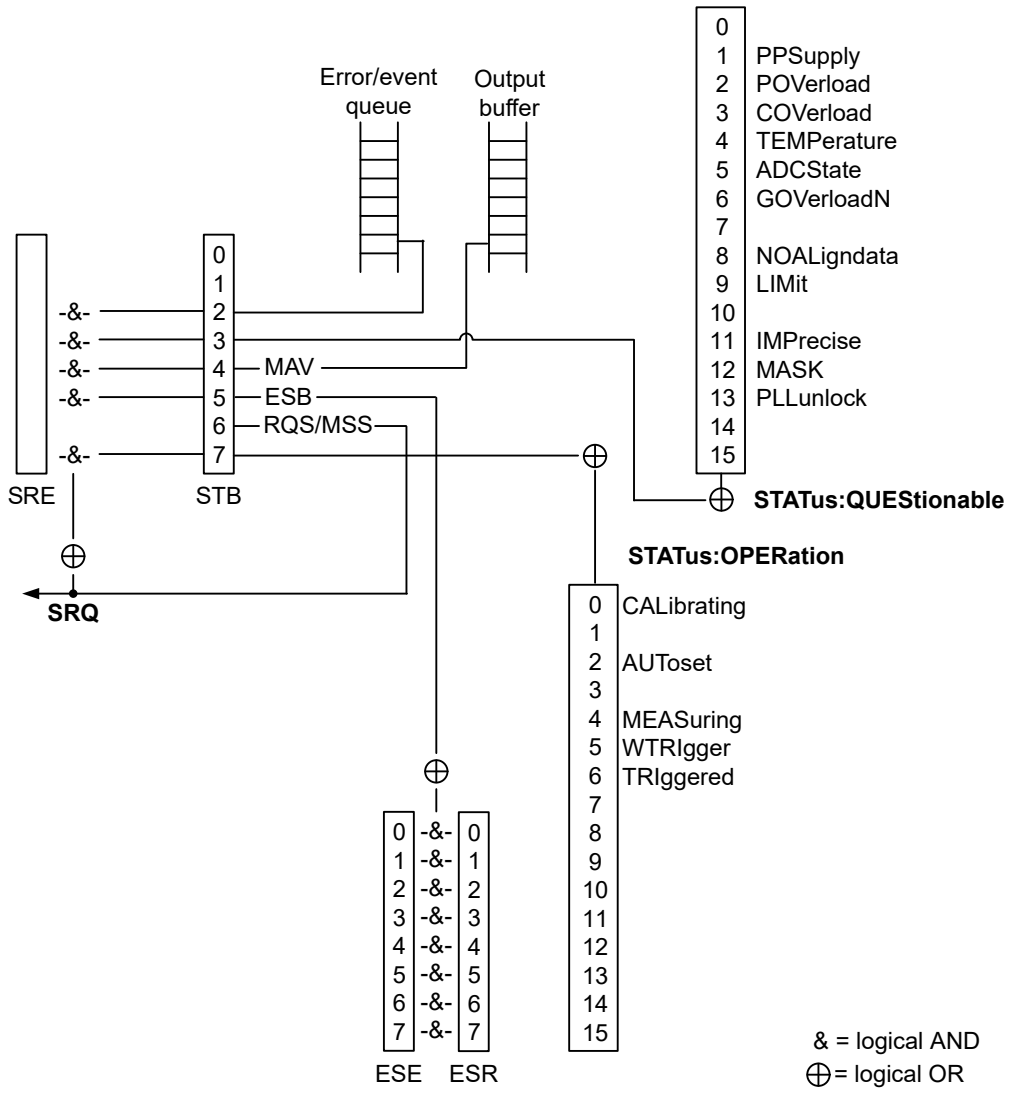



*Figure 17-1: Overview of the status registers hierarchy*

- **STB, SRE**
  The STatus Byte (`STB`) register and its associated mask register Service Request Enable (`SRE`) form the highest level of the status reporting system. The `STB` provides a rough overview of the instrument status, collecting the information of the lower-level registers.
- **ESR and SCPI registers**
  The STB receives its information from the following registers:
  - The Event Status Register (`ESR`) with the associated mask register standard Event Status Enable(`ESE`).
  - The `STATus:OPERation` and `STATus:QUEStionable` registers are defined by SCPI and contain detailed information on the instrument.
- **Output buffer**

The output buffer contains the messages that the instrument returns to the controller. It is not part of the status reporting system but determines the value of the MAV bit in the STB and thus is represented in the overview.

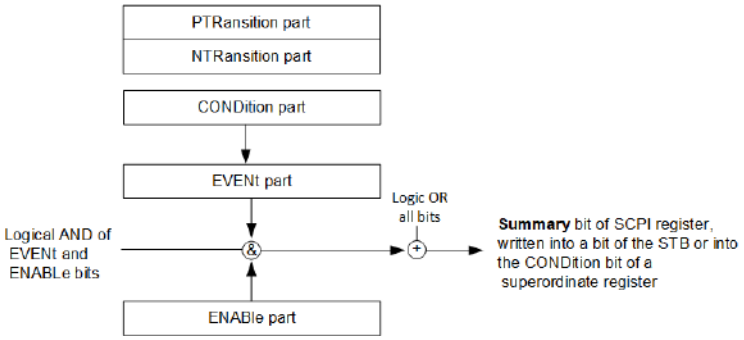
All status registers have the same internal structure.

### 17.5.2 Structure of a SCPI status register

The STATus:OPERation and the STATus:QUEStionable SCPI status registers consists of five parts.

The individual bits are independent of each other, i.e. each hardware status is assigned a bit number, which is valid for all five parts. Bit 15 (the most significant bit) is set to zero for all parts. Thus, the contents of the register parts can be processed by the controller as positive integers.

**Description of the five status register parts**




The five parts of a SCPI status register have different properties and functions:

- **CONDition**
  The CONDition part reflects the current instrument status. For summary bits, the CONDition bit mirrors the sum of the enabled EVENt bits of the next lower register.
  This register part can only be read, but not written into or cleared. Its contents are not affected by reading.
- **PTRansition / NTRansition**
  The two transition register parts define which state transition of the CONDition part (none, 0 to 1, 1 to 0 or both) is stored in the EVENt part.
  The **Positive-TRansition** part acts as a transition filter. When a bit of the CONDition part is changed from 0 to 1, the associated PTR bit decides whether the EVENt bit is set to 1.
  - PTR bit =1: the EVENt bit is set.
  - PTR bit =0: the EVENt bit is not set.

  All positive transitions are enabled by default.
  This part can be written into and read as required. Its contents are not affected by reading.

The **Negative-TRansition** part also acts as a transition filter. When a bit of the CONDition part is changed from 1 to 0, the associated NTR bit decides whether the EVENt bit is set to 1.

– NTR bit =1: the EVENt bit is set.

– NTR bit =0: the EVENt bit is not set.

All negative transitions are disabled by default.
This part can be written into and read as required. Its contents are not affected by reading.

- **EVENt**
  The EVENt part indicates whether an event has occurred since the last reading. It is the "memory" of the condition part. It allows you to detect the occurrence of events reliably, even if they have changed the state of the CONDition bit for a very brief interval. It only indicates events passed on by the transition filters.
  This part can only be read by the user. **Reading the register clears it.**
- **ENABle**
  The ENABle part determines whether the associated EVENt bit contributes to the summary bit (see below). Each bit of the EVENt part is "ANDed" with the associated ENABle bit (symbol '&'). The results of all logical operations of this part are passed on to the summary bit via an "OR" function (symbol '+').
  ENABle bit = 0: the associated EVENt bit does not contribute to the summary bit
  ENABle bit = 1: if the associated EVENt bit is "1", the summary bit is set to "1" as well.
  This part can be written into and read by the user as required. Its contents are not affected by reading.

## 17.5.3 Contents of the status registers

### 17.5.3.1 Status byte (STB) and service request enable register (SRE)

The STatus Byte (STB) is already defined in IEEE 488.2. It provides a rough overview of the instrument status by collecting the pieces of information of the lower registers. A special feature is that bit 6 acts as the sum bit of the remaining bits of the status byte.

The STB can thus be compared with the CONDition part of an SCPI register and assumes the highest level within the SCPI hierarchy.

The STB is read using the command *STB? or a serial poll.

The STatus Byte (STB) is linked to the Service Request Enable (SRE) register. Each bit of the STB is assigned a bit in the SRE. Bit 6 of the SRE is ignored. If a bit is set in the SRE and the associated bit in the STB changes from 0 to 1, a service request (SRQ) is generated. The SRE can be set using the command *SRE and read using the command *SRE?.

*Table 17-1: Meaning of the bits used in the status byte*

| Bit No. | Meaning |
|---|---|
| 0...1 | Not used |
| 2 | Error Queue not empty<br>The bit is set when an entry is made in the error queue. If this bit is enabled by the SRE, each entry of the error queue generates a service request. Thus an error can be recognized and specified in greater detail by polling the error queue. The poll provides an informative error message. This procedure is to be recommended since it considerably reduces the problems involved with remote control. |
| 3 | QUEStionable status register summary bit<br>The bit is set if an `EVENt` bit is set in the `QUEStionable` status register and the associated `ENABle` bit is set to 1. A set bit indicates a questionable instrument status, which can be specified in greater detail by querying the `STATus:QUEStionable` status register. |
| 4 | MAV bit (message available)<br>The bit is set if a message is available in the output queue which can be read. This bit can be used to enable data to be automatically read from the instrument to the controller. |
| 5 | ESB bit<br>Sum bit of the event status register. It is set if one of the bits in the event status register is set and enabled in the event status enable register. Setting of this bit indicates a serious error which can be specified in greater detail by polling the event status register. |
| 6 | MSS bit (main status summary bit)<br>The bit is set if the instrument triggers a service request. This is the case if one of the other bits of this registers is set together with its mask bit in the service request enable register SRE. |
| 7 | `STATus:OPERation` status register summary bit<br>The bit is set if an `EVENt` bit is set in the `OPERation` status register and the associated `ENABle` bit is set to 1. A set bit indicates that the instrument is just performing an action. The type of action can be determined by querying the `STATus:OPERation` status register. |

### 17.5.3.2 Event status register (ESR) and event status enable register (ESE)

The ESR is defined in IEEE 488.2. It can be compared with the `EVENt` part of a SCPI register. The event status register can be read out using command `*ESR?`.

The ESE corresponds to the `ENABle` part of a SCPI register. If a bit is set in the ESE and the associated bit in the ESR changes from 0 to 1, the ESB bit in the STB is set.

The ESE register can be set using the command `*ESE` and read using the command `*ESE?`.

*Table 17-2: Meaning of the bits used in the event status register*

| Bit No. | Meaning |
|---|---|
| 0 | Operation complete<br>This bit is set on receipt of the command `*OPC` exactly when all previous commands have been executed. |
| 1 | Not used |

| Bit No. | Meaning |
|---|---|
| 2 | Query error<br>This bit is set if the controller wants to read data from the instrument without having sent a query. It is also set if the controller does not fetch requested data and sends new instructions to the instrument instead. The cause is often a query that is faulty and hence cannot be executed. |
| 3 | Device-dependent error<br>This bit is set if a device-dependent error occurs. An error message with a number between -300 and -399 or a positive error number, which denotes the error in greater detail, is entered into the error queue. |
| 4 | Execution error<br>This bit is set if a received command is syntactically correct but cannot be performed for other reasons. An error message with a number between -200 and -300, which denotes the error in greater detail, is entered into the error queue. |
| 5 | Command error<br>This bit is set if a command is received, which is undefined or syntactically incorrect. An error message with a number between -100 and -200, which denotes the error in greater detail, is entered into the error queue. |
| 6 | User request<br>This bit is set when the instrument is switched over to manual control. |
| 7 | Power on (supply voltage on)<br>This bit is set when you turn on the instrument. |

#### 17.5.3.3 STATus:OPERation register

In the `CONDition` part, this register contains information on which actions the instrument is executing. In the `EVENt` part, it contains information on which actions the instrument has executed since the last reading.

It can be read using the commands `STATus:OPERation:CONDition?` or `STATus:OPERation[:EVENt]?`, see also Section 18.21.2, "STATus:OPERation register", on page 2046.

***Table 17-3: Bits in the STATus:OPERation register***

| Bit No. | Meaning |
|---|---|
| 0 | `CALibrating`<br>This bit is set as long as the instrument is performing a self-alignment. |
| 1 | Not used |
| 2 | `AUToset`<br>This bit is set while the instrument is performing an auto setup. |
| 3 | Not used |
| 4 | `MEASuring`<br>The bit is set as long as an acquisition - sampling and postprocessing - is running. In run continuous mode, the bit is always set. |

| Bit No. | Meaning |
|---|---|
| 5 | WTRIgger<br>Available only for single acquisitions or the first acquisition of continuous acquisitions.<br>The wait for trigger status bit indicates that the instrument is ready to trigger, and the pre-trigger time is expired.<br>The bit is set for as long as a measurement is running. |
| 6 | TRIggered<br>Available only for single acquisitions or the first acquisition of continuous acquisitions.<br>The triggered status bit indicates that the instrument has been triggered. |
| 7 - 15 | Not used |

#### 17.5.3.4 STATus:QUEStionable register

This register contains specific information on instrument operation and signal reliability. It can be read using the commands `STATus:QUEStionable:CONDition?` and `STATus:QUEStionable[:EVENt]?`.

The remote commands for the STATus:QUEStionable register are described in Section 18.21, "Status reporting", on page 2045.

***Table 17-4: Bits in the STATus:QUEStionable register***

| Bit No. | | Meaning |
|---|---|---|
| 0 | - | For future use. |
| 1 | PPSupply | This bit is set if probe power supply overload occurs. |
| 2 | POVerload | This bit is set if a probe overload occurs. |
| 3 | COVerload | This bit is set if a questionable channel overload occurs. |
| 4 | TEMPerature | This bit is set if a questionable temperature occurs. |
| 5 | ADCState | The bit is set if the signal is clipped on the upper or lower edge of the screen overflow of the ADC occurs. |
| 6 | GOVerload | This bit is set if a generator overload occurs. |
| 7 | - | For future use. |
| 8 | NOALigndata | This bit is set if no alignment data is available - the instrument is uncalibrated. |
| 9 | LIMit | For future use. |
| 10 | - | For future use. |
| 11 | IMPRecise | This bit is set if the magnitude of the signal is too low to get reliable measurement results. |
| 12 | MASK | This bit is set if a mask is violated. |

| Bit No. | | Meaning |
|---|---|---|
| 13 | PLLunlock | This bit is set if a phase-locked loop process is unlocked. There are two reasons:<br>● external reference is missing or defect<br>● hardware defect of internal PLL |
| 14 | - | For future use. |

#### 17.5.3.5 Application of the status reporting system

The purpose of the status reporting system is to monitor the status of one or several instruments in a test system. In particular, it monitors continuous measurements to detect errors in the signal. The controller must receive and evaluate the information of all devices.

The following standard methods are used:

- **Service request**: An SRQ is initiated by the instrument.
- Query of a **specific register** by commands
- Query of the **error queue**

These methods are described in the following sections.

##### Service request

The instrument can send a service request (SRQ) to the controller. Usually, this service request initiates an interrupt at the controller, to which the control program can react appropriately.

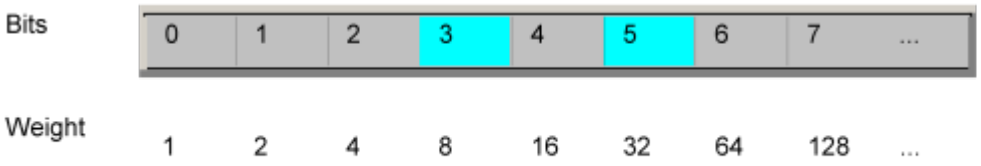
An SRQ is initiated if one or several of bits 2, 3, 4, 5 or 7 of the STB are set and enabled in the SRE. Each of these bits combines the information of a lower-level register, the error queue or the output buffer.

The ENABle parts of the status registers can be set such that arbitrary bits in an arbitrary status register initiate an SRQ. To use service requests effectively, set all bits to "1" in the mask registers SRE and ESE.

The SRQ is the only possibility for the instrument to become active on its own. Configure the instrument so that it initiates a service request if errors occur. And ensure that your program reacts appropriately to service requests.

##### Query of a register

The STB and ESR registers contain 8 bits, the SCPI registers 16 bits. The contents of a status register are specified and transferred as a single decimal number. Therefore, each bit is assigned a weighted value. The decimal number is calculated as the sum of the weighted values of all bits in the register that are set to 1.




**Example:**

The decimal value 40 = 32 + 8 indicates that bits no. 3 and 5 in the status register (e.g. the `QUEStionable` status summary bit and the `ESB` bit in the `STatus` byte) are set.

### 17.5.4 Error queue

Each error state in the instrument leads to an entry in the error queue.

The error queue should be queried after every SRQ in the controller program as the entries describe the cause of an error more precisely than the status registers. Especially in the test phase of a controller program the error queue should be queried regularly since faulty commands from the controller to the instrument are recorded there as well.

### 17.5.5 Reset values of the status reporting system

The following table contains the commands and events that reset the status reporting system. Only `*RST` and `SYSTem:PRESet` change the functional instrument settings, all others do not.

`DCL` means Device clear and is a GPIB interface command. The command aborts the processing of the commands received and sets the command processing software to a defined initial state and does not change the instrument settings. In a VISA implementation, viClear translates to the DCL interface command.

All other commands are explained in the "Remote control commands" section.

***Table 17-5: Resets of the status reporting system***

| **Event** | `DCL` **(GPIB)** **viClear (VISA)** **Device clear** | `*RST` **or** `SYSTem:PRESet` | `STATus:PRESet` | `*CLS` **Clear status** |
|---|---|---|---|---|
| Clear STB, ESR | - | - | - | yes |
| Clear SRE, ESE | - | - | - | - |
| Clear EVENt parts of the registers | - | - | - | yes |
| Clear ENABle parts of all OPERation and QUEStionable registers;<br>Fill ENABle parts of all other registers with "1". | - | - | yes | - |

| **Event** | `DCL` **(GPIB)**<br>**viClear (VISA)**<br>**Device clear** | `*RST` **or** `SYSTem:PRESet` | `STATus:PRESet` | `*CLS`<br>**Clear status** |
|---|---|---|---|---|
| Fill PTRansition parts with "1";<br>Clear NTRansition parts | - | - | yes | - |
| Clear error queue | - | - | - | yes |
| Clear output buffer | yes | 1) | 1) | 1) |
| Clear command processing and input buffer | yes | - | - | - |
| 1) The first command in a command line that immediately follows a <PROGRAM MESSAGE TERMINATOR> clears the output buffer. | | | | |

# 18 Remote control commands

This section describes all remote commands available for MXO 3 and provides examples and information how to use the commands.

For general information on remote control of Rohde & Schwarz products via SCPI, refer to www.rohde-schwarz.com/rc-via-scpi.

See also Section 17.4, "Remote control", on page 923 and Section 17.5, "Remote control - status reporting system", on page 927 for further information on remote control.

The descriptions of the remote commands assume a MXO 3 with the highest available bandwidth and number of channels. Depending on your model, some of the available command ranges can differ from the values stated in the following sections.

## 18.1 Conventions used in remote command description

The basics of SCPI command structure, syntax and command parameters are explained in www.rohde-schwarz.com/rc-via-scpi.

The following conventions are used in the remote command descriptions:

- *Command usage*
  If not specified otherwise, commands can be used both for setting and for querying parameters.
  If a command can be used for setting or querying only, or if it initiates an event, the usage is stated explicitly.
- *Parameter usage*
  If not specified otherwise, a parameter can be used to set a value, and it is the result of a query.
  Parameters required only for setting are indicated as "Setting parameters".
  Parameters required only to refine a query are indicated as "Query parameters".
  Parameters that are only returned as the result of a query are indicated as "Return values".
- *Conformity*
  Commands that are taken from the SCPI standard are indicated as "SCPI confirmed". All commands used by the MXO 3 follow the SCPI syntax rules.
- *Asynchronous commands*
  A command which does not automatically finish executing before the next command starts executing (overlapping command) is indicated as an "Asynchronous command".
- *Reset values (*RST)*
  Default parameter values that are used directly after resetting the instrument (`*RST` command) are indicated as "*RST" values, if available.
- *Default unit*
  The default unit is used for numeric values if no other unit is provided with the parameter.

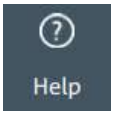
## 18.2 Finding the appropriate command

In the following sections, the commands are sorted according to the menu and dialog structure.

A list of all commands in alphabetical order is given in the "List of Commands" at the end of this documentation.

To find the appropriate command for a setting easily, you can use the context help:

1. Enable the "Help" icon on the toolbar.




2. Tap the parameter for which you need information.

   The "Help" window opens and displays the comprehensive description and the corresponding remote command.

3. Tap the remote command link to open the command description.

## 18.3 Error messages on remote commands

The systerm returns error codes and messages if it cannot execute a command, or if there is no valid result.

| Error message | Description |
|---|---|
| -131, "Invalid suffix" | The measurement with the used suffix is not configured, and similar situations. |
| -221, "Settings conflict“ | The command is not available with current settings. If you change the related settings, the command might work. |
| -394; "Overload detected" | The result is not trustworthy because the data is not in the ADC range. |
| -395; "Value not set" | Message is returned, e.g. for a measurement results when the acquisition isn’t complete. It is the default for results after preset or when relevant settings have changed. |
| -396; "Invalid data" | The acquisition is complete, but some requirements are not met that are required to calculate the result. |
| -397; "Not available" | The command is not available due to hardware restrictions, missing option, inappropriate probe is connected, and other misfits. |
| -398; "Operation aborted" | |
| -399; "Operation failed" | |

# 18.4 Frequently used parameters and suffixes

This chapter describes in general those parameters and suffixes that are used in several subsystems.

## 18.4.1 Waveform parameter

Many commands require one of the waveforms to be specified as source. The following table lists all waveforms. For each command using a waveform parameter, the available waveforms are specified in the command description.

| Waveform | Description |
|---|---|
| C1 \| C2 \| ... \| C8<br>CHAN1 = C1, CHAN2 = C2, CHAN3 = C3, CHAN4 = C4 , CHAN5 = C5, CHAN6 = C6, CHAN7 = C7, CHAN8 = C8 | Analog channel waveforms |
| M1 \| M2 \| ... \| M8 | Math waveforms |
| R1 \| R2 \| ... \| R8 | Reference waveforms |
| XY1 \| XY2 \| XY3 \| XY4 | XY-waveforms |
| SBUS1 \| SBUS2 \| SBUS3 \| SBUS4 | Serial buses |
| D0 \| D1 \| D2 \| ... \| D14 \| D15 | Digital channels |
| PBUS1 \| PBUS2 \| PBUS3 \| PBUS4 | Digital buses |
| TRK1 \| TRK2 \| TRK3 \| ... \| TRK16 | Track waveforms |
| SPECMAXH1 \| SPECMINH1 \| SPECNORM1 \| SPECAVER1 \| SPECMAXH2 \| SPECMINH2 \| SPECNORM2 \| SPECAVER2 \| SPECMAXH3 \| SPECMINH3 \| SPECNORM3 \| SPECAVER3 \| SPECMAXH4 \| SPECMINH4 \| SPECNORM4 \| SPECAVER4 | Spectrum traces:<br>SPECMAXH : Spectrum max hold<br>SPECMINH: Spectrum min hold<br>SPECNORM: Spectrum normal<br>SPECAVER: Spectrum average |
| PA1QPOWER \| PA2QPOWER \| PA3QPOWER | Power waveforms of power quality analysis |
| PA1HPOWER1 \| PA2HPOWER1 \| PA3HPOWER1\| | Power waveforms of power harmonics analysis, only available for standards EN 61000-3-2 Class C and EN 61000-3-2 Class D |
| PA1SPOWER \| PA2SPOWER \| PA3SPOWER | Power waveforms of switching loss analysis |
| PA1IPOWER \| PA2IPOWER \| PA3IPOWER | Input power waveforms of power efficiency analysis |
| PA1OPOWER1 \| PA2OPOWER1 \| PA3OPOWER1 \| PA1OPOWER2 \| PA2OPOWER2 \| PA3OPOWER2 \| PA1OPOWER3 \| PA2OPOWER3 \| PA3OPOWER3 | Output power waveforms of power efficiency analysis |
| PA1TOPOWER \| PA2TOPOWER \| PA3TOPOWER | Total power waveforms of efficieny analysis, requires more than 4 channels |
| PA1SOA \| PA2SOA \| PA3SOA | Waveform of power safe operating area (SOA) analysis |

| Waveform | Description |
|---|---|
| PTHREE1QPOWER1 \| PTHREE2QPOWER1 \| PTHREE3QPOWER1 \| PTHREE1QPOWER2 \| PTHREE2QPOWER2 \| PTHREE3QPOWER2 \| PTHREE1QPOWER3 \| PTHREE2QPOWER3 \| PTHREE3QPOWER3 \| PTHREE1QTPOWER \| PTHREE2QTPOWER \| PTHREE3QTPOWER | Power waveforms of 3-phase power quality analysis |
| PTHREE1HPOWER1 \| PTHREE2HPOWER1 \| PTHREE3HPOWER1 \| PTHREE1HPOWER2 \| PTHREE2HPOWER2 \| PTHREE3HPOWER2 \| PTHREE1HPOWER3 \| PTHREE2HPOWER3 \| PTHREE3HPOWER3 \| PTHREE1HTPOWER \| PTHREE2HTPOWER \| PTHREE3HTPOWER | Power waveforms of 3-phase power harmonics analysis, only available for standards EN 61000-3-2 Class C and EN 61000-3-2 Class D |
| PTHREE1EIP1\| PTHREE2EIP1\| PTHREE3EIP1\| PTHREE4EIP1\| PTHREE5EIP1\| PTHREE6EIP1\| PTHREE1EIP2\| PTHREE2EIP2\| PTHREE3EIP2\| PTHREE4EIP2\| PTHREE5EIP2\| PTHREE6EIP2\| PTHREE1EIP3\| PTHREE2EIP3\| PTHREE3EIP3\| PTHREE4EIP3\| PTHREE5EIP3\| PTHREE6EIP3\| PTHREE1EITP\| PTHREE2EITP\| PTHREE3EITP\| PTHREE4EITP\| PTHREE5EITP\| PTHREE6EITP | Input power waveforms of 3-phase power efficiency analysis |
| PTHREE1EOP1\| PTHREE2EOP1\| PTHREE3EOP1\| PTHREE4EOP1\| PTHREE5EOP1\| PTHREE6EOP1\| PTHREE1EOP2\| PTHREE2EOP2\| PTHREE3EOP2\| PTHREE4EOP2\| PTHREE5EOP2\| PTHREE6EOP2\| PTHREE1EOP3\| PTHREE2EOP3\| PTHREE3EOP3\| PTHREE4EOP3\| PTHREE5EOP3\| PTHREE6EOP3\| PTHREE1EOTP\| PTHREE2EOTP\| PTHREE3EOTP\| PTHREE4EOTP\| PTHREE5EOTP\| PTHREE6EOTP | Output power waveforms of 3-phase power efficiency analysis |
| FREF1 \| FREF2 \| FREF3 \| FREF4 | References of frequency analysis waveforms |

### 18.4.2 Slope parameter

The slope parameter is used with several trigger and search condition commands.

| Slope | Description |
|---|---|
| POSitive | Rising edge, which is a positive voltage change. |
| NEGative | Falling edge, which is a negative voltage change. |
| EITHer | Rising as well as the falling edge. |

### 18.4.3 Polarity parameter

The polarity parameter is used with several trigger and search condition commands.

| Polarity | Description |
|---|---|
| POSitive | Positive going pulses. |
| NEGative | Negative going pulses. |
| EITHer | Both positive and negative going pulses. |

## 18.5 Programming examples

### 18.5.1 SmartGrid layout with zoom

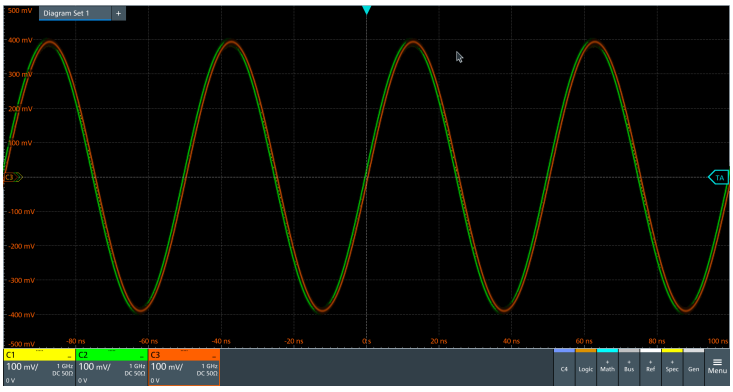
The following example uses 3 sine waves, all with frequency 20 MHz and an amplitude of ±0.4 V. The vertical scale is 100 mV/div, the time scale is 20 ns/div.

In the example, 2 layouts are created, waveforms are assigned, and zoom and spectrum are added.

For a description of the commands, see:

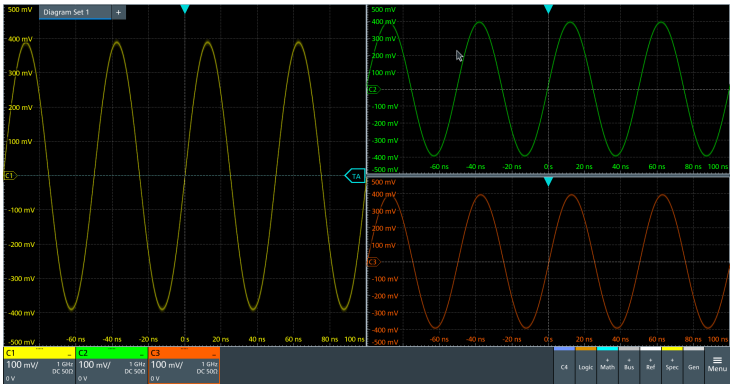
- Section 18.8.3, "SmartGrid", on page 957
- Section 18.12.1, "Zoom", on page 1089
- Section 18.16.1, "Spectrum setup", on page 1190

```
//Prepare: 3 channel signals.
//Enable C1, C2 and C3.
CHAN1:STAT 1
CHAN2:STAT 1
CHAN3:STAT 1
```

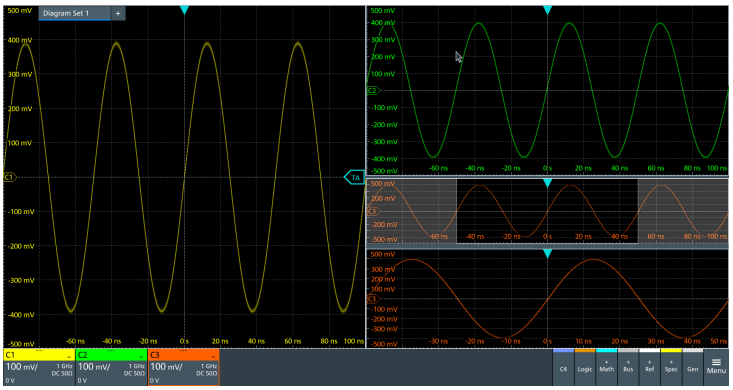



```
//Create new diagrams for C2 and C3.
//Assign sources C2 and C3 to these diagrams.
LAY:DIAG2:ENAB 1
LAY:DIAG2:SOUR C2
LAY:DIAG3:ENAB 1
LAY:DIAG3:SOUR C3
//Create new node (node 2) and assign new diagrams to that node
```

```
//Place C2 on top, C3 at bottom
LAY:NODE2:ENAB 1
LAY:NODE2:CHIL1:CONT:TYPE DIAG
LAY:NODE2:CHIL1:CONT:ID 2
LAY:NODE2:CHIL2:CONT:TYPE DIAG
LAY:NODE2:CHIL2:CONT:ID 3
LAY:NODE2:STYP VERT
//Insert node 2 into the base node (initial node)
//Diagram1 is initially located at child 1
LAY:NODE1:CHIL2:CONT:TYPE NODE
LAY:NODE1:CHIL2:CONT:ID 2
//Place both nodes side by side
LAY:NODE1:STYP HOR
```



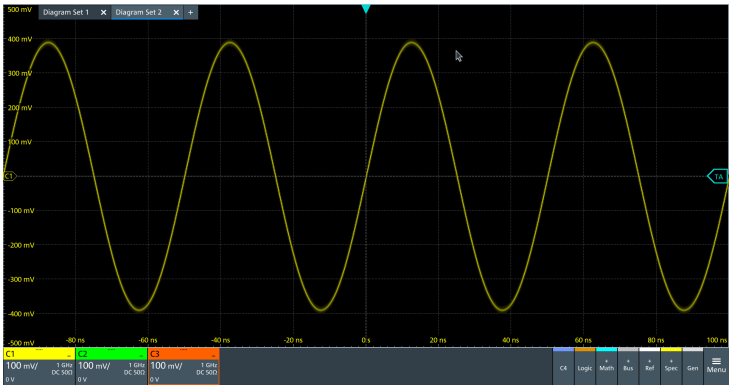
```
//Add zoom to C3. The zoom is located in diagram 3.
LAY:ZOOM:ENAB 1
LAY:ZOOM:SOUR 3
```



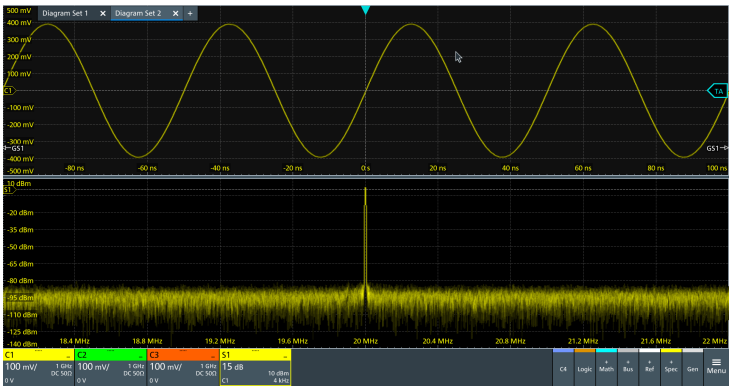
```
//Add a second layout with C1 and spectrum of C1.
//By default, the new layout is empty.
LAY2:ENAB 1
```



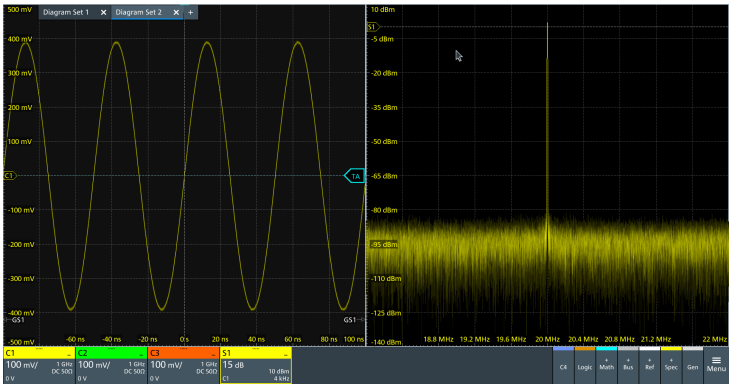
```
//Assign C1 to layout 2.
LAY2:DIAG1:SOUR C1
```



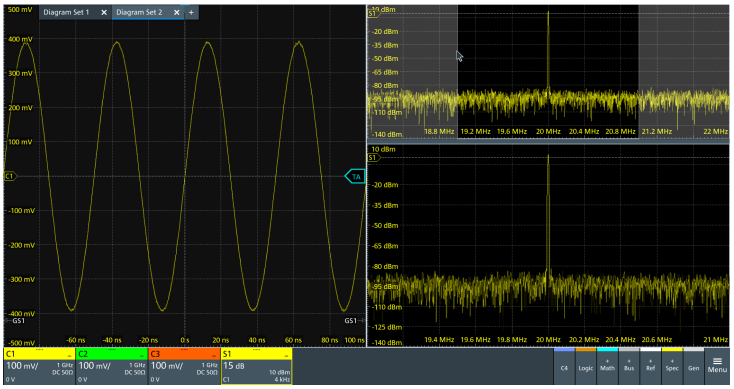
```
//Turn on spectrum of C1.
//By default, the spectrum is set below its analog source, and into the next
//free diagram (here diagram 2).
CALC:SPEC:STAT 1
CALC:SPEC:FREQ:CENT 20e6
CALC:SPEC:FREQ:SPAN 4e6
CALC:SPEC:FREQ:SCAL 15
```



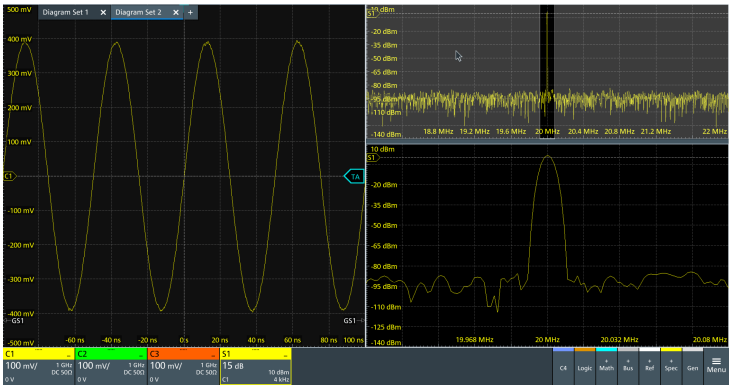
```
//Set C1 and spectrum side by side by changing the split type.
LAY2:NODE1:STYP HOR
```



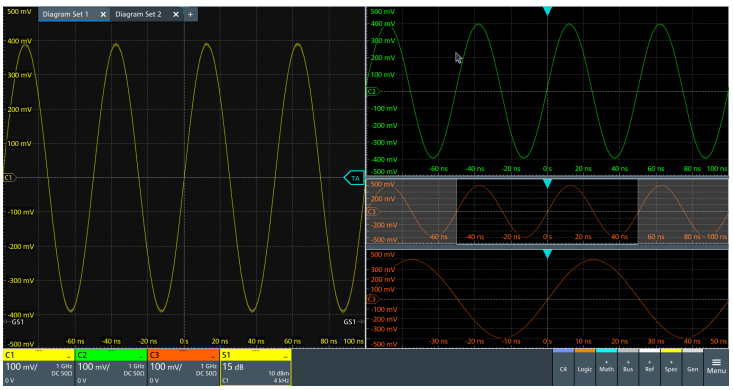
```
//Add zoom to spectrum. The zoom is located in layout 2, in diagram 2.
LAY2:ZOOM:ENAB 1
LAY2:ZOOM:SOUR 2
```



```
//Change zoom area.
LAY2:ZOOM:HOR:MODE REL
LAY2:ZOOM:HOR:REL:START 48
LAY2:ZOOM:HOR:REL:STOP 52
```



```
//Switch back to first layout (Tab 1).
LAY1:SACT
```



## 18.5.2 Creating a mask

The example creates a segment in mask 1 and defines the positions of the corner points.

```
:MTESt:ADD                      //Create mask 1
:MTESt:SEGMent:ADD              //Create segment 1 in mask 1
:MTESt:SEGMent:POINt:ADD        //Add 4 points at coordinate 0;0
:MTESt:SEGMent:POINt:ADD
:MTESt:SEGMent:POINt:ADD
:MTESt:SEGMent:POINt:ADD
:MTESt:SEGMent:POINt:COUNt?     //Query the number of points
-->4
:MTESt:SEGMent:POINt:X -60NS    //Set the coordinates of point 1
:MTESt:SEGMent:POINt:Y 50 MV
:MTESt:SEGMent:POINt2:X -40NS   //Set the coordinates of point 2
:MTESt:SEGMent:POINt2:Y -50 MV
:MTESt:SEGMent:POINt3:X -20NS   //Set the coordinates of point 3
:MTESt:SEGMent:POINt4:Y 100 MV  //Set the coordinates of point 4, X = 0
```

## 18.5.3 Saving screenshots to file

The example saves three display images in png format to the files `Print.png`, `Print_001.png`, and `Print_002.png` on USB flash drive in the directory `/run/media/usb/<MyDriveName>`. For <MyDriveName>, use the name that was defined when formatting of the USB flash drive. To get a correct screenshot, turn on the display first.

Command description in: Section 18.13.8, "Screenshots", on page 1137.

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

► Connect a USB flash drive.

```
SYST:DISP:UPD ON
HCOP:DEST MMEM
```

```
HCOP:DEV:LANG PNG
*OPC?
MMEM:NAME '/run/media/usb/<MyDriveName>/Print.png'
HCOP:IMMediate; *OPC?
HCOP:IMM:NEXT; *OPC?
HCOP:IMM:NEXT; *OPC?
```

### 18.5.4 Data transfer in roll mode

The example shows the effects of `EXPort:WAVeform:SCOPe` and `ACQuire:ROLLmode:OSCapture` in roll mode.

Sample rate mode and record length mode are set to AUTO, and the record length is 10 M points.

```
ACQuire:SRATe:MODE AUTO
ACQuire:POINts:MODE AUTO
ACQuire:POINts?
--> 10000000
ACQuire:ROLLmode:OSCapture OFF
CHANnel1:DATA:HEADer?
--> -5,4.999999,10000000,1     //10 M points of data are transferred
ACQuire:ROLLmode:OSCapture ON
EXPort:WAVeform:SCOPe DISP     //Export of the displayed data
CHANnel1:DATA:HEADer?
--> -5,4.999999,10000000,1     //10 M points of data are transferred
EXPort:WAVeform:SCOPe ALL      //Export of all data
CHANnel1:DATA:HEADer?
--> -31.702346,4.99999999,36702346,1   //36.7 M points of data are transferred
```

## 18.6 Common commands

Common commands are described in the IEEE 488.2 (IEC 625-2) standard. These commands have the same effect and are employed in the same way on different devices. The headers of these commands consist of "*" followed by three letters. Many common commands are related to the Status Reporting System.

Available common commands:

*CAL?....947
*CLS....947
*IDN?....947
*OPC....947
*OPT?....948
*RCL....948
*RST....948
*SAV....948
*SRE....949
*STB?....949

*TRG ..... 949
*TST? ..... 949
*WAI ..... 950

***CAL?**

Starts a self-alignment of the instrument, and then queries a status response. Return values ≠ 0 indicate an error.

**Return values:**

<State> 0: no error
1: alignment failed
2: not aligned, e.g. init
3: device needs longer warmup time before selfalignment can start
4: input signal connected during selfalignment

**Usage:** Query only

***CLS**

Clear status

Sets the status byte (STB), the standard event register (ESR) and the `EVENt` part of the `QUEStionable` and the `OPERation` registers to zero. The command does not alter the mask and transition parts of the registers. It clears the output buffer.

**Usage:** Setting only

***IDN?**

Identification

Returns the instrument identification.

**Return values:**

<ID> "Rohde&Schwarz,<device type>,<part number>/<serial number>,<firmware version>"

**Example:** `Rohde&Schwarz,MXO3,1335.2050K04/300222,3.0.4.1`
Model assignment:

**Usage:** Query only

**Manual operation:** See "Instrument" on page 84

***OPC**

Operation complete

Sets bit 0 in the event status register when all preceding commands have been executed. This bit can be used to initiate a service request. The query writes a "1" into the output buffer when all preceding commands have been executed, which is useful for command synchronization.

---

***OPT?**

Option identification query

Queries the options included in the instrument. For a list of all available options and their description, refer to the specifications document.

**Return values:**

<Options> The query returns a list of options. The options are returned at fixed positions in a comma-separated string. A zero is returned for options that are not installed.

**Usage:** Query only

---

***RCL** <Number>

Recall

Loads the instrument settings from an intermediate memory identified by the specified number. The instrument settings can be stored to this memory using the command `*SAV` with the associated number.

The stored instrument settings do not include waveform generator settings.

It also activates the instrument settings which are stored in a file and loaded using `MMEMory:LOAD:STATe`.

---

***RST**

Reset

Sets the instrument to a default status. The default settings are indicated in the description of commands. Default settings are fixed, they are *not* changed by user-defined preset.

The command does not affect the waveform generator settings. To reset the generator, use `WGENerator<wg>:PRESet`.

**Usage:** Setting only

---

***SAV** <Number>

Save

Stores the current instrument settings under the specified number in an intermediate memory. The settings can be recalled using the command `*RCL` with the associated number.

Waveform generator settings are not included.

To transfer the stored instrument settings in a file, use the command `:MMEM:STOR:STAT`.

---

***SRE** <Contents>

Service request enable

Sets the service request enable register to the indicated value. This command determines under which conditions a service request is triggered.

**Parameters:**

<Contents> Contents of the service request enable register in decimal form. Bit 6 (MSS mask bit) is always 0.

Range: 0 to 255

---

***STB?**

Status byte query

Reads the contents of the status byte in decimal form.

**Usage:** Query only

---

***TRG**

Trigger

Triggers all actions waiting for a trigger event. In particular, *TRG generates a manual trigger signal. This common command complements the commands of the `TRIGger` subsystem.

`*TRG` corresponds to the `INITiate:IMMediate` command.

**Usage:** Event

---

***TST?**

Self-test query

Initiates self-tests of the instrument and returns an error code.

**Return values:**

<ErrorCode> **integer > 0 (in decimal format)**
An error occurred.

**0**
No errors occurred.

**Usage:** Query only

**Manual operation:** See "Run tests" on page 116

***WAI**

Wait to continue

Prevents servicing of the subsequent commands until all preceding commands have been executed and all signals have settled (see also command synchronization and `*OPC`).

**Usage:** Event

## 18.7 General remote settings

This section describes commands that affect many other remote commands in different applications of the MXO 3.

FORMat[:DATA] ..... 950
FORMat:BORDer ..... 951
FORMat:BPATtern ..... 952
SYSTem:DISPlay:UPDate ..... 952
SYSTem:DISPlay:MESSage:STATe ..... 952
SYSTem:DISPlay:MESSage[:TEXT] ..... 952

**FORMat[:DATA]** [<Format>],[<Length>]

Selects the data type that is used for transmission of data from analog channels, math and reference waveforms, and some measurement results from the instrument to the controlling computer.

For INT and REAL formats, use `FORMat:BORDer` to set the byte order.

If you need physical data (e.g. in Volt or Ampere) for further analysis, use the floating point formats (REAL) for export. No data conversion is needed.

**Parameters:**

<Format>,<Length> ASCii | REAL,32 | REAL,64 | INT,8 | INT,16 | INT,32

**ASCii**
Data values are returned in ASCII format as a list of comma-separated values in floating point format. The length can be omitted. It is 0, which means that the instrument selects the number of digits to be returned. The query returns both values (`ASC,0`).

**REAL,32**
Physical data in single precision, 32 bit floating point format.
The data is stored as binary data (Definite Length Block Data according to IEEE 488.2). Each waveform value is formatted in the 32-Bit IEEE 754 floating point format.
The schema of the result string is as follows:
`#41024<value1><value2>…<value n>`, with:
`#4` = number of digits (= 4 in the example) of the following number
`1024` = number of following data bytes (= 1024 in the example)
`<value>` = 4-byte floating point values
For large data (≥ 1 GB), the result string starts with "#(data length)". The number inside the parentheses indicates the real data length in bytes.

**REAL,64**
Double precision, 64 bit floating point format.

**INT,8 | INT,16 | INT,32**
Signed integer data with length 8 bit, 16 bit, or 32 bit.
The result string has the same schema as the REAL format.
For INT,16, you can set the byte order using the command.
For digital channel data, math and histogram data, INT formats are not available.

*RST: ASCii

**Example:**
```
FORMat:DATA REAL,32
FORMat:DATA?
REAL,32
```

**Usage:** SCPI confirmed
Asynchronous command

**Manual operation:** See "Transfer data format" on page 90

---

**FORMat:BORDer** <ByteOrder>

Sets the endianness.

The command is only relevant for data in integer and float format.

**Parameters:**
<ByteOrder> LSBFirst | MSBFirst

LSB first: little endian, least significant byte first
MSB first: big endian, most significant byte first

*RST: LSBFirst

**Example:** A number in INT32 format is 439041101, which is hex 1A2B3C4D.
With LSBFirst (default), 4D 3C 2B 1A is returned.
With MSBFirst, 1A 2B 3C 4D is returned.

**Usage:** Asynchronous command

**FORMat:BPATtern** <BtPattFmt>

Sets the number format for remote bit pattern queries on serial protocols.

**Parameters:**
<BtPattFmt> DEC | HEX | OCT | BIN | ASCII | ASCii | STRG
*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Bit pattern format" on page 90

**SYSTem:DISPlay:UPDate** <DisplayUpdate>

Defines whether the display is updated while the instrument is in the remote state. If the display is switched off, the normal GUI is replaced by a static image while the instrument is in the remote state. Switching off the display can speed up the measurement. OFF is the recommended state.

**Parameters:**
<DisplayUpdate> OFF | ON
`ON| 1`: The display is shown and updated during remote control.
`OFF| 0`: The display shows a static image during remote control.
*RST: OFF

**Example:** `SYSTem:DISPlay:UPDate 1`
Switch on the display update.

**Usage:** Asynchronous command

**SYSTem:DISPlay:MESSage:STATe** <DispMessSt>

Enables and disables the display of an additional text in remote control.

To define the text, use `SYSTem:DISPlay:MESSage[:TEXT]`.

**Parameters:**
<DispMessSt> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**SYSTem:DISPlay:MESSage[:TEXT]** <DisplayMessage>

Defines an additional text that is displayed during remote control operation.

To enable the text display, use `SYSTem:DISPlay:MESSage:STATe`.

**Parameters:**
<DisplayMessage>

**Usage:** Asynchronous command

## 18.8 Instrument setup


- System ..... 953
- Network drive mapping ..... 955
- SmartGrid ..... 957
- Toolbar ..... 963
- Annotations ..... 964
- Appearance ..... 976
- Display ..... 981
- Save/Recall ..... 984
- Preset ..... 986
- Maintenance ..... 987


### 18.8.1 System


SYSTem:COMMunicate:NET[:HOSTname] ..... 953
SYSTem:DATE ..... 953
SYSTem:EXIT ..... 954
SYSTem:FW:FILepath ..... 954
SYSTem:PRESet ..... 954
SYSTem:SHUTdown ..... 954
SYSTem:FW:STARt ..... 955
SYSTem:TIME ..... 955


---

**SYSTem:COMMunicate:NET[:HOSTname]** <Hostname>

Sets the host name of the instrument, which is required when configuring a network. After changing the host name, you have to reboot the instrument.

The query `SYSTem:COMMunicate:NET:HOSTname?` returns the currently defined host name.

**Parameters:**
<Hostname> String parameter

**Usage:** Asynchronous command

**Manual operation:** See "Host name" on page 85

---

**SYSTem:DATE** <Year>,<Month>,<Day>

Sets the date of the internal calendar.

**Parameters:**

<Year> Year, to be entered as a four-digit number (including the century and millennium information)
Range: 2012 to 2099
Increment: 1
*RST: 2012

<Month> Range: 1 (January) to 12 (December)
Increment: 1
*RST: 1

<Day> Range: 1 to maximum number of days in the specified month
Increment: 1
*RST: 1

**Example:** `SYSTem:DATE?`
Returned value: `2022,09,28`

**Usage:** Asynchronous command

---

**SYSTem:EXIT**

Starts the shutdown of the firmware.

**Usage:** Setting only
Asynchronous command

---

**SYSTem:FW:FILepath** <FilePath>

Sets the path and the filename of the firmware installation file.

**Parameters:**

<FilePath> String with path and filename

**Usage:** SCPI confirmed
Asynchronous command

---

**SYSTem:PRESet**

Resets the instrument to the default state, has the same effect as `*RST`.

**Usage:** Setting only
SCPI confirmed
Asynchronous command

**Manual operation:** See "Factory preset" on page 110

---

**SYSTem:SHUTdown**

Starts the shutdown of the instrument (firmware and operating system).

**Usage:** Setting only
Asynchronous command

---

**SYSTem:FW:STARt**

Starts the firmware update. Before starting, make sure that the correct path is set with `SYSTem:FW:FILepath`.

**Usage:** Event
SCPI confirmed
Asynchronous command

---

**SYSTem:TIME** <Hours>,<Minutes>,<Seconds>

Returns the current time of the clock.

**Parameters:**

<Hours>
Range: 0 to 24
Increment: 1
*RST: 1

<Minutes>
Range: 0 to 59
Increment: 1
*RST: 1

<Seconds>
Range: 0 to 59
Increment: 1
*RST: 1

**Usage:** Asynchronous command

## 18.8.2 Network drive mapping

MMEMory:NDRive<m>:CONNect ..... 955
MMEMory:NDRive<m>:DCONnect ..... 956
MMEMory:NDRive<m>:LPATh ..... 956
MMEMory:NDRive<m>:NPATh ..... 956
MMEMory:NDRive<m>:PWORd ..... 956
MMEMory:NDRive<m>:UNAMe ..... 957

---

**MMEMory:NDRive<m>:CONNect**

Connects the remote network folder to the local folder.

**Suffix:**

<m> 1 to 8, index of the network drive

**Usage:** Event
Asynchronous command

**Manual operation:** See "Connect, Disconnect, Connection state" on page 90

**MMEMory:NDRive<m>:DCONnect**

Disconnects the remote network folder from the local folder.

**Suffix:**
<m> 1 to 8, index of the network drive

**Usage:** Event
Asynchronous command

**Manual operation:** See "Connect, Disconnect, Connection state" on page 90

**MMEMory:NDRive<m>:LPATh** <LocalPath>

Sets the local folder on the MXO 5, to which the network folder is mapped. With this name, you can access the connected folder on the instrument.

**Suffix:**
<m> 1 to 8, index of the network drive

**Parameters:**
<LocalPath> String with the path of the local folder

**Usage:** Asynchronous command

**Manual operation:** See "Local folder" on page 89

**MMEMory:NDRive<m>:NPATh** <FilePath>

Sets the network folder that is mapped to the local folder.

**Suffix:**
<m> 1 to 8, index of the network drive

**Parameters:**
<FilePath> String with the full path in UNC syntax, for example, `\\net1\data\team1\osc`.

**Usage:** Asynchronous command

**Manual operation:** See "Network folder" on page 89

**MMEMory:NDRive<m>:PWORd** <Password>

Sets the password that is used to access the network folder.

Note that the SCPI protocol is NOT encrypted, so the password can be read on the network. Use the password at your own risk.

**Suffix:**
<m> 1 to 8, index of the network drive

**Setting parameters:**
<Password> String with the password

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "User name, Password" on page 89

---

**MMEMory:NDRive<m>:UNAMe** <UserName>

Sets the user name that is used to access the network folder.

**Suffix:**
<m> 1 to 8, index of the network drive

**Parameters:**
<UserName> String with the user name

**Usage:** Asynchronous command

**Manual operation:** See "User name, Password" on page 89

### 18.8.3 SmartGrid

The following `LAYout` commands configure the SmartGrid. In manual operation, you configure the SmartGrid by drag and drop.See Section 4.7, "Rohde & Schwarz SmartGrid", on page 69.

***Table 18-1: Terms and definitions of SmartGrid configuration***

| Term | Definition |
|---|---|
| Layout | A layout is a SmartGrid configuration. Several layouts can exist but only one layout is active.<br>By default, layouts are named "Tab <n>" on the display. |
| Children | A child is an area where data (data table, result table) or waveforms (diagram) are displayed. A node can also be a child. A child can be empty. |
| Node | A node consists of one or two children. A node is created by default with one child that has content. |
| Diagram | A diagram displays waveforms, the graphical visualization of data. |

LAYout<ly>:COUNt? 958
LAYout<ly>[:ENABle] 958
LAYout<ly>:ACTive 958
LAYout<ly>:SACTive 958
LAYout<ly>:LABel 959
LAYout<ly>:DIAGram<da>:COUNt? 959
LAYout<ly>:DIAGram<da>[:ENABle] 959
LAYout<ly>:DIAGram<da>:SOURce 960
LAYout<ly>:DIAGram<da>:LABel 960
LAYout<ly>:NODE<no>:COUNt? 960
LAYout<ly>:NODE<no>[:ENABle] 960
LAYout<ly>:NODE<no>:CHILdren<o>:CONTent<p>:ID 961
LAYout<ly>:NODE<no>:CHILdren<o>:CONTent<p>:TYPE 961
LAYout<ly>:NODE<no>:RATio 962

LAYout<ly>:NODE<no>:STYPe ..... 962
LAYout<ly>:RESult<n>:HORizontal:RATio ..... 962
LAYout<ly>:RESult<n>:VERTical:RATio ..... 963
LAYout<ly>:RPOSition ..... 963

**LAYout<ly>:COUNt?**

Returns the number of available layouts, i.e. SmartGrid configurations. By default, they named "Diagram Set" on the display.

You can query the maximum number of layouts with `LAYout:COUNt? MAX`.

**Suffix:**
<ly> Irrelevant, omit the suffix.

**Return values:**
<Count> Number of SmartGrid configurations

**Usage:** Query only
Asynchronous command

**LAYout<ly>[:ENABle]** <State>

Creates a new SmartGrid configuration and sets it active.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

**Parameters:**
<State> OFF | ON

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Asynchronous command

**LAYout<ly>:ACTive** <ActiveKey>

Sets the active SmartGrid configuration. The query returns the index of the active layout.

**Suffix:**
<ly> Irrelevant, omit the suffix.

**Parameters:**
<ActiveKey> Index of the active layout

**Usage:** Asynchronous command

**LAYout<ly>:SACTive**

Activates the specified SmartGrid configuration. The command has the same effect as `LAYout<ly>:ACTive` but it has no query, and the active layout is specified by the suffix.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Setting only
Asynchronous command

---

**LAYout<ly>:LABel** <Label>

Defines a name for the specified layout (SmartGrid configuration).

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

**Parameters:**
<Label> String with the layout name

**Usage:** Asynchronous command

---

**LAYout<ly>:DIAGram<da>:COUNt?**

Returns the number of diagrams in a specified layout.

You can query the maximum number of diagrams with `LAYout<ly>:DIAGram:COUNt? MAX`.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<da> Irrelevant, omit the suffix.

**Return values:**
<Count> Number of diagrams

**Usage:** Query only
Asynchronous command

---

**LAYout<ly>:DIAGram<da>[:ENABle]** <State>

Creates and displays a specified diagram in a specified layout. OFF deletes the diagram.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<da> 1 to 8, index of the diagram

**Parameters:**
<State> OFF | ON

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Asynchronous command

**LAYout<ly>:DIAGram<da>:SOURce** <SignalKeys>

Assigns the waveforms to a diagram.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<da> 1 to 8, index of the diagram

**Parameters:**
<SignalKeys> String with a comma-separated list of waveforms, e.g. `"C1, C2, M1"`

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Asynchronous command

**LAYout<ly>:DIAGram<da>:LABel** <Label>

Defines a name for the specified diagram in a specified layout.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<da> 1 to 8, index of the diagram

**Parameters:**
<Label> String with the diagram name

**Usage:** Asynchronous command

**LAYout<ly>:NODE<no>:COUNt?**

Returns the maximum number of nodes that can be defined. This number is the maximum value for the node suffix.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<no> Irrelevant, omit the suffix.

**Return values:**
<Count> Maximum value for the node suffix

**Usage:** Query only
Asynchronous command

**LAYout<ly>:NODE<no>[:ENABle]** <State>

Creates the specified node in the specified layout. OFF deletes the node and its children.

The query returns whether the specified node exists (1) or not (0).

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<no> Index of the node

**Parameters:**
<State> OFF | ON

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Asynchronous command

---

**LAYout<ly>:NODE<no>:CHILdren<o>:CONTent<p>:ID** <ID>

Sets the content ID, the number of the specified content type.

For example, the "Diagram5" has Type=DIAGRAM and ID=5.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<no> Index of the node

<o> 1 | 2, child index

<p> Irrelevant, omit the suffix.

**Parameters:**
<ID> Numeric value

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Asynchronous command

---

**LAYout<ly>:NODE<no>:CHILdren<o>:CONTent<p>:TYPE** <Type>

Sets the content type for a specified child in a specified node: diagram, result table, another node, or empty.

For example, the "Diagram5" has Type=DIAGRAM and ID=5.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<no> Index of the node

<o> 1 | 2, child index

<p> Irrelevant, omit the suffix.

**Parameters:**
<Type> NONE | DIAG | DIAGRAM | RES | RESULT | NODE
DIAG = DIAGRAM, RES = RESULT
*RST: NONE

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Asynchronous command

**LAYout<ly>:NODE<no>:RATio** <SplitRatio>

Sets the size ratio of the two children in the specified node.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<no> Index of the node

**Parameters:**
<SplitRatio> Size ratio of the children. 0.5 assigns 50% of the node size to each child. 0.3 assigns 30% to child 1% and 70% to child 2.

Range: 0 to 1
Increment: 0.0001
*RST: 0.5

**Usage:** Asynchronous command

**LAYout<ly>:NODE<no>:STYPe** <SplitType>

Creates a second child (e.g. diagram) in the node if only one child exists, and sets the splitting of the node. If two children exist, only the splitting is set.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<no> Index of the node

**Parameters:**
<SplitType> HOR | HORizontal | VERT | VERTical

HOR = HORizontal, VERT = VERTical

*RST: NONE

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Asynchronous command

**LAYout<ly>:RESult<n>:HORizontal:RATio** <SplitRatio>

Sets the horizontal ratio between the cursor result table and the measurement result table inside the result display container if the results are displayed at the bottom.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<n> Suffix is irrelevant, index of the result container

**Parameters:**
<SplitRatio> Range: 0 to 1
Increment: 0.0001
*RST: 0.8

**Usage:** Asynchronous command

**LAYout<ly>:RESult<n>:VERTical:RATio** <SplitRatio>

Sets the vertical ratio between the cursor result table and the other measurement results inside the result display container if the results are displayed on the right or left.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<n> Suffix is irrelevant, index of the result container

**Parameters:**
<SplitRatio> Range: 0 to 1
Increment: 0.0001
*RST: 0.8

**Usage:** Asynchronous command

**LAYout<ly>:RPOSition** <ResultPosition>

Defines the position of the result container inside the layout.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

**Parameters:**
<ResultPosition> BOTT | BOTTOM | LEFT | RIGH | RIGHT
*RST: NONE

**Usage:** Asynchronous command

### 18.8.4 Toolbar

DISPlay:TOOLbar:COUNt? ..... 963
DISPlay:TOOLbar:DESelect ..... 963
DISPlay:TOOLbar:RESTore ..... 964

**DISPlay:TOOLbar:COUNt?**

Returns the number of tools that are currently assigned to the toolbar.

**Return values:**
<ToolCount> Range: 0 to 100
Increment: 1
*RST: 8

**Usage:** Query only
Asynchronous command

**DISPlay:TOOLbar:DESelect**

Removes all tools from the toolbar.

**Usage:** Setting only
SCPI confirmed
Asynchronous command

---

**DISPlay:TOOLbar:RESTore**

Resets the toolbar to the factory configuration.

**Usage:** Setting only
SCPI confirmed
Asynchronous command

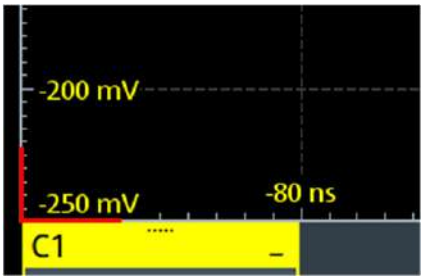
## 18.8.5 Annotations

The following `DISPlay:ANNotation` commands configure the annotations. With these commands you can select the type of annotation you want to add, define its value, position and color.

### Defining the position of the annotation

With some of the following commands, you can define the position of the annotation on the screen. When defining the position, consider the following:

- A position is defined with two elements. The first element refers to the X dimension and the second to the Y dimension.
- The minimum position (0,0) is the lower left corner of the SmartGrid.




- The maximum position (100,100) is the upper right corner of the SmartGrid.




- The position is defined in percent with a range from 0 to 100. The minimum increment is 0.01.
- An annotation element cannot be moved outside the SmartGrid, the position is limited depending on the size of the annotation element.

### Color catalog

When using annotations, you can select their color from a list of predefined colors. The following table gives an overview on the available colors and their command parameter name.

*Table 18-2: Color catalog for annotations*

| Color | Color command parameter | Description |
|---|---|---|
|  | WHITe | White |
|  | LGRay | Light gray |
|  | MGRay | Middle gray |
|  | GRAY | Gray |
|  | DGRay | Dark gray |
|  | RED | Red |
|  | DORange | Dark orange |
|  | ORANge | Orange |
|  | LORange | Light orange |
|  | YELLow | Yellow |
|  | DGReen | Dark green |
|  | GREen | Green |
|  | LGReen | Light green |
|  | LBLue | Light blue |
|  | BLUE | Blue |
|  | PINK | Pink |

| Color | Color command parameter | Description |
|---|---|---|
| | LPINk | Light pink |
| | TURQuoise | Turquoise |
| | LPURple | Light purple |
| | PURPle | Purple |

### 18.8.5.1 General settings

DISPlay:ANNotation:CLEar ..... 966
DISPlay:ANNotation:COLor<m>:CATalog? ..... 966
DISPlay:ANNotation:TRANsparency ..... 966

**DISPlay:ANNotation:CLEar**

Removes all existing annotations.

**Usage:** Setting only
Asynchronous command

**DISPlay:ANNotation:COLor<m>:CATalog?**

Returns the list of possible colors, see Color catalog for annotations.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Return values:**
<ColorCatalog> WHITe | LGRay | MGRay | GRAY | DGRay | RED | DORange | ORANge | LORange | YELLow | DAGReen | GREen | LIGReen | LBLue | BLUE | PINK | LPINk | TURQuoise | LPURple | PURPle

Returned as comma-separated values:
WHIT,LGR,MGR,GRAY,DGR,RED,DOR,ORAN,LOR,YELL,DAGR,GRE,LIGR,LBL,BLUE,PINK,LPIN,TURQ,LPUR,PURP

**Usage:** Query only
Asynchronous command

**DISPlay:ANNotation:TRANsparency** <Transparency>

Sets a transparency of all annotations. For high transparency values, you can see the waveform display in the background. For lower transparency values, readability of the annotation improves.

**Parameters:**

<Transparency> Range: 0 to 90
Increment: 1
*RST: 0

**Usage:** Asynchronous command

### 18.8.5.2 Rectangle

DISPlay:ANNotation:RECTangle<m>:CLEar ..... 967
DISPlay:ANNotation:RECTangle<m>:COLor ..... 967
DISPlay:ANNotation:RECTangle<m>:HEIGht ..... 967
DISPlay:ANNotation:RECTangle<m>:HORizontal:POSition ..... 968
DISPlay:ANNotation:RECTangle<m>:REMove ..... 968
DISPlay:ANNotation:RECTangle<m>:VERTical:POSition ..... 968
DISPlay:ANNotation:RECTangle<m>:WIDTh ..... 969
DISPlay:ANNotation:RECTangle<m>[:VALue] ..... 969

**DISPlay:ANNotation:RECTangle<m>:CLEar**

Deletes all rectangle annotations.

**Suffix:**

<m> Irrelevant, omit the suffix.

**Usage:** Setting only
Asynchronous command

**DISPlay:ANNotation:RECTangle<m>:COLor** <Type>

Sets the color of the indicated rectangle annotation.

**Suffix:**

<m> 1 to <number of rectangle annotations>
Index of the rectangle annotation

**Parameters:**

<Type> WHITe | LGRay | MGRay | GRAY | DGRay | RED | DORange | ORANge | LORange | YELLow | DAGReen | GREen | LIGReen | LBLue | BLUE | PINK | LPINk | TURQuoise | LPURple | PURPle

See Color catalog for annotations.

**Usage:** Asynchronous command

**DISPlay:ANNotation:RECTangle<m>:HEIGht** <Height>

Sets the height (vertical size) of the rectangle annotation.

**Suffix:**

<m> 1 to <number of rectangle annotations>
Index of the rectangle annotation

**Parameters:**

<Height> In % of the screen.

Range: 0 to 100
Increment: 1
0 results in an invisible rectangle, 100 takes the complete height of the screen.

**Usage:** Asynchronous command

---

**DISPlay:ANNotation:RECTangle<m>:HORizontal:POSition** <HorizontalPos>

Sets the horizontal position of the left edge of the rectangle annotation. See also Defining the position of the annotation.

**Suffix:**

<m> 1 to <number of rectangle annotations>
Index of the rectangle annotation

**Parameters:**

<HorizontalPos> In % of the screen

Range: 0 to 99

**Usage:** Asynchronous command

---

**DISPlay:ANNotation:RECTangle<m>:REMove**

Removes the specified rectangle annotation from the screen.

**Suffix:**

<m> 1 to <number of rectangle annotations>
Index of the rectangle annotation

**Usage:** Setting only
Asynchronous command

---

**DISPlay:ANNotation:RECTangle<m>:VERTical:POSition** <VerticalPos>

Sets the vertical position of the lower edge of the rectangle annotation. See also Defining the position of the annotation.

**Suffix:**

<m> 1 to <number of rectangle annotations>
Index of the rectangle annotation

**Parameters:**

<VerticalPos> In % of the screen

Range: 0 to 99

**Usage:** Asynchronous command

**DISPlay:ANNotation:RECTangle<m>:WIDTh** <Width>

Sets the width (horizontal size) of the rectangle annotation.

**Suffix:**
<m> 1 to <number of rectangle annotations>
Index of the rectangle annotation

**Parameters:**
<Width> In % of the screen
Range: 0 to 100
Increment: 1
0 results in an invisible rectangle, 100 takes the complete width of the screen.

**Usage:** Asynchronous command

**DISPlay:ANNotation:RECTangle<m>[:VALue]** [<HorizontalPos>],[<VerticalPos>], [<Width>],[<Height>]

Adds a new rectangle annotation, or replaces the annotation if it already exists. If no parameters are defined, the default values are used.

All parameters are given in % of the screen.

**Suffix:**
<m> *
Index of the created rectangle annotation

**Parameters:**
<HorizontalPos> Range: 0 to 99

<VerticalPos> Range: 0 to 99

<Width> Range: 0 to 100

<Height> Range: 0 to 100

**Usage:** Asynchronous command

### 18.8.5.3 Arrow

DISPlay:ANNotation:ARRow<m>:CLEar....970
DISPlay:ANNotation:ARRow<m>:COLor....970
DISPlay:ANNotation:ARRow<m>:DIRection....970
DISPlay:ANNotation:ARRow<m>:HEIGht....970
DISPlay:ANNotation:ARRow<m>:HORizontal:POSition....971
DISPlay:ANNotation:ARRow<m>:REMove....971
DISPlay:ANNotation:ARRow<m>:VERTical:POSition....971
DISPlay:ANNotation:ARRow<m>:WIDTh....972
DISPlay:ANNotation:ARRow<m>[:VALue]....972

**DISPlay:ANNotation:ARRow<m>:CLEar**

Deletes all arrow annotations.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Usage:** Setting only
Asynchronous command

**DISPlay:ANNotation:ARRow<m>:COLor** <Type>

Sets the color of the indicated arrow annotation.

**Suffix:**
<m> 1 to <number of arrow annotations>
Index of the arrow annotation

**Parameters:**
<Type> WHITe | LGRay | MGRay | GRAY | DGRay | RED | DORange | ORANge | LORange | YELLow | DAGReen | GREen | LIGReen | LBLue | BLUE | PINK | LPINk | TURQuoise | LPURple | PURPle

See Color catalog for annotations.

**Usage:** Asynchronous command

**DISPlay:ANNotation:ARRow<m>:DIRection** <Type>

Sets the direction of the indicated arrow annotation from starting point to arrow tip.

**Suffix:**
<m> 1 to <number of arrow annotations>
Index of the arrow annotation

**Parameters:**
<Type> TLEFt | TRIGht | BLEFt | BRIGht

`TLEFt`: to top left
`TRIGht`: to top right
`BLEFt`: to bottom left
`BRIGht`: to bottom right

**Usage:** Asynchronous command

**DISPlay:ANNotation:ARRow<m>:HEIGht** <Height>

Sets the height (vertical size) of the arrow annotation.

**Suffix:**
<m> 1 to <number of arrow annotations>
Index of the arrow annotation

**Parameters:**

<Height> In % of the screen.

Range: 0 to 100
Increment: 1
0 results in an invisible rectangle, 100 takes the complete height of the screen.

**Usage:** Asynchronous command

---

**DISPlay:ANNotation:ARRow<m>:HORizontal:POSition** <HorizontalPos>

Sets the horizontal position of the left point of the arrow annotation. See also Defining the position of the annotation.

**Suffix:**

<m> 1 to <number of arrow annotations>
Index of the arrow annotation

**Parameters:**

<HorizontalPos> In % of the screen

Range: 0 to 99

**Usage:** Asynchronous command

---

**DISPlay:ANNotation:ARRow<m>:REMove**

Removes the specified arrow annotation from the screen.

**Suffix:**

<m> 1 to <number of arrow annotations>
Index of the arrow annotation

**Usage:** Setting only
Asynchronous command

---

**DISPlay:ANNotation:ARRow<m>:VERTical:POSition** <VerticalPos>

Sets the vertical position of the lower point of the arrow annotation. See also Defining the position of the annotation.

**Suffix:**

<m> 1 to <number of arrow annotations>
Index of the arrow annotation

**Parameters:**

<VerticalPos> In % of the screen

Range: 0 to 99

**Usage:** Asynchronous command

**DISPlay:ANNotation:ARRow<m>:WIDTh** <Width>

Sets the width (horizontal size) of the arrow annotation.

**Suffix:**
<m> 1 to <number of arrow annotations>
Index of the arrow annotation

**Parameters:**
<Width> In % of the screen
Range: 0 to 100
Increment: 1
0 results in an invisible rectangle, 100 takes the complete width of the screen.

**Usage:** Asynchronous command

**DISPlay:ANNotation:ARRow<m>[:VALue]** [<HorizontalPos>],[<VerticalPos>], [<Width>],[<Height>]

Adds a new arrow annotation, or replaces the annotation if it already exists. If no parameters are defined, the default values are used.

All parameters are given in % of the screen.

**Suffix:**
<m> *
Index of the created arrow annotation

**Parameters:**
<HorizontalPos> Range: 0 to 99
<VerticalPos> Range: 0 to 99
<Width> Range: 0 to 100
<Height> Range: 0 to 100

**Usage:** Asynchronous command

### 18.8.5.4 Text

DISPlay:ANNotation:TEXT<m>:CLEar ... 972
DISPlay:ANNotation:TEXT<m>:COLor ... 973
DISPlay:ANNotation:TEXT<m>:FONTsize ... 973
DISPlay:ANNotation:TEXT<m>:HORizontal:POSition ... 973
DISPlay:ANNotation:TEXT<m>:REMove ... 974
DISPlay:ANNotation:TEXT<m>:VERTical:POSition ... 974
DISPlay:ANNotation:TEXT<m>[:VALue] ... 974

**DISPlay:ANNotation:TEXT<m>:CLEar**

Deletes all text annotations.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Usage:** Setting only
Asynchronous command

---

**DISPlay:ANNotation:TEXT<m>:COLor** <Type>

Sets the color of the indicated text annotation.

**Suffix:**
<m> 1 to <number of text annotations>
Index of the text annotation

**Parameters:**
<Type> WHITe | LGRay | MGRay | GRAY | DGRay | RED | DORange | ORANge | LORange | YELLow | DAGReen | GREen | LIGReen | LBLue | BLUE | PINK | LPINk | TURQuoise | LPURple | PURPle

See Color catalog for annotations.

**Usage:** Asynchronous command

---

**DISPlay:ANNotation:TEXT<m>:FONTsize** <Fontsize>

Sets the font size of the text.

**Suffix:**
<m> 1 to <number of text annotations>
Index of the text annotation

**Parameters:**
<Fontsize> Range: 12 to 30
Default unit: px

**Usage:** Asynchronous command

---

**DISPlay:ANNotation:TEXT<m>:HORizontal:POSition** <HorizontalPos>

Sets the horizontal position of the left edge of the text annotation. See also Defining the position of the annotation.

**Suffix:**
<m> 1 to <number of text annotations>
Index of the text annotation

**Parameters:**
<HorizontalPos> In % of the screen
Range: 0 to 99

**Usage:** Asynchronous command

**DISPlay:ANNotation:TEXT<m>:REMove**

Removes the specified text annotation from the screen.

**Suffix:**
<m> 1 to <number of text annotations>
Index of the text annotation

**Usage:** Setting only
Asynchronous command

**DISPlay:ANNotation:TEXT<m>:VERTical:POSition** <VerticalPos>

Sets the vertical position of the lower edge of the text annotation. See also Defining the position of the annotation.

**Suffix:**
<m> 1 to <number of text annotations>
Index of the text annotation

**Parameters:**
<VerticalPos> In % of the screen
Range: 0 to 99

**Usage:** Asynchronous command

**DISPlay:ANNotation:TEXT<m>[:VALue]** [<Text>]

Adds a new text annotation or replaces the text value of an already existing text annotation.

**Suffix:**
<m> *
Index of the created text annotation

**Parameters:**
<Text> String with the annotation text

**Usage:** Asynchronous command

### 18.8.5.5 Draw

DISPlay:ANNotation:PLINe<m>:CLEar....974
DISPlay:ANNotation:PLINe<m>:COLor....975
DISPlay:ANNotation:PLINe<m>:EXTend....975
DISPlay:ANNotation:PLINe<m>:REMove....975
DISPlay:ANNotation:PLINe<m>[:VALue]....976

**DISPlay:ANNotation:PLINe<m>:CLEar**

Deletes all draw annotations.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Usage:** Setting only
Asynchronous command

---

**DISPlay:ANNotation:PLINe<m>:COLor** <Type>

Sets the color of the indicated draw annotation.

**Suffix:**
<m> 1 to <number of draw annotations>
Index of the drawn annotation.

**Parameters:**
<Type> WHITe | LGRay | MGRay | GRAY | DGRay | RED | DORange | ORANge | LORange | YELLow | DAGReen | GREen | LIGReen | LBLue | BLUE | PINK | LPINk | TURQuoise | LPURple | PURPle

See Color catalog for annotations.

**Usage:** Asynchronous command

---

**DISPlay:ANNotation:PLINe<m>:EXTend** [<x>],[<y>]

Expands the drawn item with a segment. The x and y position of the existing endpoint is the start point of the new segment. The segments are always straight lines.

**Suffix:**
<m> 1 to <number of draw annotations>
Index of the draw annotation

**Setting parameters:**
<x> Horizontal position of the endpoint of the new segment

<y> Vertical position of the endpoint of the new segment

**Usage:** Setting only
Asynchronous command

---

**DISPlay:ANNotation:PLINe<m>:REMove**

Removes the specified draw annotation from the screen.

**Suffix:**
<m> 1 to <number of draw annotations>
Index of the draw annotation

**Usage:** Setting only
Asynchronous command

**DISPlay:ANNotation:PLINe<m>[:VALue]** <x1>,<y1>,<x2>,<y2>,[<xn>]

Adds a new draw annotation, or replaces the annotation if it already exists. If no parameters are defined, the default values are used.

All parameters are given in % of the screen.

**Suffix:**

<m> *
Index of the created draw annotation

**Parameters:**

<x1> Horizontal position of the start point of the new line

<y1> Vertical position of the start point of the new line

<x2> Horizontal position of the endpoint of the new line

<y2> Vertical position of the endpoint of the new line

<xn> Vertical or horizontal position of extension points. For each point, you need two values, first the horizontal and second the vertical value. All values are separated by commas.

**Usage:** Asynchronous command

## 18.8.6 Appearance


- Waveform colors ..... 976
- Grid appearance ..... 978
- Dialog appearance ..... 980
- Measurement appearance ..... 981
- Peak list appearance ..... 981


### 18.8.6.1 Waveform colors


DISPlay:COLor:SIGNal:CATalog? ..... 976
DISPlay:COLor:SIGNal:COLor ..... 977
DISPlay:COLor:SIGNal:ASSign ..... 977
DISPlay:COLor:SIGNal:USE ..... 977


**DISPlay:COLor:SIGNal:CATalog?**

Returns a list of valid signal names. The signal names are needed in other DISPlay:COLor commands to set the <Signal> parameter.

**Return values:**

<Signals> Comma-separated list of signal names, see Section 18.4.1, "Waveform parameter", on page 939

**Usage:** Query only
Asynchronous command

**DISPlay:COLor:SIGNal:COLor** <Signal>,<Value>
**DISPlay:COLor:SIGNal:COLor?** <Signal>

Sets the color of the selected waveform.

**Parameters:**

<Value> Decimal value of the ARGB color. Use the color dialog box on the instrument to get the hex value of the color, and convert the hex value to a decimal value.
0 is fully transparent black.
4278190080 (dec) = FF000000 (hex) is opaque black.
4294967295 (dec) = FFFFFFFF (hex) is opaque white.
To reset the color to its default, use `DISPlay:COLor:SIGNal:COLor <Signal>,DEF`.

**Parameters for setting and query:**

<Signal> Signal name as returned by `DISPlay:COLor:SIGNal:CATalog?`.

**Usage:** Asynchronous command

**Manual operation:** See "Color" on page 98

---

**DISPlay:COLor:SIGNal:ASSign** <Signal>,<ColorTable>
**DISPlay:COLor:SIGNal:ASSign?** <Signal>

Assigns a color table to the source waveform instead of a dedicated color.

**Parameters:**

<ColorTable> String with the name of the color table.
Valid values are: `"FalseColors"`, `"Spectrum"`, `"SingleEvent"` and `"Temperature"`.

**Parameters for setting and query:**

<Signal> Signal name as returned by `DISPlay:COLor:SIGNal:CATalog?`.

**Usage:** Asynchronous command

**Manual operation:** See "Assigned color table" on page 97

---

**DISPlay:COLor:SIGNal:USE** <Signal>,<State>
**DISPlay:COLor:SIGNal:USE?** <Signal>

If enabled, the selected waveform is displayed according to its assigned color table.

If disabled, the selected color is displayed, and the intensity of the signal color varies according to the cumulative occurrence of the values.

The setting is not available for digital channels and parallel buses.

**Parameters:**

<State> OFF | ON

**Parameters for setting and query:**

<Signal> Signal name as returned by `DISPlay:COLor:SIGNal:CATalog?`.

**Usage:** Asynchronous command

**Manual operation:** See "Use color table" on page 97

### 18.8.6.2 Grid appearance

DISPlay:DIAGram:CROSshair....978
DISPlay:DIAGram:FINegrid....978
DISPlay:DIAGram:GRID....978
DISPlay:DIAGram:LABels....979
DISPlay:DIAGram:XPDivlabel....979
DISPlay:DIAGram:XFIXed....979
DISPlay:DIAGram:YFIXed....979

---

**DISPlay:DIAGram:CROSshair** <Crosshair>

If selected, a crosshair is displayed in the diagram area. A crosshair allows you to select a specific data point by its coordinates.

**Parameters:**

<Crosshair> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Show crosshair" on page 100

---

**DISPlay:DIAGram:FINegrid** <ShowFineScale>

If selected, the crosshair is displayed as a ruler with scale markers. If disabled, the crosshair is shown as dashed lines.

**Parameters:**

<ShowFineScale> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Show fine grid scale" on page 100

---

**DISPlay:DIAGram:GRID** <Show>

If selected, a grid is displayed in the diagram area. A grid helps you associate a specific data point to its exact value on the x- or y-axis.

**Parameters:**

<Show> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Show grid" on page 100

---

**DISPlay:DIAGram:LABels** <ShowLabels>

If selected, labels mark values on the x- and y-axes in specified intervals in the diagram.

**Parameters:**
<ShowLabels> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Show labels" on page 100

---

**DISPlay:DIAGram:XPDivlabel** <ShowDivLabelX>

If selected, the time scale value is shown at the diagram bottom instead of the horizontal grid labels. For example, 10 ns/div is shown instead of the values 0, 10, 20, 30... ns.

**Parameters:**
<ShowDivLabelX> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show div label (x)" on page 100

---

**DISPlay:DIAGram:XFIXed** <XGridFixed>

If enabled, the vertical grid lines remain in their position when the horizontal position is changed. Only the values at the grid lines are adapted.

**Parameters:**
<XGridFixed> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Keep X-grid fixed" on page 100

---

**DISPlay:DIAGram:YFIXed** <YGridFixed>

If enabled, the horizontal grid lines remain in their position when the position of the curve is changed. Only the values at the grid lines are adapted.

Fixed horizontal grid lines correspond to the behavior of traditional oscilloscopes.

**Parameters:**
<YGridFixed> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Keep Y-grid fixed" on page 100

### 18.8.6.3 Dialog appearance

DISPlay:DIALog:FONTsize....980
DISPlay:DIALog:TRANsparency....980
DISPlay:RESult:FONTsize....980

---

**DISPlay:DIALog:FONTsize** <DialogFontSize>

Sets the font size of the text in dialog boxes.

**Parameters:**

<DialogFontSize>
Range: 16 to 25
Increment: 1
*RST: 21

**Usage:** Asynchronous command

**Manual operation:** See "Font size (Dialog)" on page 102

---

**DISPlay:DIALog:TRANsparency** <DialogTransp>

Sets the transparency of the dialog box background. For high transparency values, you can see the waveform display in the background, and possibly check the effect of the changed setting. For lower transparency values, readability in the dialog box improves.

**Parameters:**

<DialogTransp>
Range: 0 to 70
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Transparency (Dialog)" on page 102

---

**DISPlay:RESult:FONTsize** <ResultFontSize>

Sets the font size of the text in result tables.

**Parameters:**

<ResultFontSize>
Range: 16 to 25
Increment: 1
*RST: 21

**Usage:** Asynchronous command

**Manual operation:** See "Font size (Result dialog)" on page 102

#### 18.8.6.4 Measurement appearance

**DISPlay:MEASurements:LEVels** <RefLevelValues>

If selected, the values of the reference levels are displayed.

**Parameters:**
<RefLevelValues> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show reference level values" on page 103

#### 18.8.6.5 Peak list appearance

CALCulate:SPECtrum<sp>:PLISt:LABel:BORDer....981

**CALCulate:SPECtrum<sp>:PLISt:LABel:BORDer** <LabelBorder>

Defines the layout of the labels, full border or none.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<LabelBorder> NOBorder | FULL
`FULL`: Full border
*RST: FULL

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 103

### 18.8.7 Display


- Persistence....981
- Signal....983
- Backlight....983
- Clear results....984


#### 18.8.7.1 Persistence

DISPlay:PERSistence:INFinite....982
DISPlay:PERSistence:RESet....982
DISPlay:PERSistence:TIME....982
DISPlay:PERSistence[:STATe]....982

**DISPlay:PERSistence:INFinite** <State>

If infinite persistence is enabled, each new waveform point remains on the screen until this option is disabled. Use infinite persistence to display rare events in the signal.

**Parameters:**

<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Infinite persistence" on page 105

**DISPlay:PERSistence:RESet**

Resets the display, removing all persistent waveform points.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Reset" on page 105

**DISPlay:PERSistence:TIME** <Time>

Sets a time factor that controls how long the waveforms points fade away from the display. Thus, the MXO 3 emulates the persistence of analog phosphor screens.

**Parameters:**

<Time> Range: 0.05 to 50
Increment: 0.05
*RST: 0.05
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Time" on page 105

**DISPlay:PERSistence[:STATe]** <State>

If enabled, each new data point in the diagram area remains on the screen for the duration defined using `DISPlay:PERSistence:TIME`, or as long as `DISPlay:PERSistence:INFinite` is enabled.

If disabled, the signal value is only displayed as long as it actually occurs.

**Parameters:**

<State> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 105

### 18.8.7.2 Signal

DISPlay:INTensity ..... 983
DISPlay:DIAGram:STYLe ..... 983

**DISPlay:INTensity** <Intensity>

The intensity determines the strength of the waveform line in the diagram. Enter a percentage between 0 (not visible) and 100% (strong). The default value is 50%.

**Parameters:**

<Intensity>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Intensity" on page 106

**DISPlay:DIAGram:STYLe** <Style>

Selects the style in which the waveform is displayed.

**Parameters:**

<Style>
VECTors | DOTS

**VECTors**
The individual data points are connected by a line.

**DOTS**
Only the individual data points are displayed.

*RST: VECTors

**Usage:** Asynchronous command

**Manual operation:** See "Waveform style" on page 106

### 18.8.7.3 Backlight

DISPlay:BACKlight[:BRIGtness] ..... 983
DISPlay:BACKlight:DIMMing ..... 984

**DISPlay:BACKlight[:BRIGtness]** <LCDIntensity>

Sets the background luminosity of the touchscreen.

**Parameters:**

<LCDIntensity>
Range: 5 to 100
Increment: 1
*RST: 90
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Brightness" on page 107

---

**DISPlay:BACKlight:DIMMing** <InactiveTime>

Selects a time, after which the monitor brightness is reduced, if the instrument was inactive. Remote control of the instrument is also considered as an activity.

**Parameters:**

<InactiveTime> OFF | T1Minute | T2Minutes | T3Minutes | T5Minutes | T10Minutes | T15Minutes | T20Minutes | T25Minutes | T30Minutes | T45Minutes | T1Hour | T2Hours | T3Hours | T4Hours

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Dimm on user inactivity" on page 107

#### 18.8.7.4 Clear results

---

**DISPlay:CLR**

Deletes all measurement results including all waveforms and statistics.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Clear" on page 65

### 18.8.8 Save/Recall

#### 18.8.8.1 Autonaming

MMEMory:AUTonaming:PREFix ..... 985
MMEMory:AUTonaming:TIME ..... 985
MMEMory:AUTonaming:INDex ..... 985
MMEMory:AUSave:ENABle ..... 985
MMEMory:AUSave:INTerval ..... 985
MMEMory:AUTonaming:USERtext ..... 985
MMEMory:AUTonaming:DEFaultpath ..... 986
MMEMory:AUTonaming:RESall ..... 986
MMEMory:AUTonaming:RESPath ..... 986
MMEMory:AUTonaming:TEXT ..... 986

**MMEMory:AUTonaming:PREFix** <MainNmeStemSt>
**MMEMory:AUTonaming:TIME** <DateTime>
**MMEMory:AUTonaming:INDex** <NameIndex>

Includes or excludes the prefix/ date/time /index in the filename pattern for automatic filename generation. This name is used as the default filename.

The prefix indicates the type of data that is saved, for example, RefCurve, Settings.

**Parameters:**
<NameIndex> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Index" on page 118

**MMEMory:AUSave:ENABle** <EnableAutosave>

Enables the automatic saving of the waveform. You can set the autosave interval with `MMEMory:AUSave:INTerval`.

**Parameters:**
<EnableAutosave> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable autosave" on page 118

**MMEMory:AUSave:INTerval** <AutosaveIntvl>

Defines the time interval for the automatic saving of the waveform, if `MMEMory:AUSave:ENABle` is set to `ON`.

**Parameters:**
<AutosaveIntvl> Range: 1 to 360000
Increment: 1
*RST: 300
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Enable autosave" on page 118

**MMEMory:AUTonaming:USERtext** <State>

If enabled, inserts the specified user text after the prefix.

You can define the text with `MMEMory:AUTonaming:TEXT`.

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "User text" on page 117

---

**MMEMory:AUTonaming:DEFaultpath** <Path>

Sets the path where data and settings files are stored. On the instrument, all user data is written to `/home/storage/userData`. You can create subfolders in this folder.

**Parameters:**
<Path> String parameter

**Usage:** Asynchronous command

**Manual operation:** See "Default path for all file operations" on page 118

---

**MMEMory:AUTonaming:RESall**

Resets all autonaming settings to the default value, including the path.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Reset path" on page 118

---

**MMEMory:AUTonaming:RESPath**

Resets the path for file operations to the factory default path.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Reset path" on page 118

---

**MMEMory:AUTonaming:TEXT** <NameString>

Defines a text that can be included in the autonaming pattern.

**Parameters:**
<NameString> String parameter

**Usage:** Asynchronous command

**Manual operation:** See "User text" on page 117

## 18.8.9 Preset

USERdefined:PRESet:NAME ..... 987
USERdefined:PRESet:OPEN ..... 987
USERdefined:PRESet:SAVE ..... 987
USERdefined:PRESet[:ENABle] ..... 987

**USERdefined:PRESet:NAME** <Name>

Sets the path, the filename and the file format of the preset file.

**Parameters:**

<Name> String with path and filename with extension `.set`.

**Example:** `USERdefined:PRESet:NAME '/home/storage/userData/settings/Preset_SPI.set'`

**Manual operation:** See "Preset file selection" on page 110

**USERdefined:PRESet:OPEN**

Opens and loads the preset file that is defined with `USERdefined:PRESet:NAME`.

**Usage:** Event

**Manual operation:** See "Preset file selection" on page 110

**USERdefined:PRESet:SAVE**

Saves the current settings as a preset file. You define the storage location and filename with `USERdefined:PRESet:NAME`.

**Usage:** Event

**Manual operation:** See "Save, Open" on page 110

**USERdefined:PRESet[:ENABle]** <Name>

If enabled, the settings from the selected saveset are restored when the [Preset] key is pressed.

If disabled, [Preset] sets the instrument to the factory defaults.

The saveset to be used as preset file is defined with `USERdefined:PRESet:NAME`.

**Parameters:**

<Name> OFF | ON

*RST: OFF

**Manual operation:** See "Enable user-defined preset" on page 109

### 18.8.10 Maintenance

Use also:

- `*CAL?` on page 947
- `*TST?` on page 949

CALibration[:ALL] ..... 988
CALibration:DATE? ..... 988
CALibration:TIME? ..... 988
CALibration:RESult? ..... 988
SERVice:REPort ..... 989
SYSTem:APUP ..... 989

### CALibration[:ALL]

`Calibration:ALL` starts the self-alignment process without returning status information. To get the status, use the commands of the operation status register.

`Calibration:ALL?` starts the self-alignment process and returns information on the state of the self-alignment. (Same as `*CAL?`).

The process can take several minutes. Consider your timeout settings.

**Return values:**
<Result> Numeric status indicator. Return values ≠ 0 indicate an error.

**Usage:** Asynchronous command

### CALibration:DATE?

Returns the date of the last self-alignment.

**Return values:**
<Date>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Date, Time, Overall alignment state" on page 113

### CALibration:TIME?

Returns the time of the last self-alignment.

**Return values:**
<Time>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Date, Time, Overall alignment state" on page 113

### CALibration:RESult?

Returns the result of the last self-alignment and the current alignment status. In remote mode, `*CAL?` provides more detailed information.

**Return values:**
<ResultState> PASSed | FAILed | NOALigndata
*RST: FAILed

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Date, Time, Overall alignment state" on page 113

---

**SERVice:REPort**

Creates a service report.

The service report is a ZIP file with a complete bug report, all relevant setup information, reporting and log files, alignment files, and the instrument configuration.

If a USB flash drive is connected, the report is saved on the USB flash drive. Otherwise, the report is saved in the user data folder `/home/storage/userData`.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Create report" on page 116

---

**SYSTem:APUP** <AutoPowerUp>

If enabled, the instrument powers up automatically when it is connected to the mains voltage.

**Parameters:**
<AutoPowerUp> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Auto power up" on page 115

## 18.9 Acquisition and waveform setup

### 18.9.1 Starting and stopping acquisition

Use the following commands to start and stop acquisitions.

RUNCont ..... 989
RUN ..... 989
RUNSingle ..... 990
SINGle ..... 990
STOP ..... 990

---

**RUNCont**
**RUN**

Starts the continuous acquisition.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "[Run / Stop]" on page 46

---

**RUNSingle**
**SINGle**

Starts a defined number of acquisition cycles. The number of cycles is set with `ACQuire:COUNt`.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "[Single]" on page 46

---

**STOP**

Stops the running acquisition.

**Usage:** Event
Asynchronous command

**Manual operation:** See "[Run / Stop]" on page 46

## 18.9.2 Horizontal setup

AUToscale....990
TIMebase:SCALe....990
TIMebase:RANGe....991
TIMebase:DIVisions?....991
TIMebase:HORizontal:POSition....991
TIMebase:REFerence....992

---

**AUToscale**

Performs an autoset process: analyzes the enabled channel signals, and obtains appropriate horizontal, vertical, and trigger settings to display stable waveforms.

Rohde & Schwarz does not recommend using the autoset in remote control. To adjust the oscilloscope remotely, especially for automated testing applications, use the remote commands that adjust the horizontal, vertical and trigger settings.

**Usage:** Event
Asynchronous command

---

**TIMebase:SCALe** <TimebaseScale>

Sets the horizontal scale, the time per division, for all waveforms in the time domain, for example, channel and math waveforms.

**Parameters:**

<TimebaseScale> Range: 200E-12 to 10E+3
Increment: 1E-12
*RST: 20E-9
Default unit: s/div

**Usage:** Asynchronous command

**Manual operation:** See "[Scale]" on page 47

---

**TIMebase:RANGe** <TimebaseRange>

Sets the time of one acquisition, which is the time across the 10 divisions of the diagram: *Acquisition time = Time scale * 10 divisions*.

**Parameters:**

<TimebaseRange> Range: 2E-9 to 100E+3
Increment: 1E-12
*RST: 200E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Timebase range" on page 121

---

**TIMebase:DIVisions?**

Returns the number of horizontal divisions on the screen. The number cannot be changed.

**Return values:**

<HorizDivCnt> *RST: 10

**Usage:** Query only
Asynchronous command

---

**TIMebase:HORizontal:POSition** <Position>

Defines the time distance between the reference point and the trigger point, which is the zero point of the diagram. The horizontal position is also known as trigger offset.

**Parameters:**

<Position> Range: -159.99E-3 to 1E+26
Increment: 1E-12
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "[Position]" on page 47

**TIMebase:REFerence** <RescaleCtrPos>

Sets the position of the reference point in % of the screen. It defines which part of the waveform is shown.

**Parameters:**

<RescaleCtrPos>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Reference point" on page 122

### 18.9.3 Roll mode

TIMebase:ROLL:ENABle ..... 992
TIMebase:ROLL:MTIMe ..... 992
ACQuire:ROLLmode:OSCapture ..... 993
ACQuire:ROLLmode:POINts? ..... 993
TIMebase:ROLL:STATe? ..... 993

**TIMebase:ROLL:ENABle** <Mode>

Selects, if the roll mode is started automatically by the instrument or if it is turned off.

**Parameters:**

<Mode>
AUTO | OFF
*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 123

**TIMebase:ROLL:MTIMe** <MinAcquTime>

Sets the minimum acquisition time for the automatic start of the roll mode.

**Parameters:**

<MinAcquTime>
Range: 0.5 to 100000
Increment: 1
*RST: 2
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Start roll time" on page 123

**ACQuire:ROLLmode:OSCapture** <OffScreenCapture>

If enabled, the analyzable waveform in roll mode is extended. You can run the roll mode, stop the acquisition after some time, and analyze the data that is on the display and in the unvisible area on the left.

**Parameters:**

<OffScreenCapture> OFF | ON
*RST: OFF

**Example:** See Section 18.5.4, "Data transfer in roll mode", on page 946.

**Usage:** Asynchronous command

**Manual operation:** See "Off screen capture" on page 124

**ACQuire:ROLLmode:POINts?**

Returns the record length of the roll mode. In roll mode, the complete record is always captured, independently of the displayed waveform.

To process and analyze the complete roll waveform, enable `ACQuire:ROLLmode:OSCapture`.

**Return values:**

<RecordLength> Range: 0 to 18446744073709551615
Increment: 1
*RST: 0
Default unit: pts

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Off screen capture" on page 124

**TIMebase:ROLL:STATe?**

Returns the status of the roll mode.

**Return values:**

<State> OFF | ON
*RST: OFF

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Mode" on page 123

## 18.9.4 Acquisition setup

This section describes all remote commands for acquisition.

ACQuire:AVAilable?....994
ACQuire:AVERage?....994
ACQuire:COUNt....994
ACQuire:CURRent?....995
ACQuire:INTerpolate....995
ACQuire:POINts[:VALue]....995
ACQuire:POINts:ARATe?....996
ACQuire:POINts:MAXimum....996
ACQuire:POINts:MODE....996
ACQuire:POMemory?....997
ACQuire:RESolution?....997
ACQuire:RLReal?....997
ACQuire:SRATe[:VALue]....998
ACQuire:SRATe:MINimum....998
ACQuire:SRATe:MODE....998
ACQuire:SRReal?....999
ACQuire:TYPE....999

**ACQuire:AVAilable?**

Number of acquisitions that is saved in the memory and available for history viewing. It is also the number of acquisitions in a fast segmentation acquisition series.

**Return values:**

<AcquisitionCount> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Available acqs" on page 243

**ACQuire:AVERage?**

Returns the current number of acquired waveforms that contribute to the average.

**Return values:**

<CurrAverageCount> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Current average count" on page 130

**ACQuire:COUNt** <MaxAcqCnt>

Sets the acquisition and average count, which has a double effect:

- It sets the number of waveforms acquired with `RUNSingle`.

- It defines the number of waveforms used to calculate the average waveform.

**Parameters:**

<MaxAcqCnt>
Range: 1 to 16777215
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "N-single/avg count" on page 129

---

**ACQuire:CURRent?**

Returns the current number of acquisitions that have been acquired.

**Return values:**

<CurrAcqCnt>
Range: 0 to 18446744073709551615
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**ACQuire:INTerpolate** <IntpolMd>

Selects the interpolation method.

**Parameters:**

<IntpolMd>
SINX | LINear | SMHD

**LINear**
Linear interpolation between two adjacent sample points

**SINX**
Interpolation with a sin(x)/x function.

**SMHD**
Sample/Hold causes a histogram-like interpolation.

*RST: SINX

**Usage:** Asynchronous command

**Manual operation:** See "Interpolation" on page 130

---

**ACQuire:POINts[:VALue]** <RecordLength>

Sets the record length, if `ACQuire:POINts:MODE` is set to `MANual`.

**Parameters:**

<RecordLength>
Range: 1000 to 500E+6
Increment: 2
*RST: 1000
Default unit: pts

**Usage:** Asynchronous command

**Manual operation:** See "Record length" on page 129

---

**ACQuire:POINts:ARATe?**

Returns the sample rate of the ADC, before waveform processing. The result is the interleaved sample rate or the non-interleaved one, depending on the channel usage.

**Return values:**

<ADCSampleRate> Range: 5 Gsample/s (interleaved), 2,5 Gsample/s (non-interleaved)
Default unit: Hz

**Usage:** Query only
Asynchronous command

---

**ACQuire:POINts:MAXimum** <RecLengthLimit>

Sets the maximum record length, if `ACQuire:POINts:MODE` is set to `AUTO`.

**Parameters:**

<RecLengthLimit> Range: 1000 to 500E+6
Increment: 2
*RST: 10E+6
Default unit: pts

**Usage:** Asynchronous command

**Manual operation:** See "Record length limit" on page 128

---

**ACQuire:POINts:MODE** <RecLengthMode>

Selects the mode of the waveform record length adjustment.

The record length is the number of waveform samples that are stored in one waveform record after processing, including interpolation. It determines the length of the displayed waveform.

**Parameters:**

<RecLengthMode> AUTO | MANual

**AUTO**
Record length is determined automatically and changes due to instrument internal adjustments.

**MANual**
The waveform record length is defined with `ACQuire:POINts[:VALue]`.

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "RL mode" on page 128

**ACQuire:POMemory?**

The command returns 1 if the memory is not sufficient to process the data with the current settings. To solve the problem, reduce the record length or use automatic record length setting.

**Return values:**

<OutOfMemory> OFF | ON
*RST: OFF

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Record length" on page 129

**ACQuire:RESolution?** [<MIN | MAX>]

Returns the current resolution.

The resolution is the time between two waveform samples in the waveform record. It considers the processing of the captured samples including interpolation. A fine resolution with low values produces a more precise waveform record. The resolution is the reciprocal of the sample rate.

You can query the minimum and maximum values with `<command>? MIN` and `<command>? MAX`.

**Return values:**

<Resolution> Range: 1E-15 to 100
Increment: 1E-11
*RST: 200E-12
Default unit: s

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Resolution" on page 129

**ACQuire:RLReal?**

Returns the internal record length used by the acquisition system.

**Return values:**

<HWRecordLength> Range: 2 to 4294967295
Increment: 1
*RST: 1000
Default unit: pts

**Usage:** Query only
Asynchronous command

**ACQuire:SRATe[:VALue]** <SampleRate>

Sets the number of waveform points per second if `ACQuire:SRATe:MODE` is set to MANual.

**Parameters:**

<SampleRate>
Range: 2 to 5E+12
Increment: 1
*RST: 5E+9
Default unit: Sa/s

**Usage:** Asynchronous command

**Manual operation:** See "Sample rate" on page 128

**ACQuire:SRATe:MINimum** <SampleRateMin>

Sets the minimum sample rate if `ACQuire:SRATe:MODE` is set to `AUTO`.

**Parameters:**

<SampleRateMin>
Range: 2 to 5E+12
Increment: 1
*RST: 2
Default unit: Sa/s

**Usage:** Asynchronous command

**Manual operation:** See "Min. sample rate" on page 128

**ACQuire:SRATe:MODE** <SampleRateMode>

Defines how the sample rate is set.

The sample rate considers the samples of the ADC, and the processing of the captured samples including interpolation.

**Parameters:**

<SampleRateMode> AUTO | MANual

**AUTO**
Sample rate is determined automatically and changes due to instrument internal adjustments. You can set a minimum sample rate with `ACQuire:SRATe:MINimum`.

**MANual**
The sample rate is defined with `ACQuire:SRATe[:VALue]`.

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "SR mode" on page 128

**ACQuire:SRReal?** [<MIN | MAX>]

Returns the sample rate of the waveform after HW processing. Interpolation is not considered. This value is shown in the acquisition label above the diagram.

You can query the minimum and maximum values with `<command>? MIN` and `<command>? MAX`.

**Return values:**

<HWSampleRate> Range: 2 to ADC sample rate
Increment: 1
*RST: 5E+9
Default unit: Sa/s

**Usage:** Query only
Asynchronous command

**ACQuire:TYPE** <AcquMode>

Sets how the waveform is built from the captured samples.

**Parameters:**

<AcquMode> SAMPle | PDETect | ENVelope | AVERage
*RST: SAMPle

**Usage:** Asynchronous command

**Manual operation:** See "Acquisition mode" on page 129

### 18.9.5 Vertical setup

The channel suffix <ch> selects the input channel that is affected by the command.

CHANnel<ch>:STATe ... 999
CHANnel<ch>:SCALe ... 1000
CHANnel<ch>:RANGe ... 1000
CHANnel<ch>:OFFSet ... 1000
CHANnel<ch>:POSition ... 1001
CHANnel<ch>:COUPling ... 1001
CHANnel<ch>:INVert ... 1002
DISPlay:SIGNal:LABel ... 1002
CHANnel<ch>:SKEW:TIME ... 1002
CHANnel<ch>:BANDwidth ... 1003
CHANnel<ch>:EATScale ... 1003
CHANnel<ch>:EATTenuation ... 1003
CHANnel<ch>:IMPedance ... 1004

**CHANnel<ch>:STATe** <State>

Switches the selected channel signal on or off.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "[C<n>]" on page 48

---

**CHANnel<ch>:SCALe** <Scale>

Sets the vertical scale, which defines the displayed amplitude of the selected waveform.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<Scale> Range: 0.001 to 1
Increment: 0.001
*RST: 0.05
Default unit: Depends on the connected probe

**Usage:** Asynchronous command

**Manual operation:** See "[Scale]" on page 49

---

**CHANnel<ch>:RANGe** <Range>

Sets the voltage range across the 10 vertical divisions of the diagram. The command is an alternative to `CHANnel<ch>:SCALe`.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<Range> Range: 0.01 to 10
Increment: 0.01
*RST: 0.5
Default unit: V/div

**Usage:** Asynchronous command

---

**CHANnel<ch>:OFFSet** <Offset>

Sets the offset voltage, which corrects an offset-affected signal. The vertical center of the selected channel is shifted by the offset value and the signal is repositioned within the diagram.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**

<Offset>
Range: -1 to 1
Increment: 0.01
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "[Position]" on page 48

---

**CHANnel<ch>:POSition** <Position>

Moves the selected signal up or down in the diagram. While the offset sets a voltage, position is a graphical setting given in divisions. The visual effect is the same as for offset.

**Suffix:**

<ch> 1 to 8, index of the analog channel

**Parameters:**

<Position> Positive values move up the waveform, negative values move it down.

Range: -5 to 5
Increment: 0.02
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "[Position]" on page 48

---

**CHANnel<ch>:COUPling** <UserCoupling>

Sets the connection of the channel signal, i.e. the input impedance (termination) and a filter (coupling). The command determines what part of the signal is used for waveform analysis and triggering.

**Suffix:**

<ch> 1 to 8, index of the analog channel

**Parameters:**

<UserCoupling> DC | DCLimit | AC

**DC**
Connection with 50 Ω termination, passes both DC and AC components of the signal.

**DCLimit**
Connection with 1 MΩ termination, passes both DC and AC components of the signal.

**AC**
Connection with 1 MΩ termination through DC capacitor, removes DC and very low-frequency components. The waveform is centered on zero volts.

*RST: Instrument specific

**Usage:** Asynchronous command

**Manual operation:** See "Coupling" on page 137

---

**CHANnel<ch>:INVert** <InvertChannel>

Turns the inversion of the signal amplitude on or off. To invert means to reflect the voltage values of all signal components against the ground level.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<InvertChannel> OFF | ON

ON: inverted waveform
OFF: normal waveform

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Invert channel" on page 138

---

**DISPlay:SIGNal:LABel** <Signal>,<Label>

Defines and assigns a label to the specified channel waveform.

**Parameters:**
<Label> String with the waveform label

**Setting parameters:**
<Signal> C1 | C2 | ... | C8

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 138

---

**CHANnel<ch>:SKEW:TIME** <Offset>

Sets a skew value to compensate for the delay of the measurement setup or from the circuit specifics that the instrument cannot compensate automatically. It affects only the selected input channel.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<Offset> Range: -100E-9 to 100E-9
Increment: 1E-13
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Skew" on page 138

---

**CHANnel<ch>:BANDwidth** <BandwidthLimit>

Sets the bandwidth limit. The specified bandwidth indicates the range of frequencies that the instrument can acquire and display accurately with less than 3 dB attenuation. Frequencies above the limit are removed from the signal, and noise is reduced.

To set the bandwidth, use one of the <BandwidthLimit> values. The query returns always a numeric result.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Setting parameters:**
<BandwidthLimit> FULL | B20 | B50 | B100 | B200 | B350 | B500 | B700

Availability depends on the maximum bandwidth of the instrument and installed bandwidth extension options. Instead of the given values, you can also set the corresponding numeric value in Hz.

FULL sets the bandwidth to the maximum bandwidth of the instrument. Bandwidth extension options are considered.
All other values set a bandwidth limit lower than the maximum. The number indicates the bandwidth limit in MHz.

**Return values:**
<Result> Numeric result in Hz

**Usage:** Asynchronous command

**Manual operation:** See "Bandwidth" on page 139

---

**CHANnel<ch>:EATScale** <ExtAttScl>

Sets the attenuation scale for an external divider: linear or logarithmic.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<ExtAttScl> LIN | LOG

*RST: LIN

**Usage:** Asynchronous command

**Manual operation:** See "External Attenuation: Scale, Attenuation" on page 140

---

**CHANnel<ch>:EATTenuation** <ExtAtt>

Consider a voltage divider that is part of the DUT before the measuring point. The external attenuation is included in the measurement, and the instrument shows the results that would be measured before the divider. External attenuation can be used with all probes.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<ExtAtt> Values depend on the selected scale (`CHANnel<ch>:EATScale` on page 1003) and the unit of the waveform. See "External Attenuation: Scale, Attenuation" on page 140. Limits below are for linear scale.

Range: 0.005 to 1000000
Increment: 0.01
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "External Attenuation: Scale, Attenuation" on page 140

---

**CHANnel<ch>:IMPedance** <Impedance>

Sets the impedance of the connected probe for power calculations and measurements.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<Impedance>

Range: 0.1 to 100000
Increment: 1
*RST: 50
Default unit: Ω

**Usage:** Asynchronous command

**Manual operation:** See "Impedance" on page 140

## 18.9.6 Probes

This section describes all remote commands for probes.


- Common probe settings....1004
- Settings for active voltage probes....1009
- Settings for modular probes....1015
- Settings for current probes....1019
- Optical isolated probes....1021
- Probe adapter R&S RT-Z2T....1024
- Probe attributes....1024
- Probe at external trigger input....1027


### 18.9.6.1 Common probe settings

The probe suffix <ch> selects the input channel to which the probe is connected.

PROBe<ch>:SETup:ATTenuation[:AUTO]? ..... 1005
PROBe<ch>:SETup:ATTenuation:MANual ..... 1005
PROBe<ch>:SETup:ATTenuation:MODE ..... 1005
PROBe<ch>:SETup:ATTenuation:DEFProbe ..... 1006
PROBe<ch>:SETup:ATTenuation:UNIT ..... 1007
PROBe<ch>:SETup:BANDwidth? ..... 1007
PROBe<ch>:SETup:OFFSet:TOMean ..... 1007
PROBe<ch>:SETup:OFFSet:AZERo ..... 1008
PROBe<ch>:SETup:OFFSet:USEautozero ..... 1008
PROBe<ch>:SETup:NAME? ..... 1008
PROBe<ch>:SETup:STATe? ..... 1009
PROBe<ch>:SETup:TYPE? ..... 1009

**PROBe<ch>:SETup:ATTenuation[:AUTO]?**

Returns the attenuation of a detected or predefined probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<Attenuation> Range: 0.0002 to 1000
Increment: 0.1
*RST: 1
Default unit: V/V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Attenuation" on page 142

**PROBe<ch>:SETup:ATTenuation:MANual** <Attenuation>

Sets the attenuation for an unknown probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<Attenuation> Range: 0.0001 to 10000
Increment: 0.1
*RST: 1
Default unit: V/V

**Usage:** Asynchronous command

**Manual operation:** See "Attenuation" on page 142

**PROBe<ch>:SETup:ATTenuation:MODE** <AttenuationMode>

Set the mode to MANual if the instrument does not detect the passive probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<AttenuationMode> AUTO | MANual
*RST: AUTO

**Usage:** Asynchronous command

---

**PROBe<ch>:SETup:ATTenuation:DEFProbe** <PredefinedProbe>

Selects one of the predefined probes, or a user-defined probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<PredefinedProbe> NONE | ZC10 | ZC20 | ZC30 | ZD01A100 | ZD01A1000 | HZ15 | HZ17 | RTZ25C1 | ZZ80 | ZS10L | ZD02 | ZD08 | ZC02100 | ZC021000 | ZC03 | ZD002A10 | ZD002A100 | ZD003A20 | ZD003A200 | ZC3110 | ZC311 | ZC3101 | ZH03 | ZP1X | CWT15 | CWT30 | CWTMN1 | CMN50HF3 | CMN50HF15 | CMN50HF30 | CMN50HF015 | CMN50HF06 | CWTUM015 | CWTUM06 | CWTUM1 | V1V1 | V10V1 | V100V1 | V200V1 | V1000V1 | V1A1000 | V1A100 | V1A10 | V1A1 | USER

**ZC10 | ZC20 | ZC30 | ZC03**
Current probes

**ZD02 | ZD08 | ZD01A100 | ZD01A1000 | ZD002A10 | ZD002A100 | ZD003A20 | ZD003A200**
Active differential probes, attenuation ratio according to the setting on the probe.
A100 = 100:1
A1000 = 1000:1

**HZ15 | HZ17**
Near field probes (EMI)

**ZC02100 | ZC021000 | ZC3110 | ZC311 | ZC3101**
Current probes 100 A/V or 1000 A/V according to the setting on the probe.

**CWT15 | CWT30 | CWTMN1 |CWTUM015 | CWTUM06 | CWTUM1 | CMN50HF3 | CMN50HF15 | CMN50HF30 | CMN50HF015 | CMN50HF06**
PEM Rogowski probes

**ZZ80 | ZH03 | ZP1X |**
Passive probes

**V1V1 | V10V1 | V100V1 | V200V1 | V1000V1 | V1A1000 | V1A100 | V1A10 | V1A1**
Common gain and attenuation settings if the probe type is not in the list.

**USER**
Probe is not detected and not known to the instrument. Set unit and attenuation manually.

*RST: NONE

**Usage:** Asynchronous command

**Manual operation:** See "Predefined probe, Type and Name of the probe" on page 141

---

**PROBe<ch>:SETup:ATTenuation:UNIT** <Unit>

Returns the unit of the connected probe if the probe is detected or predefined. For unknown probes, you can select the required unit.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<Unit> V | A | W

Voltage probe (V), current probe (A), power probe (W)

*RST: V

**Usage:** Asynchronous command

**Manual operation:** See "Vertical unit" on page 142

---

**PROBe<ch>:SETup:BANDwidth?**

Returns the bandwidth of the connected probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<Bandwidth> Range: 10000 to 20000000000
Increment: 10
*RST: 1500000000
Default unit: Hz

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Probe bandwidth" on page 142

---

**PROBe<ch>:SETup:OFFSet:TOMean**

Compensates automatically for a DC component of the input signal using the result of a background mean measurement.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Usage:** Event
Asynchronous command

**Manual operation:** See "Set offset to mean" on page 142

---

**PROBe<ch>:SETup:OFFSet:AZERo**

Measures the zero error of the probe. Short the signal pin and the ground pin together, then send the command.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Usage:** Event
Asynchronous command

**Manual operation:** See "AutoZero, Use AutoZero" on page 143

---

**PROBe<ch>:SETup:OFFSet:USEautozero** <UseAutoZeroOffset>

Corrects the zero error of the probe. The zero error is detected with `PROBe<ch>:SETup:OFFSet:AZERo`.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<UseAutoZeroOffset> OFF | ON
*RST: OFF

**Example:**
```
PROB2:SET:OFFS:AZER
PROB2:SET:OFFS:USE ON
```
Detects the zero error and uses it for correction.

**Usage:** Asynchronous command

**Manual operation:** See "AutoZero, Use AutoZero" on page 143

---

**PROBe<ch>:SETup:NAME?**

Queries the name of the probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<Name> Name string

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Predefined probe, Type and Name of the probe" on page 141

**PROBe<ch>:SETup:STATe?**

Queries if the probe at the specified input channel is active (detected) or not active (not detected). To switch the probe on, use `CHANnel<ch>:STATe`.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<State> DETected | NDETected
*RST: NDETected

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Predefined probe, Type and Name of the probe" on page 141

**PROBe<ch>:SETup:TYPE?**

Queries the class of the probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<Type> String containing the probe class, e.g. None (no probe detected), passive, current, active diff., active single-ended.

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Predefined probe, Type and Name of the probe" on page 141

### 18.9.6.2 Settings for active voltage probes

The probe suffix <ch> selects the input channel to which the probe is connected.

PROBe<ch>:SETup:MODE ..... 1010
PROBe<ch>:SETup:ACCoupling ..... 1011
PROBe<ch>:SETup:CMOFfset ..... 1011
PROBe<ch>:SETup:DISPlaydiff ..... 1011
PROBe<ch>:SETup:EPATtenuator ..... 1012
PROBe<ch>:PMETer:STATe ..... 1012
PROBe<ch>:PMETer:RESults:SINGle? ..... 1012
PROBe<ch>:PMETer:RESults:COMMon? ..... 1013
PROBe<ch>:PMETer:RESults:DIFFerential? ..... 1013
PROBe<ch>:PMETer:RESults:NEGative? ..... 1013
PROBe<ch>:PMETer:RESults:POSitive? ..... 1014
PROBe<ch>:SETup:ADVanced:AUDioverload ..... 1014

PROBe<ch>:SETup:ADVanced:FILTer ..... 1014
PROBe<ch>:SETup:ADVanced:RANGe ..... 1015
PROBe<ch>:SETup:ADVanced:PMToffset ..... 1015

**PROBe<ch>:SETup:MODE** <Mode>

The micro button is located on the probe head. Pressing this button, you initiate an action on the instrument directly from the probe. The button is disabled during internal automatic processes, for example, during self-alignment, autoset, and level detection.

Select the action that you want to start from the probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<Mode> RCONtinuous | RSINgle | AUToset | AZERo | OTMean | SITFile | NOACtion | FTRiglevel | PRSetup

**RCONtinuous**
Run continuous: the acquisition is running as long as the probe button is pressed.

**RSINgle**
Run single: starts a defined number of acquisitions (same as [Single] key).

**AUToset**
Starts the autoset procedure.

**AZERo**
AutoZero: performs an automatic correction of the zero error.

**OTMean**
Set offset to mean: performs an automatic compensation for a DC component of the input signal.

**SITFile**
Save image to file: saves the display image in a file.

**NOACtion**
Nothing is started on pressing the micro button.

**FTRiglevel**
Sets the trigger level automatically to *0.5 * (MaxPeak – MinPeak)*. The function is not available for an external trigger source.

**PRSetup**
Opens the "Probes Setup" dialog box.

*RST: RCONtinuous

**Usage:** Asynchronous command

**Manual operation:** See "MicroButton" on page 145

**PROBe<ch>:SETup:ACCoupling** <ProbeCouplingAC>

Enables AC coupling in R&S RT-ZPR power rail probes, which removes DC and very low-frequency components. The R&S RT-ZPR probe requires 50 Ω input termination, for which the channel AC coupling is not available. The probe setting allows AC coupling also at 50 Ω inputs.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<ProbeCouplingAC> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "AC Coupling" on page 148

**PROBe<ch>:SETup:CMOFfset** <CMOffset>

Sets the common-mode offset to compensate for a common DC voltage that is applied to both input sockets (referenced to the ground socket). The setting is available for Rohde & Schwarz differential probes and for modular probes in CM measurement mode.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<CMOffset> Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "CM offset" on page 146

**PROBe<ch>:SETup:DISPlaydiff** <DisplayDiff>

Selects the voltage to be measured by the R&S ProbeMeter of differential active probes:

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<DisplayDiff> DIFFerential | SINGleended

**DIFFerential**
Measures differential and common mode voltages

**SINGleended**
Measures the voltage between the positive/negative signal socket and the ground.

*RST: DIFFerential

**Usage:** Asynchronous command

**Manual operation:** See "Display" on page 147

---

**PROBe<ch>:SETup:EPATtenuator** <ProbeAttenuator>

If you use an external attenuator together with one of the differential active probes R&S RT-ZD10/20/30, select the attenuator used to include its attenuation in the measurements.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<ProbeAttenuator> NONE | ZA15 | ZA100
*RST: NONE

**Usage:** Asynchronous command

**Manual operation:** See "Probe attenuator" on page 147

---

**PROBe<ch>:PMETer:STATe** <State>

Activates the integrated R&S ProbeMeter on probes with Rohde & Schwarz probe interface.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "ProbeMeter" on page 145

---

**PROBe<ch>:PMETer:RESults:SINGle?**

Returns the R&S ProbeMeter measurement result of single-ended active Rohde & Schwarz probes, the voltage measured between the probe tip and the ground.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<Result> Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "ProbeMeter" on page 145

---

**PROBe<ch>:PMETer:RESults:COMMon?**

Returns the R&S ProbeMeter measurement result of differential active R&S probes: the common mode voltage, which is the mean voltage between the signal sockets and the ground socket.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<Result> Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "ProbeMeter" on page 145

---

**PROBe<ch>:PMETer:RESults:DIFFerential?**

Returns the R&S ProbeMeter measurement result of differential active Rohde & Schwarz probes, the differential voltage - the voltage between the positive and negative signal sockets.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<Result> Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "ProbeMeter" on page 145

---

**PROBe<ch>:PMETer:RESults:NEGative?**

Returns the R&S ProbeMeter measurement result of differential active R&S probes, the voltage that is measured between the negative signal socket and the ground.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<Result> Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "ProbeMeter" on page 145

---

**PROBe<ch>:PMETer:RESults:POSitive?**

Returns the R&S ProbeMeter measurement result of differential active R&S probes: the voltage that is measured between the positive signal socket and the ground.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<Result> Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "ProbeMeter" on page 145

---

**PROBe<ch>:SETup:ADVanced:AUDioverload** <Sound>

Activates the acoustic overrange warning in the probe control box. The command is relevant for R&S RT-ZHD probes.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<Sound> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Audible overrange" on page 150

---

**PROBe<ch>:SETup:ADVanced:FILTer** <State>

Activates the lowpass filter in the probe control box. The filter frequency depends on the probe type and is indicated on the probe control box.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Bandwidth limit" on page 150

---

**PROBe<ch>:SETup:ADVanced:RANGe** <ProbeRange>

Sets the voltage range of an R&S RT-ZHD probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<ProbeRange> AUTO | MHIGh | MLOW

**AUTO**
The voltage range is set with `CHANnel<ch>:SCALe`.

**MHIGh**
Sets the higher voltage range of the connected probe. To query the value, use `PROBe<ch>:SETup:ATTenuation[:AUTO]?`.

**MLOW**
Sets the lower voltage range of the connected probe. To query the value, use `PROBe<ch>:SETup:ATTenuation[:AUTO]?`.

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 150

---

**PROBe<ch>:SETup:ADVanced:PMToffset**

Sets the measured ProbeMeter value as offset. Thus, the value is considered in measurements.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Set offset to ProbeMeter value, ProbeMeter to offset" on page 148

### 18.9.6.3 Settings for modular probes

PROBe<ch>:SETup:PRMode....1016
PROBe<ch>:SETup:CMOFfset....1016
PROBe<ch>:SETup:DMOFfset....1017
PROBe<ch>:SETup:NOFFset....1017

PROBe<ch>:SETup:POFFset....1017
PROBe<ch>:SETup:TERM:ADJust....1018
PROBe<ch>:SETup:TERM:MEASure?....1018
PROBe<ch>:SETup:TERM:MODE....1018
PROBe<ch>:SETup:TERM:STATe....1019

**PROBe<ch>:SETup:PRMode** <MeasMode>

Sets the measurement mode of modular probes.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<PrbMeasMd> DMODe | CMODe | PMODe | NMODe

**DMODe**
Differential mode input voltage ($V_{dm}$), the voltage between the positive and negative input terminal.

**CMODe**
Common mode input voltage ($V_{cm}$), the mean voltage between the positive and negative input terminal vs. ground.

**PMODe**
Positive single-ended input voltage ($V_p$). the voltage between the positive input terminal and ground.

**NMODe**
Negative single-ended input voltage ($V_N$). the voltage between the negative input terminal and ground.

*RST: DMODe

**Usage:** Asynchronous command

**Manual operation:** See "Probe mode" on page 153

**PROBe<ch>:SETup:CMOFfset** <CMOffset>

Sets the common-mode offset to compensate for a common DC voltage that is applied to both input sockets (referenced to the ground socket). The setting is available for Rohde & Schwarz differential probes and for modular probes in CM measurement mode.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<CMOffset>
Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "CM offset" on page 146

**PROBe<ch>:SETup:DMOFfset** <DMOffset>

Sets the differential offset to compensate a DC voltage applied between the positive ($V_p$) and the negative ($V_n$) input terminal at the probe tip.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<DMOffset>
Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "DM offset, CM offset, P offset, N offset" on page 154

**PROBe<ch>:SETup:NOFFset** <NOffset>

Sets the negative offset to compensate a DC voltage applied to the negative input terminal ($V_n$) referenced to ground.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<NOffset>
Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "DM offset, CM offset, P offset, N offset" on page 154

**PROBe<ch>:SETup:POFFset** <POffset>

Sets the positive offset to compensate a DC voltage applied to the positive input terminal ($V_p$) referenced to ground.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<POffset>
Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "DM offset, CM offset, P offset, N offset" on page 154

**PROBe<ch>:SETup:TERM:ADJust** <VoltageAdj>

Activates control of the termination voltage.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<VoltageAdj> Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Adjustment" on page 155

**PROBe<ch>:SETup:TERM:MEASure?**

Returns the measured common mode voltage.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<VoltageMeas> Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Measurement" on page 155

**PROBe<ch>:SETup:TERM:MODE** <Mode>

Selects the voltage that is used for termination.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<Mode> AUTO | MANual

AUTO: the instrument uses the measured common mode voltage for termination.
MANual: enter the voltage to be used for termination with `PROBe<ch>:SETup:TERM:ADJust`.

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 155

**PROBe<ch>:SETup:TERM:STATe** <VoltageState>

Activates control of the termination voltage.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<VoltageState> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 155

### 18.9.6.4 Settings for current probes

The probe suffix <ch> selects the input channel to which the probe is connected.

PROBe<ch>:SETup:GAIN:AUTO? ..... 1019
PROBe<ch>:SETup:GAIN:MANual ..... 1019
PROBe<ch>:SETup:DEGauss ..... 1020
PROBe<ch>:SETup:OFFSet:ZADJust ..... 1020
PROBe<ch>:SETup:OFFSet:STPRobe ..... 1020

**PROBe<ch>:SETup:GAIN:AUTO?**

Returns the gain of a detected or predefined current probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<Gain> Range: 0.001 to 5000
Increment: 0.1
*RST: 1
Default unit: A/V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Gain, Manual gain" on page 151

**PROBe<ch>:SETup:GAIN:MANual** <Gain>

Sets the gain for an unknown current probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**

<Gain>
Range: 0.0001 to 10000
Increment: 0.0001
*RST: 1
Default unit: V/V

**Usage:** Asynchronous command

**Manual operation:** See "Gain, Manual gain" on page 151

---

**PROBe<ch>:SETup:DEGauss**

Demagnetizes the core if it has been magnetized by switching the power on and off, or by an excessive input. Always carry out demagnetizing before measurement.

**Suffix:**

<ch> 1 to 8, index of the analog channel

**Usage:** Event
Asynchronous command

**Manual operation:** See "DeGauss" on page 151

---

**PROBe<ch>:SETup:OFFSet:ZADJust** <ZeroAdjustValue>

Set the waveform to zero position. It corrects the effect of a voltage offset or temperature drift. To set the value by the instrument, use `PROBe<ch>:SETup:OFFSet:AZERo`.

**Suffix:**

<ch> 1 to 8, index of the analog channel

**Parameters:**

<ZroAdjVal>
Range: -100 to 100
Increment: 0.1
*RST: 0
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Zero adjust" on page 152

---

**PROBe<ch>:SETup:OFFSet:STPRobe**

Saves the zero adjust value in the probe box. If you connect the probe to another channel or to another Rohde & Schwarz oscilloscope, the value is read out again, and you can use the probe without further adjustment.

**Suffix:**

<ch> 1 to 8, index of the analog channel

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save to probe" on page 152

### 18.9.6.5 Optical isolated probes

The commands of this section are relevant for optical isolated probes. The commands are available in firmware version 2.2 and higher.

PROBe<ch>:SETup:ADVanced:UNIT ..... 1021
PROBe<ch>:SETup:ADVanced:PMToffset ..... 1021
PROBe<ch>:SETup:LASer:STATe? ..... 1021
PROBe<ch>:SETup:ALIGnment:GAIN:EXECute ..... 1022
PROBe<ch>:SETup:ALIGnment:ZERO:EXECute ..... 1022
PROBe<ch>:SETup:ALIGnment:WRITe ..... 1022
PROBe<ch>:SETup:ADVanced:RDEFaults ..... 1023
PROBe<ch>:SETup:TIPModel:NAME? ..... 1023
PROBe<ch>:SETup:TIPModel:INFO? ..... 1023

---

**PROBe<ch>:SETup:ADVanced:UNIT** <SelectUnit>

Sets the unit of the R&S RT-ZISO signal.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<SelectUnit> V | A | W
*RST: V

**Usage:** Asynchronous command

**Manual operation:** See "Select input unit" on page 156

---

**PROBe<ch>:SETup:ADVanced:PMToffset**

Sets the measured ProbeMeter value as offset. Thus, the value is considered in measurements.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Set offset to ProbeMeter value, ProbeMeter to offset" on page 148

---

**PROBe<ch>:SETup:LASer:STATe?**

Returns the current status of the laser.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<LaserState> 1 | 2 | 3

**1**
The laser is working.

**2**
The laser needs service, but is still working.

**3**
Defective laser, send to your Rohde & Schwarz service center.

*RST: 1

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Laser state" on page 157

---

**PROBe<ch>:SETup:ALIGnment:GAIN:EXECute**

Corrects the zero point error and the gain error of the R&S RT-ZISO probe.

To write the alignment result to the non-volatile flash of the probe, use `PROBe<ch>:SETup:ALIGnment:WRITe`.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Usage:** Event
Asynchronous command

**Manual operation:** See "Gain/Zero-error alignment" on page 158

---

**PROBe<ch>:SETup:ALIGnment:ZERO:EXECute**

Performs a self-alignment and corrects the zero point error of the R&S RT-ZISO probe.

To write the alignment result to the non-volatile flash of the probe, use `PROBe<ch>:SETup:ALIGnment:WRITe`.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Usage:** Event
Asynchronous command

**Manual operation:** See "Zero-error alignment" on page 157

---

**PROBe<ch>:SETup:ALIGnment:WRITe**

The command writes the alignment result to the non-volatile flash of the probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Usage:** Event
Asynchronous command

---

**PROBe<ch>:SETup:ADVanced:RDEFaults**

Resets the zero point and gain error correction to the factory default values.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Usage:** Event
Asynchronous command

**Manual operation:** See "Restore default values" on page 158

---

**PROBe<ch>:SETup:TIPModel:NAME?**

Returns the name of the tip module that is connected to the R&S RT-ZISO probe at the specified channel.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<ProbeTipModel> UNKNOWN | Z101 | Z201 | Z202 | Z203 | Z301 | Z302 | NONE
*RST: NONE

**Usage:** Query only
Asynchronous command

---

**PROBe<ch>:SETup:TIPModel:INFO?**

Return detailed information on the tip module that is connected to a R&S RT-ZISO probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<TipInfo> Comma-separated list of tip module attributes: Name, Type, PartNumber, SerialNumber, InputCapacity, InputImpedance, InputMin, InputMax, OffsetMin, OffsetMax, Unit, Bandwidth, Attenuation.

**Usage:** Query only
Asynchronous command

### 18.9.6.6 Probe adapter R&S RT-Z2T

PROBe<ch>:SETup:ADAPter....1024
PROBe<ch>:SETup:ATTenuation:TDEFprobe....1024

**PROBe<ch>:SETup:ADAPter** <Type>

Selects the adapter. The query returns the identified adapter type.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<Type> NONE | Z2T
*RST: NONE

**Usage:** Asynchronous command

**PROBe<ch>:SETup:ATTenuation:TDEFprobe** <TekPredefProbe>

Selects the Tektronix probe that is connected to the R&S RT-Z2T adapter.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Parameters:**
<TekPredefProbe> NONE | P5205A50 | P5205A500 | P5210A100 | P5210A1000 | P6205 | P6241 | P6243 | P6245 | P6246A1 | P6246A10 | P6247A1 | P6247A10 | P6248A1 | P6248A10 | P6249 | P6250A5 | P6250A50 | P6251A5 | P6251A50 | P6701B | P6703B | P6711 | P6713 | TCP202
*RST: NONE

**Example:**
```
PROBe2:SETup:ADAPter
<-- Z2T
PROBe2:SETup:ATTenuation:TDEFprobe P5205A50
```
Checks the adapter state on channel 2, and selects the probe P5205A50.

**Usage:** Asynchronous command

### 18.9.6.7 Probe attributes

PROBe<ch>:ID:PARTnumber?....1025
PROBe<ch>:ID:PRDate?....1025
PROBe<ch>:ID:SRNumber?....1025
PROBe<ch>:ID:SWVersion?....1025
PROBe<ch>:SETup:CAPacitance?....1026
PROBe<ch>:SETup:DCRange:MAX?....1026
PROBe<ch>:SETup:DCRange:MIN?....1026
PROBe<ch>:SETup:IMPedance?....1026

**PROBe<ch>:ID:PARTnumber?**

Queries the Rohde & Schwarz part number of the probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<PartNumber> Part number in a string.

**Usage:** Query only
Asynchronous command

**PROBe<ch>:ID:PRDate?**

Queries the production date of the probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<ProductionDate> Date in a string.

**Usage:** Query only
Asynchronous command

**PROBe<ch>:ID:SRNumber?**

Queries the serial number of the probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<SerialNo> Serial number in a string.

**Usage:** Query only
Asynchronous command

**PROBe<ch>:ID:SWVersion?**

Queries the version of the probe firmware.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<Softwareversion> Version number in a string.

**Usage:** Query only
Asynchronous command

**PROBe<ch>:SETup:CAPacitance?**

Queries the input capacitance of the probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<InputCapacity> Range: 1E-13 to 1E-07
Default unit: F

**Usage:** Query only
Asynchronous command

**PROBe<ch>:SETup:DCRange:MAX?**

Returns the maximum value of the dynamic DC range.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<DynDCrangeMax> Range: -1E+26 to 1E+26
Increment: 1E-12
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**PROBe<ch>:SETup:DCRange:MIN?**

Returns the minimum value of the dynamic DC range.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<DynDCrangeMin> Range: -1E+26 to 1E+26
Increment: 1E-12
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**PROBe<ch>:SETup:IMPedance?**

Queries the termination of the probe.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<InputImpedance> Range: 0 to 1E+9
Default unit: Ω

**Usage:** Query only
Asynchronous command

### 18.9.6.8 Probe at external trigger input

TRPRobe:SETup:ATTenuation:UNIT? ..... 1027
TRPRobe:SETup:ATTenuation[:AUTO]? ..... 1027
TRPRobe:SETup:NAME? ..... 1027
TRPRobe:SETup:STATe? ..... 1028
TRPRobe:SETup:TYPE? ..... 1028

**TRPRobe:SETup:ATTenuation:UNIT?**

Returns the unit of the probe that is connected to the external trigger input.

**Return values:**
<PrbAttUnt> V | A | W
*RST: V

**Usage:** Query only
Asynchronous command

**TRPRobe:SETup:ATTenuation[:AUTO]?**

Returns the attenuation of the probe that is connected to the external trigger input.

**Return values:**
<PrbAttMdAuto> Range: 0.001 to 1000
Increment: 0.1
*RST: 1
Default unit: V/V

**Usage:** Query only
Asynchronous command

**TRPRobe:SETup:NAME?**

Returns the name of the probe that is connected to the external trigger input.

**Return values:**
<Name> String parameter

**Usage:** Query only
Asynchronous command

**TRPRobe:SETup:STATe?**

Returns the state of the probe that is connected to the external trigger input.

**Return values:**
<State> DETected | NDETected
*RST: NDETected

**Usage:** Query only
Asynchronous command

**TRPRobe:SETup:TYPE?**

Returns the type of the probe that is connected to the external trigger input.

**Return values:**
<Type> String parameter

**Usage:** Query only
Asynchronous command

### 18.9.7 Fast segmentation

**ACQuire:SEGMented:MAX** <MaxAcqs>

If ON, the instrument acquires the maximum number of segments that can be stored in the memory. The maximum number depends on the current sample rate and record length settings.

If OFF, define the number of segments in a fast segmentation cycle with `ACQuire:COUNt`.

**Parameters:**
<MaxAcqs> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Acquire maximum" on page 131

**ACQuire:SEGMented:STATe** <State>

If fast segmentation is enabled, the acquisitions are performed as fast as possible, without processing and displaying the waveforms. When acquisition has been stopped, the data is processed and the latest waveform is displayed. Older waveforms are stored in segments. You can display and analyze the segments using the history.

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Fast segmentation" on page 132

## 18.9.8 High definition mode

HDEFinition:BFTYpe ..... 1029
HDEFinition:BWIDth ..... 1029
HDEFinition:RESolution? ..... 1029
HDEFinition:STATe ..... 1030

**HDEFinition:BFTYpe** <BwFilterCharacteristics>

Selects the filter that is used to reduce the bandwidth:

**Parameters:**

<BwFilterCharacteristics> GAUSsian | BESSel | BRICwall

*RST: BRICwall

**Usage:** Asynchronous command

**Manual operation:** See "Filter characteristics" on page 133

**HDEFinition:BWIDth** <Bandwidth>

Sets the filter bandwidth for the high definition mode.

**Parameters:**

<Bandwidth>
Range: 1000 to 500E+6
Increment: 1000
*RST: 100E+6
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Bandwidth" on page 133

**HDEFinition:RESolution?**

Displays the resulting vertical resolution in high definition mode. The higher the filter bandwidth, the lower the resolution.

**Return values:**

<Resolution>
Range: 0 to 18
Increment: 0.1
*RST: 0
Default unit: bit

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Resolution in bits" on page 133

**HDEFinition:STATe** <State>

Enables high definition mode, which increases the numeric resolution of the waveform signal.

**Parameters:**

<State> OFF | ON

`ON`: high definition mode
`OFF`: normal oscilloscope mode

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 133

### 18.9.9 Waveform data export

To transfer the waveform data to a remote computer, use:

- `CHANnel<ch>:DATA:HEADer?` on page 1142
- `CHANnel<ch>:DATA[:VALues]?` on page 1142

For fast export of several waveforms at once, use `EXPort:WAVeform:DATA[:VALues]?` on page 1150.

For export of waveform data to file, see Section 18.13.4, "Waveform data: export to file", on page 1127.

## 18.10 Trigger

Trigger commands use several suffixes.

**LEVel<n>, NOISe<m>**

The suffix indicates the analog channel for which the command takes effect. C1 has suffix 1, C2 has suffix 2, and so on.

**Event<ev>**

The suffix indicates the sequence step, for which the command takes effect when you trigger on a sequence. If you trigger on a single event, the suffix = 1 and can be omitted.

- 1 = A-trigger
- 2 = B-trigger
- 3 = R-trigger (reset event)

**Asynchronous commands**

Some of the commands in the following section are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, section "Command sequence and synchronization"

## 18.10.1 Common trigger settings

TRIGger:MEVents:MODE ..... 1031
TRIGger:EVENt<ev>:TYPE ..... 1031
TRIGger:EVENt<ev>:LEVel<n>[:VALue] ..... 1032
TRIGger:FINDlevel ..... 1032
TRIGger:EVENt<ev>:SOURce ..... 1032

**TRIGger:MEVents:MODE** <Class>

Selects, if you want to trigger on a single event, or on a series of events.

**Parameters:**

<Class> SINGle | SEQuence
*RST: SINGle

**Usage:** Asynchronous command

**Manual operation:** See "Trigger on" on page 165

**TRIGger:EVENt<ev>:TYPE** <Type>

Selects the trigger type. In a trigger sequence, the trigger type is set for each condition.

**Suffix:**

<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**

<Type> EDGE | GLITch | WIDTh | RUNT | WINDow | TIMeout | INTerval | SLEWrate | ANEDge | SETHold | STATe | PATTern
ANEDge = analog edge trigger is the only trigger type if the extern trigger source is used.
For SETHold, also DATatoclock can be used.
*RST: EDGE

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 166

**TRIGger:EVENt<ev>:LEVel<n>[:VALue]** <Level>

Sets the trigger level for the specified event and source (channel).

If the trigger source is serial bus, the trigger level is set by the thresholds in the protocol configuration.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

<n> 1 to 8, index of the analog channel

**Parameters:**
<Level> Range: -10 to 10
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "[Level]" on page 46

**TRIGger:FINDlevel**

Sets the trigger level automatically to *0.5 * (MaxPeak – MinPeak)*.

In a trigger sequence, "Find level" affects all active events of the sequence (A, B, and R event).

**Usage:** Event
Asynchronous command

**Manual operation:** See "Find level" on page 167

**TRIGger:EVENt<ev>:SOURce** <SourceDetailed>

Selects the source of the trigger signal for the selected trigger event. The trigger source works even if it is not displayed in a diagram.

Available sources depend on the trigger sequence setting. If you trigger on a single event, all inputs can be used as trigger source. If you trigger on a sequence, only analog channels can be set as trigger source for A, B, and R-events.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<SourceDetailed> **C1 | C2 | ... | C8**
Available for single event, and A, B and R-events in a trigger sequence

**EXTernanalog | LINE |**
Available for single event (suffix 1)

**D0 | D1 | D2 | ... | D14 | D15**
Digital channels, require MSO option. Available for single event (suffix 1)

**SBUS1 | SBUS2 | SBUS3 | SBUS4**
Available if one or more serial protocol options are installed. If the hardware trigger is supported for a protocol, triggering on single event is possible with hardware trigger settings. For all protocols, the software trigger is supported in a "A → Serial bus" (event 2).

*RST: C1

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 165

## 18.10.2 Trigger sequence

TRIGger:MEVents:AEVents....1033
TRIGger:MEVents:SEQuence<se>:COUNt....1033
TRIGger:MEVents:SEQuence<se>:DELay....1034
TRIGger:MEVents:SEQuence<se>:RESet:EVENt....1034
TRIGger:MEVents:SEQuence<se>:RESet:TIMeout:TIME....1034
TRIGger:MEVents:SEQuence<se>:RESet:TIMeout[:ENABle]....1035

**TRIGger:MEVents:AEVents** <Type>

Selects the type of the trigger sequence.

**Parameters:**

<Type> AONLy | ABR | AZ | ASB

AONLy = single event, same as `TRIGger:MEVents:MODE SINGle`
ABR = sequence A → B → R
AZ = sequence A → Zone trigger
ASB = sequence A → Serial bus

*RST: AONLy

**Usage:** Asynchronous command

**Manual operation:** See "Trigger sequence" on page 168

**TRIGger:MEVents:SEQuence<se>:COUNt** <Events>

Sets the number of B-trigger conditions to be fulfilled after an A-trigger. The last B-trigger causes the trigger event. The waiting time for B-triggers can be restricted with a reset condition: timeout or reset event.

**Suffix:**

<se> 1..3
2

**Parameters:**

<Events> Range: 1 to 2147483647
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "B event count" on page 169

---

**TRIGger:MEVents:SEQuence<se>:DELay** <Delay>

Sets the time that the instrument waits after an A-trigger until it recognizes B-triggers.

**Suffix:**

<se> 1

**Parameters:**

<Delay> Range: 0 to 50
Increment: 1E-12
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Delay" on page 169

---

**TRIGger:MEVents:SEQuence<se>:RESet:EVENt** <State>

If enabled, the trigger sequence is restarted by the R-trigger condition if the specified number of B-triggers does not occur before the R-trigger conditions are fulfilled.

**Suffix:**

<se> 3

**Parameters:**

<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable reset event" on page 169

---

**TRIGger:MEVents:SEQuence<se>:RESet:TIMeout:TIME** <ResetTimeout>

The time the instrument waits for the number of B-events specified using `TRIGger:MEVents:SEQuence<se>:COUNt`, before the sequence is restarted with the A-trigger.

**Suffix:**

<se> Irrelevant, omit the suffix.

**Parameters:**

<ResetTimeout>
Range: 0 to 50
Increment: 1E-12
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Enable reset by time, Reset timeout" on page 169

---

**TRIGger:MEVents:SEQuence<se>:RESet:TIMeout[:ENABle]** <State>

If set to ON, the instrument waits for the time defined using `TRIGger:MEVents:SEQuence<se>:RESet:TIMeout:TIME` for the specified number of B-events. If no trigger occurs during that time, the sequence is restarted with the A-event.

**Suffix:**

<se> Irrelevant, omit the suffix.

**Parameters:**

<State>
OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable reset by time, Reset timeout" on page 169

## 18.10.3 Edge trigger

TRIGger:EVENt<ev>:EDGE:SLOPe....1035
TRIGger:ANEDge:LEVel....1036
TRIGger:ANEDge:COUPling....1036
TRIGger:ANEDge:FILTer....1036
TRIGger:ANEDge:CUToff:HIGHpass....1036
TRIGger:ANEDge:CUToff:LOWPass....1037
TRIGger:ANEDge:NREJect....1037

---

**TRIGger:EVENt<ev>:EDGE:SLOPe** <Slope>

Sets the edge direction for the trigger.

**Suffix:**

<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**

<Slope>
POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 170

**TRIGger:ANEDge:LEVel** <ExtTrigLev>

Sets the trigger level for the external trigger source.

**Parameters:**

<ExtTrigLev> Range: - 5 to 5
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Level" on page 166

**TRIGger:ANEDge:COUPling** <Coupling>

Sets the connection of the external trigger signal, i.e. the input impedance and a termination. The coupling determines what part of the signal is used for triggering.

**Parameters:**

<Coupling> DC | DCLimit | AC

**DC**
Connection with 50 Ω termination, passes both DC and AC components of the signal.

**DCLimit**
Connection with 1 MΩ termination, passes both DC and AC components of the signal.

**AC**
Connection with 1 MΩ termination through DC capacitor, removes DC and very low-frequency components. The waveform is centered on zero volts.

*RST: DCLimit

**Usage:** Asynchronous command

**Manual operation:** See "Coupling" on page 172

**TRIGger:ANEDge:FILTer** <Filter>

Selects the filter mode for the external trigger signal.

**Parameters:**

<Filter> OFF | LFReject | RFReject

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Filter, Cut-off" on page 172

**TRIGger:ANEDge:CUToff:HIGHpass** <CutOffFreq>

Frequencies below the cutoff frequency are rejected, higher frequencies pass the filter.

**Parameters:**

<CutOffFreq> KHZ5 | KHZ50

KHZ5 = 5 kHz
KHZ50 = 50 kHz

*RST: KHZ50

**Usage:** Asynchronous command

**Manual operation:** See "Filter, Cut-off" on page 172

---

**TRIGger:ANEDge:CUToff:LOWPass** <CutOffFreq>

Frequencies higher than the cutoff frequency are rejected, lower frequencies pass the filter.

**Parameters:**

<CutOffFreq> KHZ50 | MHZ50

KHZ50 = 50 kHz
MHZ50 = 50 MHz

*RST: KHZ50

**Usage:** Asynchronous command

**Manual operation:** See "Filter, Cut-off" on page 172

---

**TRIGger:ANEDge:NREJect** <NoiseReject>

Enables an automatic hysteresis on the trigger level to avoid unwanted trigger events caused by noise.

**Parameters:**

<NoiseReject> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Trigger noise reject" on page 172

### 18.10.4 Glitch trigger

TRIGger:EVENt<ev>:GLITch:POLarity ..... 1037
TRIGger:EVENt<ev>:GLITch:RANGe ..... 1038
TRIGger:EVENt<ev>:GLITch:WIDTh ..... 1038

---

**TRIGger:EVENt<ev>:GLITch:POLarity** <Polarity>

Sets the polarity of a pulse, which is the direction of the first pulse slope.

**Suffix:**

<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<Polarity> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 173

---

**TRIGger:EVENt<ev>:GLITch:RANGe** <RangeMode>

Selects which glitches are identified: shorter or longer than the width specified using `TRIGger:EVENt<ev>:GLITch:WIDTh`.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<RangeMode> SHORter | LONGer

**SHORter**
Glitches shorter than the specified width are identified.

**LONGer**
Glitches longer than the specified width are identified.

*RST: SHORter

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 173

---

**TRIGger:EVENt<ev>:GLITch:WIDTh** <Width>

Sets the length of a glitch. The instrument triggers on pulses shorter or longer than this value, depending on the value set with `TRIGger:EVENt<ev>:GLITch:RANGe`.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<Width> Range: 1E-10 to 10000
Increment: 0.0001
*RST: 1E-09
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Width" on page 173

### 18.10.5 Width trigger

TRIGger:EVENt<ev>:WIDTh:DELTa ..... 1039
TRIGger:EVENt<ev>:WIDTh:POLarity ..... 1039
TRIGger:EVENt<ev>:WIDTh:RANGe ..... 1039
TRIGger:EVENt<ev>:WIDTh:WIDTh ..... 1039

**TRIGger:EVENt<ev>:WIDTh:DELTa** <WidthDelta>

Defines a range around the width value.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<WidthDelta> Range: 0 to 432
Increment: 5E-10
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "±Delta" on page 175

**TRIGger:EVENt<ev>:WIDTh:POLarity** <Polarity>

Sets the polarity of a pulse, which is the direction of the first pulse slope.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<Polarity> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 175

**TRIGger:EVENt<ev>:WIDTh:RANGe** <RangeMode>

Selects how the range of a pulse width is defined.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<RangeMode> WITHin | OUTSide | SHORter | LONGer
*RST: WITHin

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 175

**TRIGger:EVENt<ev>:WIDTh:WIDTh** <Width>

Depending on the values of `TRIGger:EVENt<ev>:WIDTh:RANGe` the width sets:

- For the ranges `Within` and `Outside`, the width defines the center of a time range which is defined by the limits "±Delta" (see `TRIGger:EVENt<ev>:WIDTh:DELTa`).

- For the ranges `Shorter` and `Longer`, it defines the maximum and minimum time lapse, respectively.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<Width>
Range: 1E-10 to 10000
Increment: 1E-07
*RST: 5E-09
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Width" on page 175

### 18.10.6 Runt trigger

TRIGger:EVENt<ev>:LEVel<n>:RUNT:LOWer ..... 1040
TRIGger:EVENt<ev>:LEVel<n>:RUNT:UPPer ..... 1040
TRIGger:EVENt<ev>:RUNT:DELTa ..... 1041
TRIGger:EVENt<ev>:RUNT:POLarity ..... 1041
TRIGger:EVENt<ev>:RUNT:RANGe ..... 1041
TRIGger:EVENt<ev>:RUNT:WIDTh ..... 1042

---

**TRIGger:EVENt<ev>:LEVel<n>:RUNT:LOWer** <Level>

Sets the lower voltage limit.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

<n> 1 to 8, index of the analog channel

**Parameters:**
<Level>
Range: -10 to 10
Increment: 0.001
*RST: -0.1
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Lower level" on page 177

---

**TRIGger:EVENt<ev>:LEVel<n>:RUNT:UPPer** <Level>

Sets the upper voltage limit.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

<n> 1 to 8, index of the analog channel

**Parameters:**

<Level>

Range: -10 to 10
Increment: 0.001
*RST: 0.1
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level" on page 176

---

**TRIGger:EVENt<ev>:RUNT:DELTa** <WidthDelta>

Defines a range around the runt width specified using `TRIGger:EVENt<ev>:RUNT:WIDTh`.

Available if `TRIGger:EVENt<ev>:RUNT:RANGe` is set to `WITHin` or `OUTSide`.

**Suffix:**

<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**

<WidthDelta>

Range: 1E-10 to 864
Increment: 1E-07
*RST: 1E-10
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "±Delta" on page 177

---

**TRIGger:EVENt<ev>:RUNT:POLarity** <Polarity>

Sets the polarity of a pulse, which is the direction of the first pulse slope.

**Suffix:**

<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**

<Polarity>

POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 173

---

**TRIGger:EVENt<ev>:RUNT:RANGe** <Mode>

Defines the time limit of the runt pulse in relation to the `TRIGger:EVENt<ev>:RUNT:WIDTh` and `TRIGger:EVENt<ev>:RUNT:DELTa` settings.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<Mode> ANY | LONGer | SHORter | WITHin | OUTSide

**ANY**
Triggers on all runts fulfilling the level condition, without time limitation.

**LONGer**
Triggers on runts longer than the given runt width.

**SHORter**
Triggers on runts shorter than the given runt width.

**WITHin**
Triggers if the runt length is inside a given time range. The range is defined by runt width and ±Delta.

**OUTSide**
Triggers if the runt length is outside a given time range. The range is defined by runt width and ±Delta.

*RST: ANY

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 177

---

**TRIGger:EVENt<ev>:RUNT:WIDTh** <Width>

Defines the upper or lower voltage threshold.

It is not available if `TRIGger:EVENt<ev>:RUNT:RANGe` is set to `ANY`.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<Width>
Range: 1E-10 to 10000
Increment: 1E-07
*RST: 5E-09
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Runt width" on page 177

## 18.10.7 Window trigger

TRIGger:EVENt<ev>:LEVel<n>:WINDow:LOWer ... 1043
TRIGger:EVENt<ev>:LEVel<n>:WINDow:UPPer ... 1043
TRIGger:EVENt<ev>:WINDow:DELTa ... 1043
TRIGger:EVENt<ev>:WINDow:RANGe ... 1044
TRIGger:EVENt<ev>:WINDow:TIME ... 1044
TRIGger:EVENt<ev>:WINDow:WIDTh ... 1045

**TRIGger:EVENt<ev>:LEVel<n>:WINDow:LOWer** <Level>

Sets the lower voltage limit.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

<n> 1 to 8, index of the analog channel

**Parameters:**
<Level>
Range: -10 to 10
Increment: 0.001
*RST: -0.1
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Lower level" on page 179

**TRIGger:EVENt<ev>:LEVel<n>:WINDow:UPPer** <Level>

Sets the upper voltage limit.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

<n> 1 to 8, index of the analog channel

**Parameters:**
<Level>
Range: 0 to 1000
Increment: 1E-06
*RST: 0.1
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level" on page 178

**TRIGger:EVENt<ev>:WINDow:DELTa** <WidthDelta>

Defines a range around the width value.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<WidthDelta>
Range: 0 to 432
Increment: 5E-10
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "±Delta" on page 179

**TRIGger:EVENt<ev>:WINDow:RANGe** <RangeMode>

Selects how the signal run is compared with the window.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<RangeMode> ENTer | EXIT | WITHin | OUTSide

**ENTer**
Triggers when the signal crosses the upper or lower level and thus enters the window made up of these two levels.

**EXIT**
Triggers when the signal leaves the window.

**WITHin**
Triggers if the signal stays between the upper and lower level for a specified time. The time is defined with `TRIGger:EVENt<ev>:WINDow:TIME`.

**OUTSide**
Triggers if the signal stays above the upper level or below the lower level for a specified time. The time is defined with `TRIGger:EVENt<ev>:WINDow:TIME`.

*RST: ENTer

**Usage:** Asynchronous command

**Manual operation:** See "Vertical condition" on page 178

**TRIGger:EVENt<ev>:WINDow:TIME** <TimeRangeMode>

Available for `TRIGger:EVENt<ev>:WINDow:RANGe` = `WITHin` and `OUTSide`.

Selects how the time limit of the window is defined.

You can specify the width with `TRIGger:EVENt<ev>:WINDow:WIDTh` and the delta with `TRIGger:EVENt<ev>:WINDow:DELTa`.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<TimeRangeMode> WITHin | OUTSide | SHORter | LONGer

**WITHin**
Triggers if the signal stays inside or outside the vertical window limits at least for the time *Width - Delta* and for *Width + Delta* at the most.

**OUTSide**
"Outside" is the opposite definition of "Within". The instrument triggers if the signal stays inside or outside the vertical window limits for a time shorter than *Width - Delta* or longer than *Width + Delta*.

**SHORter**
Triggers if the signal crosses vertical limits before the specified width time is reached.

**LONGer**
Triggers if the signal crosses vertical limits before the specified width time is reached.

*RST: WITHin

**Usage:** Asynchronous command

**Manual operation:** See "Time condition" on page 179

---

**TRIGger:EVENt<ev>:WINDow:WIDTh** <Width>

Depending on the values of `TRIGger:EVENt<ev>:WINDow:RANGe` the width sets:

- For the ranges `Within` and `Outside`, the width defines the center of a time range. The range is defined by the limits "±Delta", see `TRIGger:EVENt<ev>:WINDow:DELTa`.
- For the ranges `Shorter` and `Longer`, it defines the maximum and minimum time lapse, respectively.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<Width>
Range: 1E-10 to 10000
Increment: 1E-07
*RST: 5E-09
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Width" on page 179

## 18.10.8 Timeout trigger

TRIGger:EVENt<ev>:TIMeout:RANGe ..... 1045
TRIGger:EVENt<ev>:TIMeout:TIME ..... 1046

---

**TRIGger:EVENt<ev>:TIMeout:RANGe** <TimeoutMode>

Sets the relation of the signal level to the trigger level for the timeout trigger.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**

<TimeoutMode> HIGH | LOW | EITHer

HIGH = stays high, the signal level stays above the trigger level.
LOW = stays low, the signal level stays below the trigger level.
EITHer = stays high or low.

*RST: HIGH

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 180

---

**TRIGger:EVENt<ev>:TIMeout:TIME** <Time>

Sets the time limit for the timeout at which the instrument triggers.

**Suffix:**

<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**

<Time>
Range: 1E-10 to 10000
Increment: 1E-07
*RST: 1E-07
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Time" on page 180

## 18.10.9 Interval trigger

TRIGger:EVENt<ev>:INTerval:DELTa....1046
TRIGger:EVENt<ev>:INTerval:RANGe....1047
TRIGger:EVENt<ev>:INTerval:SLOPe....1047
TRIGger:EVENt<ev>:INTerval:WIDTh....1047

---

**TRIGger:EVENt<ev>:INTerval:DELTa** <WidthDelta>

Sets a range around the interval width value specified with `TRIGger:EVENt<ev>:INTerval:WIDTh`.

**Suffix:**

<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**

<WidthDelta>
Range: 0 to 10
Increment: 1E-07
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "±Delta" on page 182

**TRIGger:EVENt<ev>:INTerval:RANGe** <RangeMode>

Defines the range of an interval in relation to the interval width specified using `TRIGger:EVENt<ev>:INTerval:WIDTh` and `TRIGger:EVENt<ev>:INTerval:DELTa`.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<RangeMode> WITHin | OUTSide | SHORter | LONGer

**WITHin**
Triggers on pulses inside a given range. The range is defined by the interval width ±delta.

**OUTSide**
Triggers on pulses outside a given range. The range is defined by the interval width ±delta.

**SHORter**
Triggers on pulses shorter than the given interval width.

**LONGer**
Triggers on pulses longer than the given interval width.

*RST: OUTSide

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 181

**TRIGger:EVENt<ev>:INTerval:SLOPe** <Slope>

Sets the edge for the trigger. You can analyze the interval between positive edges or between negative edges.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<Slope> POSitive | NEGative | EITHer

*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 181

**TRIGger:EVENt<ev>:INTerval:WIDTh** <Width>

Sets the time between two pulses for the interval trigger.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**

<Width>

Range: 1E-10 to 10000
Increment: 1E-07
*RST: 5E-09
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Interv. width" on page 182

## 18.10.10 Slew rate trigger

TRIGger:EVENt<ev>:LEVel<n>:SLEW:LOWer....1048
TRIGger:EVENt<ev>:LEVel<n>:SLEW:UPPer....1048
TRIGger:EVENt<ev>:SLEW:DELTa....1049
TRIGger:EVENt<ev>:SLEW:RANGe....1049
TRIGger:EVENt<ev>:SLEW:RATE....1050
TRIGger:EVENt<ev>:SLEW:SLOPe....1050

---

**TRIGger:EVENt<ev>:LEVel<n>:SLEW:LOWer** <Level>

Sets the lower voltage threshold. When the signal crosses this level, the slew rate measurement starts or stops depending on the selected slope.

**Suffix:**

<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

<n> 1 to 8, index of the analog channel

**Parameters:**

<Level>

Range: -10 to 10
Increment: 0.001
*RST: -0.1
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Lower level" on page 183

---

**TRIGger:EVENt<ev>:LEVel<n>:SLEW:UPPer** <Level>

Sets the upper voltage threshold. When the signal crosses this level, the slew rate measurement starts or stops depending on the selected slope.

**Suffix:**

<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

<n> 1 to 8, index of the analog channel

**Parameters:**

<Level>
Range: -10 to 10
Increment: 0.001
*RST: 0.1
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level" on page 183

---

**TRIGger:EVENt<ev>:SLEW:DELTa** <TimeDelta>

Defines a time range around the given slew rate.

**Suffix:**

<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**

<TimeDelta>
Range: 0 to 10
Increment: 1E-07
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "±Delta" on page 184

---

**TRIGger:EVENt<ev>:SLEW:RANGe** <RangeMode>

Selects how the time limit for the slew rate is defined. The time measurement starts when the signal crosses the first trigger level - the upper or lower limit depending on the selected slope. The measurement stops when the signal crosses the second level.

You can select the rate with `TRIGger:EVENt<ev>:SLEW:RATE` and set the delta with `TRIGger:EVENt<ev>:SLEW:DELTa`.

**Suffix:**

<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**

<RangeMode> INSRange | OUTRange | LTHan | GTHan

**INSRange**
Triggers on pulses inside a given range. The range is defined by the slew rate ±delta.

**OUTRange**
Triggers on pulses outside a given range. The range is defined by the slew rate ±delta.

**LTHan**
Triggers on pulses shorter than the given slew rate.

**GTHan**
Triggers on pulses longer than the given slew rate.

*RST: GTHan

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 184

---

**TRIGger:EVENt<ev>:SLEW:RATE** <Time>

For `TRIGger:EVENt<ev>:SLEW:RANGe` = `INSRange` and `OUTRange`, the slew rate defines the center of a range which is defined by the limits "±Delta".

For `TRIGger:EVENt<ev>:SLEW:RANGe` = `LTHan` and `GTHan`, the slew rate defines the maximum and minimum slew rate limits, respectively. When the signal crosses this level, the slew rate measurement starts or stops depending on the selected slope (see `TRIGger:EVENt<ev>:SLEW:SLOPe`).

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<Time>
Range: 1E-10 to 10000
Increment: 1E-07
*RST: 1E-10
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Slew rate" on page 184

---

**TRIGger:EVENt<ev>:SLEW:SLOPe** <Slope>

Sets the edge direction for the trigger.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<Slope> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 183

## 18.10.11 Setup & Hold

TRIGger:EVENt<ev>:SETHold:CSOurce:EDGE....1050
TRIGger:EVENt<ev>:SETHold:CSOurce:LEVel....1051
TRIGger:EVENt<ev>:SETHold:CSOurce[:VALue]....1051
TRIGger:EVENt<ev>:SETHold:HTIMe....1051
TRIGger:EVENt<ev>:SETHold:STIMe....1052

---

**TRIGger:EVENt<ev>:SETHold:CSOurce:EDGE** <ClockEdge>

Sets the edge of the clock signal. Edge and level define the time reference point.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<ClockEdge> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Clock edge" on page 185

---

**TRIGger:EVENt<ev>:SETHold:CSOurce:LEVel** <ClockLevel>

Sets the voltage level for the clock signal.

Both the clock level and the clock edge define the starting point for calculation of the setup and hold time.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<ClockLevel> Range: -10 to 10
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Clock level" on page 186

---

**TRIGger:EVENt<ev>:SETHold:CSOurce[:VALue]** <ClockSource>

Selects the input channel of the clock signal.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<ClockSource> C1 | C2 | ... | C8
The following values are also accepted:
C1 = CHAN1 = CHANnel1, C2 = CHAN2 = CHANnel2,
C3 = CHAN3 = CHANnel3, C4 = CHAN4 = CHANnel4
*RST: C1

**Usage:** Asynchronous command

**Manual operation:** See "Clock source" on page 185

---

**TRIGger:EVENt<ev>:SETHold:HTIMe** <HoldTime>

Sets the minimum time **after** the clock edge while the data signal must stay steady above or below the data level.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<HoldTime>
Range: -9.9999E-08 to 1E-07
Increment: 1E-09
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Hold time" on page 186

---

**TRIGger:EVENt<ev>:SETHold:STIMe** <SetupTime>

Sets the minimum time **before** the clock edge while the data signal must stay steady above or below the data level.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<SetupTime>
Range: -9.9999E-08 to 1E-07
Increment: 1E-09
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Setup time" on page 186

## 18.10.12 State trigger

TRIGger:EVENt<ev>:STATe:QUALify:ANALog:CHAN<n>:HLX ..... 1052
TRIGger:EVENt<ev>:STATe:QUALify:LOGic ..... 1053
TRIGger:EVENt<ev>:STATe:SLOPe ..... 1053
TRIGger:EVENt<ev>:STATe:QUALify:DIGital:LOGic ..... 1053
TRIGger:EVENt<ev>:STATe:QUALify:DIGital:CHAN<n>:HLX ..... 1054

---

**TRIGger:EVENt<ev>:STATe:QUALify:ANALog:CHAN<n>:HLX** <HLX>

Set the state for each channel. For the state trigger, the clock source is indicated and does not get a state.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

<n> 1 to 8, index of the analog channel

**Parameters:**
<HLX> HIGH | LOW | DONTcare
State of the individual channels
*RST: DONTcare

**Example:**
```
TRIG:EVEN1:SOUR C1
TRIG:EVEN1:SLOP POS
TRIG:EVEN1:STAT:QUAL:ANAL:CHAN2:HLX HIGH
TRIG:EVEN1:STAT:QUAL:ANAL:CHAN3:HLX LOW
TRIG:EVEN1:STAT:QUAL:ANAL:CHAN4:HLX HIGH
TRIG:EVEN1:STAT:QUAL:LOG AND
```

**Usage:** Asynchronous command

**Manual operation:** See "Source: channel states" on page 190

---

**TRIGger:EVENt<ev>:STATe:QUALify:LOGic** <StateOperator>

Defines the logic combination of the channels and their states.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<StateOperator> AND | OR
*RST: AND

**Usage:** Asynchronous command

**Manual operation:** See "Logic Operator" on page 190

---

**TRIGger:EVENt<ev>:STATe:SLOPe** <Slope>

Sets the edge direction for the trigger.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<Slope> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 187

---

**TRIGger:EVENt<ev>:STATe:QUALify:DIGital:LOGic** <Logic>

Defines the logic combination of the channels and their states.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<Logic> AND | OR
*RST: AND

**Usage:** Asynchronous command

**Manual operation:** See "Logic Operator" on page 190

**TRIGger:EVENt<ev>:STATe:QUALify:DIGital:CHAN<n>:HLX** <HLX>

Sets the required state for each digital channel that is used for triggering.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

<n> 0 to 15, index of the logic channel

**Parameters:**
<HLX> HIGH | LOW | DONTcare
*RST: DONTcare

**Usage:** Asynchronous command

**Manual operation:** See "Source: channel states" on page 191

### 18.10.13 Pattern trigger

TRIGger:EVENt<ev>:PATTern:QUALify:ANALog:CHAN<n>:HLX ..... 1054
TRIGger:EVENt<ev>:PATTern:QUALify:DIGital:CHAN<n>:HLX ..... 1054
TRIGger:EVENt<ev>:PATTern:QUALify:LOGic ..... 1055
TRIGger:EVENt<ev>:PATTern:QUALify:DIGital:LOGic ..... 1055
TRIGger:EVENt<ev>:PATTern:QUALify:SOURces ..... 1055

**TRIGger:EVENt<ev>:PATTern:QUALify:ANALog:CHAN<n>:HLX** <HLX>

Set the state for each channel. For the state trigger, the clock source is indicated and does not get a state.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

<n> 1 to 8, index of the analog channel

**Parameters:**
<HLX> HIGH | LOW | DONTcare
State of the individual channels
*RST: DONTcare

**Example:**
```
TRIG:EVEN1:PATT:QUAL:ANAL:CHAN1:HLX LOW
TRIG:EVEN1:PATT:QUAL:ANAL:CHAN2:HLX HIGH
TRIG:EVEN1:PATT:QUAL:ANAL:CHAN3:HLX LOW
TRIG:EVEN1:PATT:QUAL:ANAL:CHAN4:HLX HIGH
TRIG:EVEN1:PATT:QUAL:LOG AND
```

**Usage:** Asynchronous command

**Manual operation:** See "Source: channel states" on page 190

**TRIGger:EVENt<ev>:PATTern:QUALify:DIGital:CHAN<n>:HLX** <HLX>

Sets the required state for each digital channel that is used for triggering.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

<n> 0 to 15, index of the logic channel

**Parameters:**
<HLX> HIGH | LOW | DONTcare
*RST: DONTcare

**Usage:** Asynchronous command

**Manual operation:** See "Source: channel states" on page 191

---

**TRIGger:EVENt<ev>:PATTern:QUALify:LOGic** <StateOperator>

Defines the logic combination of the channels and their states.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<StateOperator> AND | OR
*RST: AND

**Usage:** Asynchronous command

**Manual operation:** See "Logic Operator" on page 190

---

**TRIGger:EVENt<ev>:PATTern:QUALify:DIGital:LOGic** <Logic>

Defines the logic combination of the channels and their states.

**Suffix:**
<ev> 1..3

**Parameters:**
<Logic> AND | OR
*RST: AND

**Usage:** Asynchronous command

**Manual operation:** See "Logic Operator" on page 190

---

**TRIGger:EVENt<ev>:PATTern:QUALify:SOURces** <StateSources>

Selects if the source of the trigger for the pattern type is an analog or digital channel.

**Suffix:**
<ev> 1 = A-trigger, 2 = B-trigger, 3 = reset event

**Parameters:**
<StateSources> ANALog | DIGital
*RST: ANALog

**Usage:** Asynchronous command

**Manual operation:** See "State sources" on page 188

## 18.10.14 Trigger mode, holdoff

TRIGger:MODE ..... 1056
TRIGger:FORCe ..... 1056
TRIGger:HOLDoff:MODE ..... 1057
TRIGger:HOLDoff:AUTotime? ..... 1057
TRIGger:HOLDoff:SCALing ..... 1058
TRIGger:HOLDoff:EVENts ..... 1058
TRIGger:HOLDoff:MAX ..... 1058
TRIGger:HOLDoff:MIN ..... 1059
TRIGger:HOLDoff:TIME ..... 1059

---

**TRIGger:MODE** <TriggerMode>

Sets the trigger mode which determines the behavior of the instrument with and without a trigger event.

**Parameters:**

<TriggerMode> AUTO | NORMal | FREerun

**AUTO**
The instrument triggers repeatedly after a time interval if the trigger conditions are not fulfilled. If a real trigger occurs, it takes precedence. The time interval depends on the time base.

**NORMal**
The instrument acquires a waveform only if a trigger occurs.

**FREerun**
The instrument triggers after a very short time interval - faster than in AUTO mode. Real triggers are ignored.

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "[Auto Norm]" on page 46

---

**TRIGger:FORCe**

Provokes an immediate single acquisition. Force the trigger if the acquisition is running in normal mode and no valid trigger occurs. Thus, you can confirm that a signal is available and use the waveform display to determine how to trigger on it.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Force trigger" on page 193

**TRIGger:HOLDoff:MODE** <Mode>

Selects the method to define the holdoff condition.

The trigger holdoff defines when the next trigger after the current will be recognized. Thus, it affects the next trigger to occur after the current one. Holdoff helps to obtain stable triggering when the oscilloscope is triggering on undesired events.

Holdoff settings are not available if the trigger source is an external trigger input or serial bus, and if you trigger on a sequence of events.

**Parameters:**

<Mode> TIME | EVENts | RANDom | AUTO | OFF

**TIME**
Defines the holdoff directly as a time period. The next trigger occurs only after the holdoff time has passed, which is defined using `TRIGger:HOLDoff:TIME`).

**EVENts**
Defines the holdoff as a number of trigger events. The next trigger occurs only when this number of events is reached. The number of triggers to be skipped is defined with `TRIGger:HOLDoff:EVENts`.

**RANDom**
Defines the holdoff as a random time limited by `TRIGger:HOLDoff:MIN` and `TRIGger:HOLDoff:MAX`. For each acquisition cycle, the instrument selects a new random holdoff time from the specified range.

**AUTO**
The holdoff time is calculated automatically based on the current horizontal scale.

**OFF**
No holdoff

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Holdoff mode" on page 193

**TRIGger:HOLDoff:AUTotime?**

Returns the resulting holdoff time, if `TRIGger:HOLDoff:MODE` is set to `AUTO`: *Auto time = Auto time scaling * Horizontal scale*. The auto time scaling factor is defined with `TRIGger:HOLDoff:SCALing`.

**Return values:**

<AutoTime>
Range: 1E-07 to 10
Increment: 0.0002
*RST: 0.001
Default unit: s

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Holdoff mode" on page 193

---

**TRIGger:HOLDoff:SCALing** <AutoTimeScl>

Sets the auto time scaling factor that the horizontal scale is multipied with, if `TRIGger:HOLDoff:MODETRIGger:HOLDoff:MODETRIGger:HOLDoff:MODE` is set to `AUTO`.
*Auto time = Auto time scaling * Horizontal scale*

The next trigger occurs only after this time has passed.

**Parameters:**
<AutoTimeScl>
Range: 0.001 to 1000
Increment: 1
*RST: 0.5

**Usage:** Asynchronous command

**Manual operation:** See "Holdoff mode" on page 193

---

**TRIGger:HOLDoff:EVENts** <Events>

Defines the number of triggers to be skipped, if `TRIGger:HOLDoff:MODETRIGger:HOLDoff:MODETRIGger:HOLDoff:MODE` is set to `EVENts`. The next trigger only occurs when this number of events is reached.

**Parameters:**
<Events>
Range: 1 to 2147483647
Increment: 10
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Holdoff mode" on page 193

---

**TRIGger:HOLDoff:MAX** <RandomMaxTime>

Defines the upper limit for the random time holdoff, if `TRIGger:HOLDoff:MODE TRIGger:HOLDoff:MODETRIGger:HOLDoff:MODE` is set to `RANDom`.

**Parameters:**
<RandomMaxTime>
Range: 1E-07 to 10
Increment: 0.0002
*RST: 0.002
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Holdoff mode" on page 193

**TRIGger:HOLDoff:MIN** <RandomMinTime>

Defines the lower limit for the random time holdoff, if `TRIGger:HOLDoff:MODE` `TRIGger:HOLDoff:MODETRIGger:HOLDoff:MODE` is set to `RANDom`.

**Parameters:**

<RandomMinTime>
Range: 1E-07 to 5
Increment: 0.0002
*RST: 0.001
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Holdoff mode" on page 193

**TRIGger:HOLDoff:TIME** <Time>

Defines the holdoff time period, if `TRIGger:HOLDoff:MODE` is set to `TIME`. The next trigger occurs only after this time has passed.

**Parameters:**

<Time>
Range: 1E-07 to 10
Increment: 0.0002
*RST: 0.001
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Holdoff mode" on page 193

## 18.10.15 Hysteresis

TRIGger:NOISe<m>:ABSolute ..... 1059
TRIGger:NOISe<m>:EFFective? ..... 1060
TRIGger:NOISe<m>:MODE ..... 1060
TRIGger:NOISe<m>:PERDivision ..... 1060
TRIGger:NOISe<m>:RELative ..... 1061
TRIGger:NOISe<m>[:STATe] ..... 1061

**TRIGger:NOISe<m>:ABSolute** <Absolute>

Defines a range in absolute values around the trigger level. If the signal oscillates inside this range and thus crosses the trigger level, no trigger event occurs.

**Suffix:**

<m> 1 to 8, index of the analog channel

**Parameters:**

<Absolute>
Range: 0 to 10000000
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Absolute hysteresis" on page 196

---

**TRIGger:NOISe<m>:EFFective?**

Returns the hysteresis that is set by the instrument in automatic hysteresis mode.

**Suffix:**
<m> 1 to 8, index of the analog channel

**Return values:**
<Effective> numeric value

Range: 0 to 10000000
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "HW hysteresis" on page 195

---

**TRIGger:NOISe<m>:MODE** <Mode>

Selects whether the hysteresis is defined in absolute or relative values. The setting is available only in manual hysteresis mode.

**Suffix:**
<m> 1 to 8, index of the analog channel

**Parameters:**
<Mode> ABS | REL

*RST: ABS

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis mode" on page 195

---

**TRIGger:NOISe<m>:PERDivision** <InDivision>

Defines a range in divisions around the trigger level in division units. If the signal oscillates inside this range and crosses the trigger level thereby, no trigger event occurs.

**Suffix:**
<m> 1 to 8, index of the analog channel

**Parameters:**
<InDivision> Range: 0 to 5
Increment: 0.01
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Relative hysteresis" on page 196

---

**TRIGger:NOISe<m>:RELative** <Relative>

Defines a range in divisions around the trigger level as percentage. If the signal oscillates inside this range and crosses the trigger level thereby, no trigger event occurs.

**Suffix:**
<m> 1 to 8, index of the analog channel

**Parameters:**
<Relative>
Range: 0 to 50
Increment: 1
*RST: 0
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Relative hysteresis" on page 196

---

**TRIGger:NOISe<m>[:STATe]** <Mode>

Selects how the hysteresis is set.

**Suffix:**
<m> 1 to 8, index of the analog channel

**Parameters:**
<Mode> AUTO | MANual

**AUTO**
Automatic mode is the recommended mode. The hysteresis is set by the instrument to reject the internal noise of the instrument.

**MANual**
The hysteresis is defined with `TRIGger:NOISe<m>:ABSolute` or `TRIGger:NOISe<m>:RELative`.

*RST: AUTO

**Usage:** Asynchronous command

## 18.10.16 Channel filter

TRIGger:FILTermode....1061
TRIGger:LFReject....1062
TRIGger:RFReject....1062

---

**TRIGger:FILTermode** <Mode>

Selects the filter mode for the trigger channel.

**Parameters:**

<Mode> OFF | LFReject | RFReject
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Trigger filter mode" on page 196

---

**TRIGger:LFReject** <Bandwidth>

Sets the limit frequency limit for the highpass filter of the trigger signal. Frequencies lower than this value are rejected, higher frequencies pass the filter.

**Parameters:**

<Bandwidth> Range: 50 kHz
*RST: 50 kHz
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "LF reject BW" on page 197

---

**TRIGger:RFReject** <Bandwidth>

Sets the limit frequency limit for the lowpass filter of the trigger signal. Frequencies higher than this value are rejected, lower frequencies pass the filter.

**Parameters:**

<Bandwidth> Range: 1E+3 to 500E+6
Increment: 1000
*RST: 1E+6
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "RF reject BW" on page 197

## 18.10.17 Actions on trigger

TRIGger:ACTions:BEEP ..... 1063
TRIGger:ACTions:OUT:STATe ..... 1063
TRIGger:ACTions:OUT:SOURce ..... 1063
TRIGger:ACTions:OUT:DELay ..... 1063
TRIGger:ACTions:OUT:PLENgth ..... 1064
TRIGger:ACTions:OUT:POLarity ..... 1064
TRIGger:ACTions:SCReenshot ..... 1064
TRIGger:ACTions:SRESults ..... 1064
TRIGger:ACTions:STOP ..... 1065
TRIGger:ACTions:WFMSave ..... 1065

**TRIGger:ACTions:BEEP** <Beep>

Generates a beep sound if the command is set to `TRIGger`.

**Parameters:**

<Beep> NOACtion | TRIGger
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Beep" on page 197

**TRIGger:ACTions:OUT:STATe** <State>

Activates the outgoing pulse on the [Trigger Out] connector on the rear panel.

If ON, a pulse is sent out each time when a trigger occurs.

**Parameters:**

<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Trigger out pulse" on page 198

**TRIGger:ACTions:OUT:SOURce** <SignalSource>

Defines when the trigger out signal is initiated: at the trigger point, when waiting for the trigger, or when the post-trigger time is finished.

**Parameters:**

<SignalSource> TRIG | TRIGGER | POST | POSTTRIGGER | WAIT | WAITTRIGGER
TRIG = TRIGGER, POST = POSTTRIGGER, WAIT = WAIT-TRIGGER
*RST: TRIG

**Usage:** Asynchronous command

**Manual operation:** See "Signal source" on page 199

**TRIGger:ACTions:OUT:DELay** <Delay>

Defines the delay of the first pulse edge to the trigger point. The minimum delay is 600 ns.

**Parameters:**

<Delay> Range: 8E-07 to 1
Increment: 4E-09
*RST: 8E-07
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Delay" on page 199

---

**TRIGger:ACTions:OUT:PLENgth** <PulseLength>

Sets the length of the trigger out pulse.

**Parameters:**

<PulseLength>
Range: 1.6E-08 to 0.05
Increment: 1.6E-08
*RST: 9.6E-08
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Pulse length" on page 199

---

**TRIGger:ACTions:OUT:POLarity** <Polarity>

Sets the polarity of the trigger out pulse, which is the direction of the first pulse edge.

**Parameters:**

<Polarity>
POSitive | NEGative
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 199

---

**TRIGger:ACTions:SCReenshot** <SaveScreenshot>

Saves a screenshot at each trigger if the command is set to `TRIGger`.

To configure the screenshot settings, use the commands described in Section 18.13.8, "Screenshots", on page 1137.

**Parameters:**

<SaveScreenshot>
NOACtion | TRIGger
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Screenshot" on page 198

---

**TRIGger:ACTions:SRESults** <SaveResults>

Saves the selected result to file at each trigger if the command is set to `TRIGger`.

To define the path and file names, use the EXPort:WAVeform:AUTonaming:* commands:

- `EXPort:WAVeform:AUTonaming:NAME` on page 1127
- `EXPort:WAVeform:AUTonaming:PATH` on page 1127

- EXPort:WAVeform:AUTonaming:TYPE on page 1128

**Parameters:**
<SaveResults> NOACtion | TRIGger
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Save results" on page 198

---

**TRIGger:ACTions:STOP** <StopAcq>

Stops the running acquisition if the command is set to TRIGger.

**Parameters:**
<StopAcq> NOACtion | TRIGger
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Stop acq" on page 198

---

**TRIGger:ACTions:WFMSave** <SaveWfm>

Saves the waveform data to file at each trigger if the command is set to TRIGger.

To define the path and file names, use the EXPort:WAVeform:AUTonaming:* commands:

- EXPort:WAVeform:AUTonaming:NAME on page 1127
- EXPort:WAVeform:AUTonaming:PATH on page 1127
- EXPort:WAVeform:AUTonaming:TYPE on page 1128

**Parameters:**
<SaveWfm> NOACtion | TRIGger
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Save wfm" on page 198

## 18.10.18 Zone trigger

Zone trigger commands are available since FW version 2.2.x.x.

ZONE<m>:ADD ..... 1066
TRIGger:ZONE:EXPRession[:DEFine] ..... 1066
ZONE<m>:ACOMbination ..... 1066
ZONE<m>:AREA<n>:ADD ..... 1067
ZONE<m>:AREA<n>:COUNt? ..... 1067
ZONE<m>:AREA<n>:INTersect ..... 1067
ZONE<m>:AREA<n>:LABel ..... 1068
ZONE<m>:AREA<n>:VALid? ..... 1068

ZONE<m>:AREA<n>:POINt<o>:ADD ..... 1068
ZONE<m>:AREA<n>:POINt<o>:COUNt? ..... 1069
ZONE<m>:AREA<n>:POINt<o>:REMove ..... 1069
ZONE<m>:AREA<n>:POINt<o>:X ..... 1069
ZONE<m>:AREA<n>:POINt<o>:Y ..... 1069
ZONE<m>:AREA<n>:POINt<o>:VALid? ..... 1070
ZONE<m>:AREA<n>:REMove ..... 1070
ZONE<m>:AREA<n>:STATe ..... 1070
ZONE<m>:COUNt? ..... 1071
ZONE<m>:DIAGram ..... 1071
TRIGger:ZONE:HISTory ..... 1071
ZONE<m>:REMove ..... 1072
ZONE<m>:SOURce ..... 1072
ZONE<m>[:VISible] ..... 1072

**ZONE<m>:ADD**

Adds a new trigger zone.

**Suffix:**
<m> 1 to 4, index of the zone

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add zone, " on page 207

**TRIGger:ZONE:EXPRession[:DEFine]** <LogicExpression>

Defines the zone trigger. The available operators for the combination between the zones are `AND | NOT | OR | XOR`.

**Parameters:**
<LogicExpression> String with the logical expression

**Example:** Prerequisite: Zone1 and Zone2 are defined.

```
TRIGger:ZONE:EXPRession 'Zone1 and Zone2'
TRIGger:ZONE:EXPRession?
<-- Zone1 and Zone2
```

**Usage:** Asynchronous command

**Manual operation:** See "[Zone]" on page 47

**ZONE<m>:ACOMbination** <AreaCombination>

Sets the logic combination that applies to all areas in the indicated zone.

**Suffix:**
<m> 1 to 4, index of the zone

**Parameters:**
<AreaCombination> AND | OR
*RST: AND

**Usage:** Asynchronous command

**Manual operation:** See "Area" on page 207

---

**ZONE<m>:AREA<n>:ADD**

Adds a new area to the trigger zone.

**Suffix:**
<m> 1 to 4, index of the zone

<n> 1 to 8, index of the zone area

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Area" on page 207

---

**ZONE<m>:AREA<n>:COUNt?**

Returns the number of the defined areas in the zone.

`ZONE<m>:AREA:COUNt? MAX` returns the maximum number of areas that can be created.

**Suffix:**
<m> 1 to 4, index of the zone

<n> Irrelevant, omit the suffix.

**Return values:**
<Count> Number of defined zone areas

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Area" on page 207

---

**ZONE<m>:AREA<n>:INTersect** <Intersection>

Defines if the signal must intersect the zone to allow the instrument to trigger, or if it must not intersect the zone.

**Suffix:**
<m> 1 to 4, index of the zone

<n> 1 to 8, index of the zone area

**Parameters:**
<Intersection> MUST | NOT
*RST: MUST

**Usage:** Asynchronous command

**Manual operation:** See "Area" on page 207

---

**ZONE<m>:AREA<n>:LABel** <Label>

Defines a label for the selected area.

**Suffix:**
<m> 1 to 4, index of the zone

<n> 1 to 8, index of the zone area

**Parameters:**
<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Area" on page 207

---

**ZONE<m>:AREA<n>:VALid?**

Checks the validity of the indicated zone area. The area is invalid if one of its points is invalid. See Figure 7-13.

**Suffix:**
<m> 1 to 4, index of the zone

<n> 1 to 8, index of the zone area

**Return values:**
<Valid> OFF | ON

*RST: ON

**Usage:** Query only
Asynchronous command

---

**ZONE<m>:AREA<n>:POINt<o>:ADD**

Adds a new point to the area at the indicated point index.

To define the coordinates of the point, use `ZONE<m>:AREA<n>:POINt<o>:X` and `ZONE<m>:AREA<n>:POINt<o>:Y`.

**Suffix:**
<m> 1 to 4, index of the zone

<n> 1 to 8, index of the zone area

<o> 1 to 16, index of the zone area point

**Usage:** Setting only
Asynchronous command

**ZONE<m>:AREA<n>:POINt<o>:COUNt?**

Queries the number of the defined points in the area.

`ZONE<m>:AREA<n>:POINt:COUNt? MAX` returns the maximum number of points that can be created.

**Suffix:**

<m> 1 to 4, index of the zone

<n> 1 to 8, index of the zone area

<o> Irrelevant, omit the suffix.

**Return values:**

<Count> Range: 0 to 16

**Usage:** Query only
Asynchronous command

**ZONE<m>:AREA<n>:POINt<o>:REMove**

Removes the selected point from the area.

**Suffix:**

<m> 1 to 4, index of the zone

<n> 1 to 8, index of the zone area

<o> 1 to 16, index of the zone area point

**Usage:** Setting only
Asynchronous command

**ZONE<m>:AREA<n>:POINt<o>:X** <X>

Sets the horizontal X coordinates for the selected point of the area.

**Suffix:**

<m> 1 to 4, index of the zone

<n> 1 to 8, index of the zone area

<o> 1 to 16, index of the zone area point

**Parameters:**

<X> Range: -1E+26 to 1E+26
Increment: 1E-06
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**ZONE<m>:AREA<n>:POINt<o>:Y** <Y>

Sets the vertical Y coordinates for the selected point of the area.

**Suffix:**
<m> 1 to 4, index of the zone

<n> 1 to 8, index of the zone area

<o> 1 to 16, index of the zone area point

**Parameters:**
<Y> Range: -1E+26 to 1E+26
Increment: 1E-06
*RST: 0
Default unit: V

**Usage:** Asynchronous command

---

**ZONE<m>:AREA<n>:POINt<o>:VALid?**

Checks the validity of the selected point. See Figure 7-13.

**Suffix:**
<m> 1 to 4, index of the zone

<n> 1 to 8, index of the zone area

<o> 1 to 16, index of the zone area point

**Return values:**
<Valid> OFF | ON
*RST: ON

**Usage:** Query only
Asynchronous command

---

**ZONE<m>:AREA<n>:REMove**

Removes the selected area from the trigger zone.

**Suffix:**
<m> 1 to 4, index of the zone

<n> 1 to 8, index of the zone area

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Delete icon" on page 208

---

**ZONE<m>:AREA<n>:STATe** <State>

Enables the selected area.

**Suffix:**
<m> 1 to 4, index of the zone

<n> 1 to 8, index of the zone area

**Parameters:**
<State> OFF | ON
*RST: ON

**Usage:** Asynchronous command

---

**ZONE<m>:COUNt?** <Count>

Returns the number of zones.

`ZONE:COUNt? MAX` returns the maximum number of zones that can be created.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Return values:**
<Count> Number of defined zones

**Usage:** Query only
Asynchronous command

---

**ZONE<m>:DIAGram** <SignDiagKey>

Selects the diagram on which the zone trigger is applied, for example layoutset1 diagram1 (`L1_D1`).

For more information about the SmartGrid definition, see Section 18.8.3, "SmartGrid", on page 957.

**Suffix:**
<m> 1 to 4, index of the zone

**Parameters:**
<SignDiagKey> String that indicates the layout set and the diagram, e.g. "L1_D1".

**Usage:** Asynchronous command

---

**TRIGger:ZONE:HISTory** <ApplyToHistory>

Applies the zone trigger condition to the acquisitions that are stored in the history memory. Thus, you can filter the history of waveforms on zone conditions.

**Parameters:**
<ApplyToHistory> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Apply zone trigger to history" on page 210

**ZONE<m>:REMove**

Removes the selected trigger zone.

**Suffix:**
<m> 1 to 4, index of the zone

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Delete icon" on page 208

**ZONE<m>:SOURce** <Source>

Sets the source of the zone trigger.

**Suffix:**
<m> 1 to 4, index of the zone

**Parameters:**
<Source> NONE | C1 | C2 | ... | C8 | M1 | M2 | ... | M8 | XY1 | XY2 | XY3 | XY4 | SPECNORM1 | SPECNORM2 | SPECNORM3 | SPECNORM4

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 207

**ZONE<m>[:VISible]** <DisplayState>

Enables the display of the zone on the screen.

**Suffix:**
<m> 1 to 4, index of the zone

**Parameters:**
<DisplayState> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Display" on page 207

# 18.11 Signal configuration

## 18.11.1 Reference levels

- Reference level settings....1073
- Reference level results....1077

### 18.11.1.1 Reference level settings

REFLevel<rl>:ENABle ... 1073
REFLevel<rl>:LMODe ... 1073
REFLevel<rl>:AUTO:MODE ... 1073
REFLevel<rl>:ABSolute:HYSTeresis ... 1074
REFLevel<rl>:ABSolute:LLEVel ... 1074
REFLevel<rl>:ABSolute:MLEVel ... 1075
REFLevel<rl>:ABSolute:ULEVel ... 1075
REFLevel<rl>:RELative:HYSTeresis ... 1075
REFLevel<rl>:RELative:LOWer ... 1076
REFLevel<rl>:RELative:MIDDle ... 1076
REFLevel<rl>:RELative:MODE ... 1076
REFLevel<rl>:RELative:UPPer ... 1077

**REFLevel<rl>:ENABle** <First>

Enables the specified reference level.

**Suffix:**
<rl> 1 to 8, index of the reference level set

**Parameters:**
<First> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "RLx" on page 212

**REFLevel<rl>:LMODe** <LevelMode>

Defines if the reference level is set in absolute or relative values.

**Suffix:**
<rl> 1 to 8, index of the reference level set

**Parameters:**
<LevelMode> ABS | REL
*RST: REL

**Usage:** Asynchronous command

**Manual operation:** See "Level mode" on page 212

**REFLevel<rl>:AUTO:MODE** <AutoLevelMode>

Defines the high and low signal levels from which the reference levels are derived.

**Suffix:**
<rl> 1 to 8, index of the reference level set

**Parameters:**
<AutoLevelMode> AUTO | PPRobability | MPRobability | ABSolutepeak

**AUTO**
Auto select absolute probability: most suitable signal levels

**PPRobability**
Peak probability: signal levels with the highest probability value

**MPRobability**
Mean probability: signal levels with mean probability

**ABSolutepeak**
Absolute peak: absolute peak signal levels

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Auto level mode" on page 213

---

**REFLevel<rl>:ABSolute:HYSTeresis** <HystAbs>

Defines a hysteresis for the middle reference level. A rise or fall from the middle reference value that does not exceed the hysteresis is rejected as noise.

**Suffix:**
<rl> 1 to 8, index of the reference level set

**Parameters:**
<HystAbs>
Range: 0 to 1E+26
Increment: 0.001
*RST: 0.005
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis" on page 213

---

**REFLevel<rl>:ABSolute:LLEVel** <LowerLevel>

Sets the lower reference level in absolute values. This is required, e.g., to determine a fall.

**Suffix:**
<rl> 1 to 8, index of the reference level set

**Parameters:**
<LowerLevel>
Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level, Lower level" on page 212

**REFLevel<rl>:ABSolute:MLEVel** <MiddleLevel>

Sets the middle reference level in absolute values.

**Suffix:**
<rl> 1 to 8, index of the reference level set

**Parameters:**
<MiddleLevel>
Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level, Lower level" on page 212

**REFLevel<rl>:ABSolute:ULEVel** <UpperLevel>

Sets the upper reference level in absolute values. This is required to determine a rise.

**Suffix:**
<rl> 1 to 8, index of the reference level set

**Parameters:**
<UpperLevel>
Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level, Lower level" on page 212

**REFLevel<rl>:RELative:HYSTeresis** <HystRel>

Defines a hysteresis for the middle reference level. A rise or fall from the middle reference value that does not exceed the hysteresis is rejected as noise.

**Suffix:**
<rl> 1 to 8, index of the reference level set

**Parameters:**
<HystRel>
Range: 0 to 50
Increment: 1
*RST: 10
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis" on page 213

**REFLevel<rl>:RELative:LOWer** <LowRefLevRel>

Sets the lower relative reference level if `REFLevel<rl>:RELative:MODE` is set to `USER`.

**Suffix:**
<rl> 1 to 8, index of the reference level set

**Parameters:**
<LowRefLevRel>
Range: -100 to 200
Increment: 1
*RST: 10
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level,Lower level" on page 213

**REFLevel<rl>:RELative:MIDDle** <MidRefLevRel>

Sets the middle relative reference level if `REFLevel<rl>:RELative:MODE` is set to `USER`.

**Suffix:**
<rl> 1 to 8, index of the reference level set

**Parameters:**
<MidRefLevRel>
Range: -100 to 200
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level,Lower level" on page 213

**REFLevel<rl>:RELative:MODE** <RelativeLevels>

The lower, middle and upper reference levels, defined as percentages of the high signal level.

**Suffix:**
<rl> 1 to 8, index of the reference level set

**Parameters:**
<RelativeLevels> FIVE | TEN | TWENty | USER

**FIVE**
5/50/95

**TEN**
10/50/90

**TWENty**
20/50/80

**USER**
Set the reference levels to individual values with `REFLevel<rl>:RELative:LOWer`, `REFLevel<rl>:RELative:MIDDle`, and `REFLevel<rl>:RELative:UPPer`.
*RST: TEN

**Usage:** Asynchronous command

**Manual operation:** See "Relative levels" on page 213

---

**REFLevel<rl>:RELative:UPPer** <UppRefLevRel>

Sets the upper relative reference level if `REFLevel<rl>:RELative:MODE` is set to `USER`.

**Suffix:**
<rl> 1 to 8, index of the reference level set

**Parameters:**
<UppRefLevRel>
Range: -100 to 200
Increment: 1
*RST: 90
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level,Lower level" on page 213

### 18.11.1.2 Reference level results

MEASurement<mg>:REFLevel<rl>:RESult:LOWer? 1077
MEASurement<mg>:REFLevel<rl>:RESult:MIDDle? 1077
MEASurement<mg>:REFLevel<rl>:RESult:UPPer? 1077
MEASurement<mg>:REFLevel<rl>:RESult:SIGHigh? 1078
MEASurement<mg>:REFLevel<rl>:RESult:SIGLow? 1078

---

**MEASurement<mg>:REFLevel<rl>:RESult:LOWer?**
**MEASurement<mg>:REFLevel<rl>:RESult:MIDDle?**
**MEASurement<mg>:REFLevel<rl>:RESult:UPPer?**

Return the lower, middle, and upper reference level, respectively.

**Suffix:**
<mg> 1 to 16, index of the measurement

<rl> Irrelevant, omit the suffix.

**Return values:**
<Value>
Range: -1E+26 to 1E+26
Increment: 0
*RST: 0

**Usage:** Query only
Asynchronous command

**MEASurement<mg>:REFLevel<rl>:RESult:SIGHigh?**
**MEASurement<mg>:REFLevel<rl>:RESult:SIGLow?**

Return the high and low signal level, respectively.

**Suffix:**
<mg> 1 to 16, index of the measurement

<rl> Irrelevant, omit the suffix.

**Return values:**
<SignalValue> Range: -1E+26 to 1E+26
Increment: 0
*RST: 0

**Usage:** Query only
Asynchronous command

## 18.11.2 Gate

GATE<g>:ENABle....1078
GATE<g>:GCOupling....1078
GATE<g>:CURSor....1079
GATE<g>:ZDIagram....1079
GATE<g>:MODE....1080
GATE<g>:ABSolute:STARt....1080
GATE<g>:ABSolute:STOP....1080
GATE<g>:RELative:STARt....1080
GATE<g>:RELative:STOP....1080
GATE<g>:SHOW....1081

**GATE<g>:ENABle** <First>

Enables the gate.

**Suffix:**
<g> 1 to 8, index of the gate

**Parameters:**
<First> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Add" on page 214

**GATE<g>:GCOupling** <CouplingMode>

The gate coupling mode selects how the gate area is defined.

**Suffix:**
<g> 1 to 8, index of the gate

**Parameters:**
<CouplingMode> MANual | CURSor | ZOOM | SPECtrum

**MANual**
Manually define the gate with a user-defined start and stop values.

**CURSor**
Cursor coupling is available if a cursor is defined. The gate area is defined by the cursor lines of an active cursor measurement.

**ZOOM**
Zoom coupling is available if a zoom is defined. The gate area is defined identically to the zoom area - if you change the zoom, the gate changes as well.

**SPECtrum**
Spectrum coupling is available if a spectrum is enabled.

*RST: MANual

**Usage:** Asynchronous command

**Manual operation:** See "Coupling mode" on page 214

---

**GATE<g>:CURSor** <Cursor>

Available for `GATE<g>:GCOupling` = `CURSor`.

Selects the cursor set to be used for measurement gating. The gate area is defined by the cursor lines.

**Suffix:**
<g> 1 to 8, index of the gate

**Parameters:**
<Cursor> CURSor1 | CURSor2 | CURSor3 | CURSor4

**Usage:** Asynchronous command

**Manual operation:** See "Coupling mode" on page 214

---

**GATE<g>:ZDIagram** <SignalSource>,[<SignalSource2>]

Available for `GATE<g>:GCOupling` = `ZOOM`.

The gate area is defined identically to the zoom area for the selected zoom diagram.

**Suffix:**
<g> 1 to 8, index of the gate

**Parameters:**
<SignalSource>

<SignalSource2>

**Usage:** Asynchronous command

**Manual operation:** See "Coupling mode" on page 214

**GATE<g>:MODE** <Mode>

Selects if the gate settings are configured using absolute or relative values.

**Suffix:**
<g> 1 to 8, index of the gate

**Parameters:**
<Mode> ABS | REL
*RST: ABS

**Usage:** Asynchronous command

**Manual operation:** See "Mode, Start, Stop" on page 215

**GATE<g>:ABSolute:STARt** <Start>
**GATE<g>:ABSolute:STOP** <Stop>

Define the absolute start and end values for the gate, respectively.

Available, if `GATE<g>:GCOupling` = `MANUal` and `GATE<g>:MODE` =`ABS`.

**Suffix:**
<g> 1 to 8, index of the gate

**Parameters:**
<Stop>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0.01
Default unit: Depends on the underlying waveform

**Usage:** Asynchronous command

**Manual operation:** See "Mode, Start, Stop" on page 215

**GATE<g>:RELative:STARt** <RelativeStart>
**GATE<g>:RELative:STOP** <RelativeStop>

Define the relative start and end values for the gate, respectively.

Available, if `GATE<g>:GCOupling` = `MANUal` and `GATE<g>:MODE` =`REL`.

**Suffix:**
<g> 1 to 8, index of the gate

**Parameters:**
<RelativeStop>
Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Mode, Start, Stop" on page 215

**GATE<g>:SHOW** <DisplayState>

If enabled, the gate area is indicated in the source diagram.

**Suffix:**
<g> 1 to 8, index of the gate

**Parameters:**
<DisplayState> OFF | ON
*RST: ON

**Usage:** Asynchronous command

### 18.11.3 Timing reference

TREFerence<m>:CDR:SOFTware:ALGorithm....1081
TREFerence<m>:CDR:SOFTware:BWIDth....1082
TREFerence<m>:CDR:SOFTware:CFRequency:CRSYnc....1082
TREFerence<m>:CDR:SOFTware:PLL:DAMPing....1082
TREFerence<m>:CDR:SOFTware:PLL:ORDer....1083
TREFerence<m>:CDR:SOFTware:SREStimate....1083
TREFerence<m>:CDR:SOFTware:PLL:SYNC....1083
TREFerence<m>:CDR:SOFTware:RELBwidth....1084
TREFerence<m>:CDR:SOFTware:SELResults....1084
TREFerence<m>:CLK:FACTor....1084
TREFerence<m>:CLK:FREQuency....1085
TREFerence<m>:CLK:OFFSet....1085
TREFerence<m>:CLK:REFLevel....1085
TREFerence<m>:EDGE....1086
TREFerence<m>:ENABle....1086
TREFerence<m>:GATE....1086
TREFerence<m>:REFLevel....1087
TREFerence<m>:RFLSet....1087
TREFerence<m>:SOURce....1087
TREFerence<m>:STATe?....1087
TREFerence<m>:SYMRate....1088
TREFerence<m>:TYPE....1088
TRFS:COUNt?....1088

**TREFerence<m>:CDR:SOFTware:ALGorithm** <Algorithm>

Sets the software algorithm that is used for software clock data recovery.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<Algorithm> CFRequency | PLL | FF | PLLRlock
CFRequency: constant frequency
PLL: phase-locked loop control system
FF: feed forward

PLLRlock: PLL which is locked at the acquisition start.

*RST: PLLRlock

**Usage:** Asynchronous command

**Manual operation:** See "Algorithm" on page 219

---

**TREFerence<m>:CDR:SOFTware:BWIDth** <Bandwidth>

Sets the PLL bandwidth. It defines the part of the spectrum that the PLL can follow during synchronization.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<Bandwidth>
Range: 20000 to Device specific: DoubleVal(uso::db::wfa::IDeviceConstDataWFA::DOUBLE_ITEM::MaxDeviceBandwidth)
Increment: 10
*RST: 599880
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Bandwidth" on page 220

---

**TREFerence<m>:CDR:SOFTware:CFRequency:CRSYnc** <ClockResync>

Enables continuous synchronization of the clock with the data signal.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<ClockResync> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Clock resync" on page 219

---

**TREFerence<m>:CDR:SOFTware:PLL:DAMPing** <Damping>

Sets the damping factor, which is only relevant for second order PLL.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<Damping>
Range: 0.5 to 2
Increment: 0.01
*RST: 0.7

**Usage:** Asynchronous command

**Manual operation:** See "Damping" on page 220

---

**TREFerence<m>:CDR:SOFTware:PLL:ORDer** <PLLOrder>

Sets the order of the PLL: first or second order. PLL of higher order can compensate for more complex jitter behavior.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<PLLOrder> FIRSt | SECond
*RST: SECond

**Usage:** Asynchronous command

**Manual operation:** See "Order" on page 220

---

**TREFerence<m>:CDR:SOFTware:SREStimate**

Starts a software algorithm to detect the symbol rate for the selected source automatically.

**Suffix:**
<m> 1...4, index of the timing reference

**Usage:** Event
Asynchronous command

**Manual operation:** See "Estimate symbol rate" on page 219

---

**TREFerence<m>:CDR:SOFTware:PLL:SYNC** <InitialPhase>

Defines the phase reference for the first clock edge.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<InitialPhase> SAMPle | DATaedge
SAMPle: the first clock edge matches the first sample of the waveform at the left border of the display.
DATaedge: the first clock edge matches the first edge of the data signal.
*RST: DATaedge

**Usage:** Asynchronous command

**Manual operation:** See "Initial phase sync" on page 220

**TREFerence<m>:CDR:SOFTware:RELBwidth** <RelBw>

Sets the relative bandwidth, that is the ratio of the nominal bit rate to the PLL bandwidth.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<RelBw> Range: 1 to 10000
Increment: 1
*RST: 1667

**Usage:** Asynchronous command

**Manual operation:** See "Rel. bandwidth" on page 220

**TREFerence<m>:CDR:SOFTware:SELResults** <Results>

Selects when the CDR algorithm returns clock edges.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<Results> ALL | LOCKed | AISYnc
ALL: all clock edges are used.
AISYnc = LOCKed: the clock edges of the synchronization time are discarded; results are gathered after initial synchronization of the CDR.
*RST: ALL

**Usage:** Asynchronous command

**Manual operation:** See "Selected results" on page 220

**TREFerence<m>:CLK:FACTor** <Multiplier>

Sets a value for the clock multiplier if `TREFerence<m>:TYPE CLOCk` is set. The muliplier is the ratio of an internal clock rate to the externally supplied clock. It defines the number of samples per clock interval.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<Multiplier> Range: 1 to 100
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Clock multiplier" on page 217

**TREFerence<m>:CLK:FREQuency** <Frequency>

Sets the frequency of the clock signal if `TREFerence<m>:TYPE CLOCk` is set.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<Frequency>
Range: 1000 to Device specific: DoubleVal(uso::db::wfa::IDeviceConstDataWFA::DOUBLE_ITEM::MaxDeviceBandwidth)
Increment: 10
*RST: 500000000
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Clock frequency" on page 216

**TREFerence<m>:CLK:OFFSet** <Offset>

Sets the offset between the clock edge and the data edge if `TREFerence<m>:TYPE CLOCk` is set.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<Offset>
Range: -1000 to 1000
Increment: 0.01
*RST: 0
Default unit: UI

**Usage:** Asynchronous command

**Manual operation:** See "Clock offset" on page 217

**TREFerence<m>:CLK:REFLevel** <ReferenceLevel>

Selects the reference level that is used for the timing reference if the indicated timing reference uses an explicit click signal.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<ReferenceLevel> UPPer | MIDDle | LOWer
*RST: MIDDle

**Usage:** Asynchronous command

**Manual operation:** See "Reference level" on page 216

**TREFerence<m>:EDGE** <Edge>

Sets the clock edges that are used for measurements if `TREFerence<m>:TYPE CLOCk` is set for the indicated measurement.

Sets the data edges for clock data recovery if `TREFerence<m>:TYPE SCDR` is set for the indicated measurement.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<Edge> POSitive | NEGative | EITHer

**POSitive**
The positive clock slope can be used, for example, for single data rate (SDR) signals with bit start at the positive clock edge.

**NEGative**
The negative clock slope can be used, for example, for SDR signals with bit start at the negative clock edge.

**EITHer**
Can be used for double data rate (DDR) signals and clock edges. For data edges, it is the most common setting.

*RST: EITHer

**Usage:** Asynchronous command

**Manual operation:** See "Clock edges" on page 217

**TREFerence<m>:ENABle** <TimeRef>

Adds the indicated timing reference (ON) or removes it (OFF).

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<TimeRef> OFF | ON

**Usage:** Asynchronous command

**TREFerence<m>:GATE** <Gate>

Sets the gate for the timing reference. Enable and configure a gate before you assign it (`GATE<g>:ENABle` =`ON`).

The query returns 0, if no gate is assigned.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<Gate> Number of the gate to be used

**Usage:** Asynchronous command

**TREFerence<m>:REFLevel** <ReferenceLevel>

Selects the reference level that is used for the timing reference if the indicated timing reference uses an explicit click signal.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<ReferenceLevel> UPPer | MIDDle | LOWer
*RST: MIDDle

**Usage:** Asynchronous command

**TREFerence<m>:RFLSet** <RefLevelSet>

Selects the set of reference levels that is used for the timing reference measurements.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<RefLevelSet> 1 to 8, index of the reference level set
Number of the reference level set. Define the reference level set before you use it.

**Usage:** Asynchronous command

**Manual operation:** See "Reference levels" on page 215

**TREFerence<m>:SOURce** <Source>

Selects the clock source if `TREFerence<m>:TYPE CLOCk` is set.

Sets the data source for clock data recovery if `TREFerence<m>:TYPE SCDR` is set for the indicated measurement.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<Source> C1 | C2 | ... | C8 | D0 | D1 | D2 | ... | D14 | D15 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 216

**TREFerence<m>:STATe?**

Returns if the specified timing reference is used by an application, e.g. by jitter measurements.

**Suffix:**
<m> 1...4, index of the timing reference

**Return values:**
<State> OFF | ON
*RST: OFF

**Usage:** Query only
Asynchronous command

---

**TREFerence<m>:SYMRate** <SymbolRate>

Sets the symbol rate of the data signal for the indicated timing reference.

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<SymbolRate> Range: 100000 to 5000000000
Increment: 10
*RST: 1000000000
Default unit: Bd

**Usage:** Asynchronous command

**Manual operation:** See "Symbol rate" on page 217

---

**TREFerence<m>:TYPE** <Type>

Defines the origin of the clock signal - either an existing clock signal or waveform, or a clock generated by clock data recovery (CDR).

**Suffix:**
<m> 1...4, index of the timing reference

**Parameters:**
<Type> CLOCk | SCDR
CLOCk: explicit clock signal
SCDR: software CDR
*RST: SCDR

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 215

---

**TRFS:COUNt?**

Returns the number of defined timing references.

**Return values:**
<Count> Range: 1 to 4

**Usage:** Query only
Asynchronous command

## 18.12 Waveform analysis

Some of the commands in the following section are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, section "Command sequence and synchronization"

- Zoom....1089
- Mathematics....1098
- History....1102
- Reference waveforms....1105
- Waveform histograms....1114

### 18.12.1 Zoom

LAYout<ly>:ZOOM<zo>[:ENABle]....1090
LAYout<ly>:ZOOM<zo>:COUNt?....1090
LAYout<ly>:ZOOM<zo>:DIAG?....1090
LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:POSition....1090
LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:SPAN....1091
LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:STARt....1091
LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:STOP....1091
LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:WIDTh....1092
LAYout<ly>:ZOOM<zo>:HORizontal:MODE....1092
LAYout<ly>:ZOOM<zo>:HORizontal:RELative:POSition....1092
LAYout<ly>:ZOOM<zo>:HORizontal:RELative:SPAN....1093
LAYout<ly>:ZOOM<zo>:HORizontal:RELative:STARt....1093
LAYout<ly>:ZOOM<zo>:HORizontal:RELative:STOP....1094
LAYout<ly>:ZOOM<zo>:HORizontal:RELative:WIDTh....1094
LAYout<ly>:ZOOM<zo>:SOURce....1094
LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:POSition....1095
LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:RANGe....1095
LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:SPAN....1095
LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:STARt....1095
LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:STOP....1096
LAYout<ly>:ZOOM<zo>:VERTical:MODE....1096
LAYout<ly>:ZOOM<zo>:VERTical:RELative:POSition....1096
LAYout<ly>:ZOOM<zo>:VERTical:RELative:STARt....1097
LAYout<ly>:ZOOM<zo>:VERTical:RELative:STOP....1097
LAYout<ly>:ZOOM<zo>:VERTical:RELative:WIDTh....1097
LAYout<ly>:ZOOM<zo>:VERTical:RELative:SPAN....1097
LAYout<ly>:ZOOM<zo>:SSCReen....1098

**LAYout<ly>:ZOOM<zo>[:ENABle]** <State>

Enables the zoom.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<State> OFF | ON

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 223

**LAYout<ly>:ZOOM<zo>:COUNt?**

Returns the max number of available zooms.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**LAYout<ly>:ZOOM<zo>:DIAG?**

Returns the index of the diagram that shows the zoomed waveform.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Return values:**
<ZoomDiagramKey> String wiht the index of the zoom diagram, e.g. "9".

**Usage:** Query only
Asynchronous command

**LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:POSition** <Center>

Defines the x-value of the centerpoint of the zoom area in absolute values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**

<Center>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0
Default unit: Depends on the underlying waveform

**Usage:** Asynchronous command

**Manual operation:** See "Position range" on page 224

---

**LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:SPAN** <Span>

Defines the width of the zoom area in absolute values.

**Suffix:**

<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**

<Span>
Range: 0 to 1E+26
Increment: 0.01
*RST: 0.02
Default unit: Depends on the underlying waveform

**Usage:** Asynchronous command

**Manual operation:** See "Position range" on page 224

---

**LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:STARt** <Start>

Defines the lower limit of the zoom area on the x-axis in absolute values.

**Suffix:**

<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**

<Start>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: -0.01
Default unit: Depends on the underlying waveform

**Usage:** Asynchronous command

**Manual operation:** See "Start Stop" on page 224

---

**LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:STOP** <Stop>

Defines the upper limit of the zoom area on the x-axis in absolute values.

**Suffix:**

<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**

<Stop> Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0.01
Default unit: Depends on the underlying waveform

**Usage:** Asynchronous command

**Manual operation:** See "Start Stop" on page 224

---

**LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:WIDTh** <Span>

Defines the width of the zoom area in absolute values.

**Suffix:**

<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**

<Span> Range: 0 to 1E+26
Increment: 0.01
*RST: 0.02
Default unit: Depends on the underlying waveform

**Usage:** Asynchronous command

---

**LAYout<ly>:ZOOM<zo>:HORizontal:MODE** <Mode>

Defines if absolute or relative values are used to specify the x-axis values. Since the zoom area refers to the active signal, relative values ensure that the zoom area remains the same.

**Suffix:**

<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**

<Mode> ABS | REL
*RST: ABS

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 225

---

**LAYout<ly>:ZOOM<zo>:HORizontal:RELative:POSition** <RelativeCenter>

Defines the x-value of the centerpoint of the zoom area in relative values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<RelativeCenter> Relative position of the centerpoint (x-value)

Range: 0 to 100
Increment: 0.1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Position range" on page 224

---

**LAYout<ly>:ZOOM<zo>:HORizontal:RELative:SPAN** <RelativeSpan>

Defines the width of the zoom area in relative values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<RelativeSpan>

Range: 1E-15 to 100
Increment: 0.1
*RST: 1
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Position range" on page 224

---

**LAYout<ly>:ZOOM<zo>:HORizontal:RELative:STARt** <RelativeStart>

Defines the lower limit of the zoom area on the x-axis in relative values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<RelativeStart>

Range: 0 to 100
Increment: 0.1
*RST: 0
Default unit: %

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Asynchronous command

**Manual operation:** See "Start Stop" on page 224

**LAYout<ly>:ZOOM<zo>:HORizontal:RELative:STOP** <RelativeStop>

Defines the upper limit of the zoom area on the x-axis in relative values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<RelativeStop>
Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Asynchronous command

**Manual operation:** See "Start Stop" on page 224

**LAYout<ly>:ZOOM<zo>:HORizontal:RELative:WIDTh** <RelativeSpan>

Defines the width of the zoom area in relative values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<RelativeSpan>
Range: 1E-15 to 100
Increment: 0.1
*RST: 1
Default unit: %

**Usage:** Asynchronous command

**LAYout<ly>:ZOOM<zo>:SOURce** <Diagrams>

Indicates which of the waveform diagrams is selected for zooming. The number is displayed on the screen in the middle of each diagram.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<Diagrams> Index of the diagram that is zoomed.

**Example:** See Section 18.5.1, "SmartGrid layout with zoom", on page 941.

**Usage:** Asynchronous command

**Manual operation:** See "Diagram" on page 223

**LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:POSition** <Center>

Defines the y-value of the centerpoint of the zoom area in absolute values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<Center>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0.01

**Usage:** Asynchronous command

**Manual operation:** See "Position range" on page 224

**LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:RANGe** <Span>
**LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:SPAN** <Span>

Defines the height of the zoom area in absolute values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<Span>
Range: 0 to 1E+26
Increment: 0.01
*RST: 0.01

**Usage:** Asynchronous command

**Manual operation:** See "Position range" on page 224

**LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:STARt** <Start>

Defines the lower limit of the zoom area on the y-axis in absolute values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<Start>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0.01

**Usage:** Asynchronous command

**Manual operation:** See "Start Stop" on page 224

**LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:STOP** <Stop>

Defines the upper limit of the zoom area on the y-axis in absolute values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<Stop> Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0.01

**Usage:** Asynchronous command

**Manual operation:** See "Start Stop" on page 224

**LAYout<ly>:ZOOM<zo>:VERTical:MODE** <Mode>

Defines if absolute or relative values are used to specify the y-axis values. Since the zoom area refers to the active signal, relative values ensure that the zoom area remains the same.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<Mode> ABS | REL
*RST: REL

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 225

**LAYout<ly>:ZOOM<zo>:VERTical:RELative:POSition** <RelativeCenter>

Defines the y-value of the centerpoint of the zoom area in relative values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<RelativeCenter> Range: 0 to 100
Increment: 0.1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Position range" on page 224

**LAYout<ly>:ZOOM<zo>:VERTical:RELative:STARt** <RelativeStart>

Defines the lower limit of the zoom area on the y-axis in relative values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<RelativeStart>
Range: 0 to 100
Increment: 0.1
*RST: 0
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Start Stop" on page 224

**LAYout<ly>:ZOOM<zo>:VERTical:RELative:STOP** <RelativeStop>

Defines the upper limit of the zoom area on the x-axis, in relative values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<RelativeStop>
Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Start Stop" on page 224

**LAYout<ly>:ZOOM<zo>:VERTical:RELative:WIDTh** <RelativeSpan>
**LAYout<ly>:ZOOM<zo>:VERTical:RELative:SPAN** <RelativeSpan>

Defines the height of the zoom area in relative values.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Parameters:**
<RelativeSpan>
Range: 1E-15 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Position range" on page 224

**LAYout<ly>:ZOOM<zo>:SSCReen**

Sets the zoom area to the whole screen.

**Suffix:**
<ly> 1 to 4, index of the SmartGrid layout

<zo> 1 to 4, index of the zoom

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Set to screen" on page 224

## 18.12.2 Mathematics

To transfer the math waveform data to a remote computer, use:

- `CALCulate:MATH<m>:DATA:HEADer?` on page 1144
- `CALCulate:MATH<m>:DATA[:VALues]?` on page 1145
- `CALCulate:MATH<m>:DATA:STYPe?` on page 1145

For fast export of several waveforms at once, use `EXPort:WAVeform:DATA[:VALues]?` on page 1150.

For export of waveform data to file, see Section 18.13.4, "Waveform data: export to file", on page 1127.

CALCulate:MATH<m>:STATe ..... 1098
CALCulate:MATH<m>[:EXPRession][:DEFine] ..... 1098
CALCulate:MATH<m>:LABel ..... 1100
CALCulate:MATH<m>:VERTical:SCALe[:VALue] ..... 1100
CALCulate:MATH<m>:UNIT ..... 1100
CALCulate:MATH<m>:VERTical:OFFSet ..... 1101
CALCulate:MATH<m>:ENVSelection ..... 1101
CALCulate:MATH<m>:VERTical:SCALe:MODE ..... 1101

**CALCulate:MATH<m>:STATe** <First>

Activates the selected Math channel and displays the defined math waveforms.

**Suffix:**
<m> 1 to 8, index of the math waveform

**Parameters:**
<First> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 228

**CALCulate:MATH<m>[:EXPRession][:DEFine]** <Expression>

Defines the math expression to be calculated for the specified math channel.

| Operation | <Expression> | Comment |
|---|---|---|
| Addition | "C1+C2" | |
| Subtraction | "C1-C2" | |
| Multiplication | "C1*C2" | |
| Division | "C1/C2" | 0/0 = 0<br>+1 / 0 = Clip+<br>-1 / 0 = Clip- |
| Inverting | "-C1" | |
| Absolute value | "Abs(C1)" | |
| Derivation | "Derivation(C1,NoiseReject)" | NoiseReject can get any value between 1 and 5000 points<br>Default = 50 |
| Integral | "Integral(C1)" | |
| Logarithm (based on 10) | "Log(C1)" | Uses the absolute value of the source in calculation.<br>Log(0) = Clip - |
| Natural logarithm (based on e) | "Ln(C1)" | Uses the absolute value of the source in calculation.<br>Log(0) = Clip - |
| Binary logarithm (based on 2) | "Ld(C1)" | Uses the absolute value of the source in calculation.<br>Log(0) = Clip - |
| Square | "Pow(C1)" | |
| Square root | "Sqrt(C1)" | Uses the absolute value of the source in calculation. |
| Rescale | "Rescale(C1,a,b)" | a = scale, default = 1<br>b = offset, default = 0 |
| FIR | "FIR(Type,C1,Cut-Off,Characteristics)"<br><br>Examples:<br>"FIR(highpass,C1,10000000,Gaussian)"<br>"FIR(lowpass,C1,10000000,rectangle)" | Type = lowpass, highpass<br>Cut-Off = limit frequency<br>Characteristics = Gaussian, rectangle<br>Cut-Off can get any value between 4 GHz and 1 kHz |

**Suffix:**
<m> 1 to 8, index of the math waveform

**Parameters:**
<Expression> String with regular expression for calculation

**Example:** `CALC:MATH 'C1*C2'`
Defines the multiplication of waveforms channel 1 and channel 2.

**Usage:** Asynchronous command

**Manual operation:** See "Operator" on page 229

---

**CALCulate:MATH<m>:LABel** <Label>

Defines a label for the selected math waveform.

**Suffix:**
<m> 1 to 8, index of the math waveform

**Parameters:**
<Label> String with the label

**Example:**
```
CALCulate:MATH3:LABel 'M3AddC1C2'
```

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 228

---

**CALCulate:MATH<m>:VERTical:SCALe[:VALue]** <VerticalScale>

Sets the scale of the y-axis in the math function diagram. The value is defined as "<unit> per division", e.g. *50 mV/div*. In this case, the horizontal grid lines are displayed in intervals of 50 mV.

**Suffix:**
<m> 1 to 8, index of the math waveform

**Parameters:**
<VerticalScale>
Range: 1E-12 to 100E+12
Increment: 1E-05
*RST: 0.5
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "[Scale]" on page 49

---

**CALCulate:MATH<m>:UNIT** <UserUnit>

Sets a user-defined unit for the math operation.

**Suffix:**
<m> 1 to 8, index of the math waveform

**Parameters:**
<UserUnit> String with the base unit

**Example:**
```
CALCulate:MATH:UNIT 'A'
```

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 238

**CALCulate:MATH<m>:VERTical:OFFSet** <VerticalOffset>

Sets a voltage offset to adjust the vertical position of the math function on the screen. Negative values move the waveform up, positive values move it down.

**Suffix:**
<m> 1 to 8, index of the math waveform

**Parameters:**
<VerticalOffset> Range: -100E12 to 100E12
Increment: 0.01
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Vertical offset" on page 238

**CALCulate:MATH<m>:ENVSelection** <EnvelopeCurve>

Selects the upper or lower part of the input waveform for mathematic calculation, or a combination of both.

**Suffix:**
<m> 1 to 8, index of the math waveform

**Parameters:**
<EnvelopeCurve> MIN | MAX | BOTH
*RST: BOTH

**Usage:** Asynchronous command

**Manual operation:** See "Envelope wfm selection" on page 230

**CALCulate:MATH<m>:VERTical:SCALe:MODE** <VertSclMd>

Sets how the vertical scale is adapted to the current measurement results. By default, scaling is done automatically to provide an optimal display. However, if necessary, you can define scaling values manually to suit your requirements.

**Suffix:**
<m> 1 to 8, index of the math waveform

**Parameters:**
<VertSclMd> MANual | AUTO
*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Scale mode" on page 238

### 18.12.3 History

ACQuire:HISTory:CURRent....1102
ACQuire:HISTory:ISODate?....1102
ACQuire:HISTory:PLAY....1102
ACQuire:HISTory:REPLay....1103
ACQuire:HISTory:STARt....1103
ACQuire:HISTory:STOP....1103
ACQuire:HISTory:TPACq....1103
ACQuire:HISTory:TSABsolute?....1104
ACQuire:HISTory:TSDate?....1104
ACQuire:HISTory:TSRelative?....1104
ACQuire:HISTory:TSRReference?....1105
ACQuire:HISTory[:STATe]....1105

**ACQuire:HISTory:CURRent** <CurrentAcqIndex>

Accesses a particular acquisition in the memory to display it, or to save it. The newest acquisition always has the index "0". Older acquisitions have a negative index.

**Parameters:**

<CurrentAcqIndex>
Range: -2147483648 to 0
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Current acq" on page 242

**ACQuire:HISTory:ISODate?**

Returns the absolute date and time of the acquisition that is selected in history view in ISO 8601 format. The same format is used in the header of exported waveform data files.

See also: `ACQuire:HISTory:CURRent`).

**Return values:**

<DateTimeISO8601> String with absolute date and time in this order: year, month, day, hour, minutes, seconds, and milliseconds.

**Example:**

```
ACQ:HIST:ISOD?
<-- "2024-10-02T10:11:09.603208067"
```

**Usage:** Query only
Asynchronous command

**ACQuire:HISTory:PLAY**

Starts and stops the replay of the history waveforms.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Player" on page 242

---

**ACQuire:HISTory:REPLay** <AutoRepeat>

If enabled, the replay of the history waveform sequence repeats automatically.

Otherwise, the replay stops at the stop index set with `ACQuire:HISTory:STOP` on page 1103.

**Parameters:**

<AutoRepeat> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Auto repeat" on page 244

---

**ACQuire:HISTory:STARt** <StartAcqIndex>

Sets the index of the first (oldest) acquisition to be displayed or exported. The index is always negative.

**Parameters:**

<StartAcqIndex> Range: -2147483648 to 0
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Start acq" on page 242

---

**ACQuire:HISTory:STOP** <StopAcqIndex>

Sets the index of the last (newest) acquisition to be displayed or exported. The newest acquisition of the complete acquisition series always has the index "0".

**Parameters:**

<StopAcqIndex> Range: -2147483648 to 0
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Stop acq" on page 242

---

**ACQuire:HISTory:TPACq** <TimePerAcq>

Sets the display time for one acquisition. The shorter the time, the faster the replay is.

**Parameters:**

<TimePerAcq>
Range: 4E-05 to 10
Increment: 1
*RST: 0.05
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Display time" on page 242

---

**ACQuire:HISTory:TSABsolute?**

Returns the absolute daytime of the selected acquisition (`ACQuire:HISTory:CURRent`).

For automatic parsing of the time, use `ACQuire:HISTory:ISODate?`.

**Return values:**

<TimeAbsString> String containing the time and unit

**Example:**

```
ACQ:HIST:TSAB?
<-- "10:11 9,603.208.067 s"
```

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Time stamp" on page 243

---

**ACQuire:HISTory:TSDate?**

Returns the date of the selected acquisition (`ACQuire:HISTory:CURRent`).

For automatic parsing of the time, use `ACQuire:HISTory:ISODate?`.

**Return values:**

<DateAbsString> String parameter with acquisition date

**Example:**

```
ACQ:HIST:TSD?
<-- "2024:10:02"    //02 Oct 2024
```

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Time stamp" on page 243

---

**ACQuire:HISTory:TSRelative?**

Returns the relative time of the current acquisition - the time difference to the newest acquisition (index = 0).

See also: `ACQuire:HISTory:CURRent`.

**Return values:**

<TimeRelativ> Range: -1E+26 to 1E+26
Increment: 1
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Time stamp" on page 243

---

**ACQuire:HISTory:TSRReference?**

Returns the relative time of the currently selected acquisition and the internal reference time (horizontal alignment) in history view in relation to the acquisition with index 0.

**Return values:**

<TimeRelIntRef> Range: -1E+26 to 1E+26
Increment: 1
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Time stamp" on page 243

---

**ACQuire:HISTory[:STATe]** <State>

Enables the history mode and allows you to save history waveforms to file.

**Parameters:**

<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show history" on page 241

## 18.12.4 Reference waveforms


- Setup....1105
- Scaling....1109
- Reference waveform data export....1113


### 18.12.4.1 Setup


REFCurve<rc>:ABORt....1106
REFCurve<rc>:CLEar....1106
REFCurve<rc>:NAME....1106
REFCurve<rc>:OFFSet....1107

REFCurve<rc>:OPEN....1107
REFCurve<rc>:RESTore....1107
REFCurve<rc>:SAVE....1107
REFCurve<rc>:SOURce....1108
REFCurve<rc>:STATe....1109
REFCurve<rc>:UPDate....1109

**REFCurve<rc>:ABORt**

Aborts a running reference waveform export, which was started with `REFCurve<rc>:SAVE`, or a running reference waveform update, which was started with `REFCurve<rc>:UPDate`.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save as" on page 249

**REFCurve<rc>:CLEar**

Deletes the selected reference waveform. It disappears from the display, and its memory is deleted.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Clear" on page 249

**REFCurve<rc>:NAME** <Name>

Defines the path, filename and extension of the reference waveform file to be loaded, saved or deleted.

For supported file types, see "Supported file formats" on page 245.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<Name> String with path, filename and extension

**Usage:** Asynchronous command

**Manual operation:** See "Save as" on page 249

**REFCurve<rc>:OFFSet** <VerticalOffset>

The vertical offset moves the reference waveform vertically. Enter a value with the unit of the waveform.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<VerticalOffset> Range: -1E+26 to 1E+26
Increment: 1E-06
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Vertical offset" on page 252

**REFCurve<rc>:OPEN**

Loads the reference waveform file selected by `REFCurve<rc>:NAME`.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Usage:** Event
Asynchronous command

**Manual operation:** See "Open" on page 249

**REFCurve<rc>:RESTore**

Applies the original settings of the reference waveform to the horizontal and vertical settings of the selected waveform.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Restore settings" on page 251

**REFCurve<rc>:SAVE**

Saves the reference waveform to the file selected by `REFCurve<rc>:NAME`.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save as" on page 249

**REFCurve<rc>:SOURce** <Source>

Selects the source waveform from the active waveforms, e.g. input channels, math waveforms, or spectrum.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<Source> C1 | C2 | ... | C8 | M1 | M2 | ... | M8 | R1 | R2 | ... | R8 | SPECMAXH1 | SPECMINH1 | SPECNORM1 | SPECAVER1 | SPECMAXH2 | SPECMINH2 | SPECNORM2 | SPECAVER2 | SPECMAXH3 | SPECMINH3 | SPECNORM3 | SPECAVER3 | SPECMAXH4 | SPECMINH4 | SPECNORM4 | SPECAVER4 | TRK1 | TRK2 | TRK3 | ... | TRK16 | PA1QPOWER | PA2QPOWER | PA3QPOWER | PA1HPOWER1 | PA2HPOWER1 | PA3HPOWER1| | PA1SPOWER | PA2SPOWER | PA3SPOWER | PA1SOA | PA2SOA | PA3SOA | PA1IPOWER | PA2IPOWER | PA3IPOWER | PA1OPOWER1 | PA2OPOWER1 | PA3OPOWER1 | PA1OPOWER2 | PA2OPOWER2 | PA3OPOWER2 | PA1OPOWER3 | PA2OPOWER3 | PA3OPOWER3 | PTHREE1QPOWER1 | PTHREE2QPOWER1 | PTHREE3QPOWER1 | PTHREE1QPOWER2 | PTHREE2QPOWER2 | PTHREE3QPOWER2 | PTHREE1QPOWER3 | PTHREE2QPOWER3 | PTHREE3QPOWER3 | PTHREE1QTPOWER | PTHREE2QTPOWER | PTHREE3QTPOWER | PTHREE1HPOWER1 | PTHREE2HPOWER1 | PTHREE3HPOWER1 | PTHREE1HPOWER2 | PTHREE2HPOWER2 | PTHREE3HPOWER2 | PTHREE1HPOWER3 | PTHREE2HPOWER3 | PTHREE3HPOWER3 | PTHREE1HTPOWER | PTHREE2HTPOWER | PTHREE3HTPOWER | PTHREE1EIP1| PTHREE2EIP1| PTHREE3EIP1| PTHREE4EIP1| PTHREE5EIP1| PTHREE6EIP1| PTHREE1EIP2| PTHREE2EIP2| PTHREE3EIP2| PTHREE4EIP2| PTHREE5EIP2| PTHREE6EIP2| PTHREE1EIP3| PTHREE2EIP3| PTHREE3EIP3| PTHREE4EIP3| PTHREE5EIP3| PTHREE6EIP3| PTHREE1EITP| PTHREE2EITP| PTHREE3EITP| PTHREE4EITP| PTHREE5EITP| PTHREE6EITP | PTHREE1EOP1| PTHREE2EOP1| PTHREE3EOP1| PTHREE4EOP1| PTHREE5EOP1| PTHREE6EOP1| PTHREE1EOP2| PTHREE2EOP2| PTHREE3EOP2| PTHREE4EOP2| PTHREE5EOP2| PTHREE6EOP2| PTHREE1EOP3| PTHREE2EOP3| PTHREE3EOP3| PTHREE4EOP3| PTHREE5EOP3| PTHREE6EOP3| PTHREE1EOTP| PTHREE2EOTP| PTHREE3EOTP| PTHREE4EOTP| PTHREE5EOTP| PTHREE6EOTP | PA1TOPOWER | PA2TOPOWER | PA3TOPOWER

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 248

**REFCurve<rc>:STATe** <State>

Enables the display of the reference waveform in the diagram. Before you can display it, create the reference waveform.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show" on page 248

**REFCurve<rc>:UPDate**

Copies the selected source waveform with all its settings to the memory of the reference waveform. If there is a previously defined reference waveform in this memory, it is updated by the current source waveform. If the acquisition is running, the reference waveform is a snapshot.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Usage:** Event
Asynchronous command

**Manual operation:** See "Create/Update" on page 249

### 18.12.4.2 Scaling

REFCurve<rc>:AXIS[:XMODe] ... 1109
REFCurve<rc>:SCALe ... 1110
REFCurve<rc>:POSition ... 1110
REFCurve<rc>:HMODe ... 1110
REFCurve<rc>:RESCale:HORizontal:FACTor ... 1111
REFCurve<rc>:RESCale:HORizontal:OFFSet ... 1111
REFCurve<rc>:RESCale:HORizontal:STATe ... 1111
REFCurve<rc>:RESCale:VERTical:FACTor ... 1112
REFCurve<rc>:RESCale:VERTical:OFFSet ... 1112
REFCurve<rc>:RESCale:VERTical:STATe ... 1112
REFCurve<rc>:TOORiginal ... 1113
REFCurve<rc>:VMODe ... 1113

**REFCurve<rc>:AXIS[:XMODe]** <XAxisMode>

Defines the scaling method for the frequency (x-axis) of the reference curve.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<XAxisMode> LIN | LOG
`LIN`: Linear scaling
`LOG`: Logarithmic scaling
*RST: LIN

**Usage:** Asynchronous command

---

**REFCurve<rc>:SCALe** <VerticalScale>

Sets the vertical scale, which defines the displayed amplitude of the selected waveform.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<VerticalScale> Range: 1E-15 to 1E+26
Increment: 1E-05
*RST: 0.5
Default unit: Unit of the source waveform

**Usage:** Asynchronous command

**Manual operation:** See "[Scale]" on page 49

---

**REFCurve<rc>:POSition** <VertPosi>

Available, if `REFCurve<rc>:VMODe` is set to `INDependent`.

Moves the reference waveform up or down in the diagram.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<VertPosi> Range: -1E+26 to 1E+26
Increment: 0.02
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Vertical position" on page 252

---

**REFCurve<rc>:HMODe** <HorizontalMode>

Selects the coupling of horizontal settings.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<HorizontalMode> ORIGinal | COUPled

**ORIGinal**
Horizontal scaling and reference point of the source waveform are used.

**COUPled**
The current horizontal settings of the diagram are used.

*RST: ORIGinal

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 253

---

**REFCurve<rc>:RESCale:HORizontal:FACTor** <ScaleFactor>

Sets the horizontal scale factor. A factor greater than 1 stretches the waveform horizontally, a factor lower than 1 compresses the curve.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<ScaleFactor>
Range: 1E-14 to 100000000000000
Increment: 0.1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Scale factor" on page 253

---

**REFCurve<rc>:RESCale:HORizontal:OFFSet** <Offset>

Moves the waveform horizontally. Enter a value with a time unit suitable for the time scale of the diagram.

Positive values shift the waveform to the right, negative values shift it to the left.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<Offset>
Range: -5000000 to 5000000
Increment: 0.01
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Horizontal offset" on page 254

---

**REFCurve<rc>:RESCale:HORizontal:STATe** <State>

If enabled, the horizontal offset and factor are applied to the reference waveform.

Stretching and offset change the display of the waveform independent of the horizontal settings of the source waveform and of the horizontal diagram settings.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 253

---

**REFCurve<rc>:RESCale:VERTical:FACTor** <ScaleFactor>

Sets the vertical scale factor. A factor greater than 1 stretches the waveform vertically, a factor lower than 1 compresses the curve.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<ScaleFactor> Range: -100000000 to 100000000
Increment: 0.1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Scale factor" on page 253

---

**REFCurve<rc>:RESCale:VERTical:OFFSet** <Offset>

The vertical offset moves the reference waveform vertically. Enter a value with the unit of the waveform.

Like vertical offset of a channel waveform, the offset of a reference waveform is subtracted from the measured value. Negative values shift the waveform up, positive values shift it down.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<Offset> Range: -100000000 to 100000000
Increment: 1E-06
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Vertical offset" on page 253

---

**REFCurve<rc>:RESCale:VERTical:STATe** <State>

If enabled, the vertical offset and factor are applied to the reference waveform.

Stretching and offset change the display of the waveform independent of the vertical scale and position.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 253

---

**REFCurve<rc>:TOORiginal**

Available, if `REFCurve<rc>:VMODe` is set to `INDependent`.

Restores the original vertical settings of the reference waveform.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Set to original" on page 251

---

**REFCurve<rc>:VMODe** <VerticalMode>

Selects the coupling of vertical settings.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Parameters:**
<VerticalMode> COUPled | INDependent

**COUPled**
Vertical position and scale of the source are used.

**INDependent**
Scaling and position can be set specific to the reference waveform.

*RST: INDependent

**Usage:** Asynchronous command

**Manual operation:** See "Vertical mode" on page 251

### 18.12.4.3 Reference waveform data export

Commands to transmit the data of reference waveforms are the following:

- `REFCurve<rc>:DATA:HEADer?` on page 1143
- `REFCurve<rc>:DATA[:VALues]?` on page 1144

- REFCurve<rc>:DATA:STYPe? on page 1144

Commands for saving waveform data to a file are described in Section 18.13.4, "Waveform data: export to file", on page 1127.

## 18.12.5 Waveform histograms

- Histogram settings ..... 1114
- Window settings ..... 1116
- Histogram data transfer ..... 1118

### 18.12.5.1 Histogram settings

**HISTogram<m>:ADD**

Creates a new histogram with the specified index.

**Suffix:**
<m> 1...8, index of the histogram

**Usage:** Setting only
Asynchronous command

**HISTogram<m>:COUNt?** <Count>

Returns the number of created histograms.

You can query the minimum and maximum values with `<command>? MIN` and `<command>? MAX`.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Return values:**
<Count> Counted number of histograms

**Usage:** Query only
Asynchronous command

**HISTogram<m>:ENABle** <State>

Enables or disables the histogram evaluation and display. The histogram settings are kept until the histogram is deleted.

**Suffix:**
<m> 1...8, index of the histogram

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 256

---

**HISTogram<m>:MODE** <Mode>

Defines the type of histogram.

**Suffix:**
<m> 1...8, index of the histogram

**Parameters:**
<Mode> VERTical | HORizontal

**VERTical**
A vertical histogram has horizontal bars that show the occurence of amplitude, or vertical values.

**HORizontal**
A horizontal histogram has vertical bars that show the occurrence of a sample at a given time on the x-axis.

*RST: VERTical

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 256

---

**HISTogram<m>:REMove**

Deletes the specified histogram.

**Suffix:**
<m> 1...8, index of the histogram

**Usage:** Setting only
Asynchronous command

---

**HISTogram<m>:RESet**

Restarts the calculation of the selected histogram.

To reset all histograms and measurements, use `DISPlay:CLR`.

**Suffix:**
<m> 1...8, index of the histogram

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Reset" on page 256

---

**HISTogram<m>:SOURce** <Source>

Defines the source of the histogram.

**Suffix:**
<m> 1...8, index of the histogram

**Parameters:**
<Source> C1 | C2 | ... | C8 | M1 | M2 | ... | M8 | R1 | R2 | ... | R8

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 256

### 18.12.5.2 Window settings

HISTogram<m>:WINDow:ENABle ... 1116
HISTogram<m>:WINDow:HORizontal:MODE ... 1116
HISTogram<m>:WINDow:VERTical:MODE ... 1116
HISTogram<m>:WINDow:HORizontal:ABSolute:POSition ... 1117
HISTogram<m>:WINDow:HORizontal:ABSolute:SPAN ... 1117
HISTogram<m>:WINDow:HORizontal:ABSolute:STARt ... 1117
HISTogram<m>:WINDow:HORizontal:ABSolute:STOP ... 1117
HISTogram<m>:WINDow:HORizontal:RELative:POSition ... 1117
HISTogram<m>:WINDow:HORizontal:RELative:SPAN ... 1117
HISTogram<m>:WINDow:HORizontal:RELative:STARt ... 1117
HISTogram<m>:WINDow:HORizontal:RELative:STOP ... 1117
HISTogram<m>:WINDow:VERTical:ABSolute:POSition ... 1118
HISTogram<m>:WINDow:VERTical:ABSolute:SPAN ... 1118
HISTogram<m>:WINDow:VERTical:ABSolute:STARt ... 1118
HISTogram<m>:WINDow:VERTical:ABSolute:STOP ... 1118
HISTogram<m>:WINDow:VERTical:RELative:POSition ... 1118
HISTogram<m>:WINDow:VERTical:RELative:SPAN ... 1118
HISTogram<m>:WINDow:VERTical:RELative:STARt ... 1118
HISTogram<m>:WINDow:VERTical:RELative:STOP ... 1118

---

**HISTogram<m>:WINDow:ENABle** <State>

When you use a histogram window, the analyzed part of the source waveform is limited vertically and horizontally.

**Suffix:**
<m> 1...8, index of the histogram

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Use window" on page 257

---

**HISTogram<m>:WINDow:HORizontal:MODE** <Mode>
**HISTogram<m>:WINDow:VERTical:MODE** <Mode>

The commands define whether the window limits are entered as absolute or relative values, in horizontal and vertical direction.

**Suffix:**
<m> 1...8, index of the histogram

**Parameters:**
<Mode> ABS | REL
*RST: ABS

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 257

---

**HISTogram<m>:WINDow:HORizontal:ABSolute:POSition** <Value>
**HISTogram<m>:WINDow:HORizontal:ABSolute:SPAN** <Value>
**HISTogram<m>:WINDow:HORizontal:ABSolute:STARt** <Value>
**HISTogram<m>:WINDow:HORizontal:ABSolute:STOP** <Value>

Set the horizontal window limits as absolute values. Enter the unit of the horizontal axis together with the value.

**Suffix:**
<m> 1...8, index of the histogram

**Parameters:**
<Value> Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0 for position, 0.02 for span, -0.01 for start and 0.01 for stop

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop, Range, Position" on page 257

---

**HISTogram<m>:WINDow:HORizontal:RELative:POSition** <Value>
**HISTogram<m>:WINDow:HORizontal:RELative:SPAN** <Value>
**HISTogram<m>:WINDow:HORizontal:RELative:STARt** <Value>
**HISTogram<m>:WINDow:HORizontal:RELative:STOP** <Value>

Set the horizontal window limits as relative values in % of the diagram width.

**Suffix:**
<m> 1...8, index of the histogram

**Parameters:**
<Value> Range: 0 for position, start and stop, 1E-15 for span to 100
Increment: 0.1
*RST: position: 50, span: 1, start: 0, stop: 100
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Relative start, Relative stop, Relative range, Relative position" on page 258

**HISTogram<m>:WINDow:VERTical:ABSolute:POSition** <Value>
**HISTogram<m>:WINDow:VERTical:ABSolute:SPAN** <Value>
**HISTogram<m>:WINDow:VERTical:ABSolute:STARt** <Value>
**HISTogram<m>:WINDow:VERTical:ABSolute:STOP** <Value>

Set the vertical window limits as absolute values. Enter the unit of the vertical axis together with the value.

**Suffix:**
<m> 1...8, index of the histogram

**Parameters:**
<Value>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0 for position, 0.02 for span, -0.01 for start and 0.01 for stop

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop, Range, Position" on page 258

**HISTogram<m>:WINDow:VERTical:RELative:POSition** <Value>
**HISTogram<m>:WINDow:VERTical:RELative:SPAN** <Value>
**HISTogram<m>:WINDow:VERTical:RELative:STARt** <Value>
**HISTogram<m>:WINDow:VERTical:RELative:STOP** <Value>

Set the vertical window limits as relative values in % of the diagram height.

**Suffix:**
<m> 1...8, index of the histogram

**Parameters:**
<Value>
Range: 0 for position, start and stop, 1E-15 for span to 100
Increment: 0.1
*RST: position: 50, span: 1, start: 0, stop: 100
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop, Range, Position" on page 258

### 18.12.5.3 Histogram data transfer

To transfer the waveform histogram data to a remote computer, use:

- `EXPort:HISTogram<m>:DATA:HEADer?` on page 1149
- `EXPort:HISTogram<m>:DATA[:VALues]?` on page 1150

For export of histogram data to file, see Section 18.13.5, "Histogram data: export to file", on page 1132.

# 18.13 Data management

Some of the commands in the following section are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, section "Command sequence and synchronization"

## 18.13.1 Instrument settings

The Mass MEMory subsystem provides commands to access the storage media and to save and reload instrument settings.

**File and directory names**

The <file_name> and <directory_name> parameters are strings. Some commands use a fixed directory; for others the <file_name> can contain the complete path including the drive name and all subdirectories. If no complete path is specified, the file location is relative to the current directory, queried with `MMEMory:CDIRectory?`. The filename itself may contain the period as a separator for extensions.

File and directory names can be chosen according to Windows™ conventions. All letters and numbers are allowed, as well as the special characters "_", "^", "$", "~", "!", "#", "%", "&", "-", "{", "}", "(", ")", "@" and "`". Reserved filenames are CON, AUX, COM1, ..., COM4, LPT1, ..., LPT3, NUL and PRN.

The use of wildcards ? and * is not allowed.

MMEMory:DRIVes? 1120
MMEMory:DCATalog? 1120
MMEMory:DCATalog:LENGth? 1120
MMEMory:CDIRectory 1121
MMEMory:MDIRectory 1121
MMEMory:RDIRectory 1121
MMEMory:CATalog? 1121
MMEMory:CATalog:LENGth? 1122
MMEMory:COPY 1122
MMEMory:MOVE 1123
MMEMory:DELete 1123
MMEMory:DATA 1123
MMEMory:ATTRibute 1124
MMEMory:SAV 1124
MMEMory:RCL 1125
MMEMory:STORe:STATe 1125

MMEMory:LOAD:STATe ..... 1125
MMEMory:GENerator:RCL ..... 1125
MMEMory:GENerator:SAV ..... 1126

**MMEMory:DRIVes?**

Returns the path list of available drives.

**Return values:**

<Drive> List of strings, for example:
Instrument only: `"/home/storage/userData"`
Instrument with connected USB flash drive:
`"/home/storage/userData","/run/media/usb/<MyDriveName>/MYD`
. MYDATA is the partition name, which is also shown in the file explorer.
Instrument with connected USB flash drive:
`"/home/storage/userData","/run/media/usb/<MyDriveName>/8AF`
. 8AF8-3EBA is an example ID. ID is used if the partition does not have a name, or the name cannot be read.

**Usage:** Query only

**MMEMory:DCATalog?** [<PathName>]

Returns the subdirectories of the current or of a specified directory.

**Query parameters:**

<PathName> String parameter to specify the directory. If the directory is omitted, the command queries the contents of the current directory, to be set and queried with `MMEMory:CDIRectory`.

**Return values:**

<FileEntry> Names of the subdirectories separated by colons. The first two strings are related to the parent directory.

**Example:** `MMEM:DCAT?`

**Usage:** Query only

**MMEMory:DCATalog:LENGth?** [<PathName>]

Returns the number of subdirectories of the current or of a specified directory. The number includes the parent directory strings "." and ".." and corresponds to the number of strings returned by the `MMEMory:DCATalog?` command.

**Query parameters:**

<PathName> String parameter to specify the directory. If the directory is omitted, the command queries the contents of the current directory, to be set and queried with `MMEMory:CDIRectory`.

**Return values:**

<FileEntryCount> Number of parent directories and subdirectories.

**Example:** `MMEM:DCAT:LENG?`
`5`

**Usage:** Query only

---

**MMEMory:CDIRectory** [<DirectoryName>]

Changes the default directory for file access.

**Parameters:**
<DirectoryName> String parameter to specify the directory.
*RST: "\"

**Example:** `MMEM:CDIR '/home/storage/userData/Waveforms'`

**Usage:** SCPI confirmed

---

**MMEMory:MDIRectory** <DirectoryName>

Creates a new directory with the specified name.

**Setting parameters:**
<DirectoryName> String parameter to specify the new directory. If the path consists of several subdirectories, the complete tree is created if necessary.

**Example:** `MMEM:MDIR '/home/storage/userData/Waveforms'`

**Usage:** Setting only

---

**MMEMory:RDIRectory** <DirectoryName>

Deletes the specified directory.

**Setting parameters:**
<DirectoryName> String parameter to specify the directory to be deleted.
This directory must be empty, otherwise it is not deleted.

**Example:** `MMEM:RDIR '/home/storage/userData/Test'`

**Usage:** Setting only

---

**MMEMory:CATalog?** [<PathName>][, <Format>]

Returns a list of files contained in the specified directory. The result corresponds to the number of files returned by the MMEMory:CATalog:LENgth command.

**Query parameters:**
<PathName> String parameter to specify the directory. If the directory is omitted, the command queries directory specified with `MMEMory:CDIRectory`.

<Format> ALL | WTIMe

ALL: Extended result including file, date, time and attributes
WTIMe: Extended result including file, date, time

**Return values:**

<UsedMemory> Total amount of storage currently used in the directory, in bytes.

<FreeMemory> Total amount of storage available in the directory, in bytes.

<FileEntry> All files and subdirectories of the directory are listed with their filename, format and size in bytes. The first two strings are related to the parent directory.

**Example:**

```
MMEM:CAT? '/home/storage/userData'
529479,1831777894400,".,DIR,0",",.,DIR,0",
"Backup,DIR,0","CSS,DIR,0","DATEN,DIR,0",
"Commands.jar,BIN,529479","FAVORITES,DIR,0",
"LOG,DIR,0","DATA,DIR,0","test,DIR,0",
"TotalCMD,DIR,0"
```

**Usage:** Query only
SCPI confirmed

---

**MMEMory:CATalog:LENGth?** [<PathName>]

Returns the number of files and subdirectories of the current or specified directory. The number includes the parent directory strings "." and ".." and it corresponds to the number of <FileEntry> strings returned by the `MMEMory:CATalog?` command.

**Query parameters:**

<PathName> String parameter, directory to be queried. If the directory is omitted, the current directory is queried, specified with `MMEMory:CDIRectory`.

**Return values:**

<Count> Number of files and subdirectories including parent directory entries.

**Example:**

```
MMEM:CDIR '/home/storage/userData'
MMEM:CAT:LENG?
11
```

**Usage:** Query only

---

**MMEMory:COPY** <FileSource>[, <FileDestination>]

Copies an existing file to a new file.

**Setting parameters:**

<FileSource> String parameter. Contains name and path of the file to be copied. Wildcards (* and ?) are allowed.

<FileDestination> String parameter. Contains name and path of the new file. If the file already exists, it is overwritten without notice. If no file destination is specified, the source file is written to the current directory specified with MMEMory:CDIRectory.

**Usage:** Setting only
SCPI confirmed

---

**MMEMory:MOVE** <FileSource>, <FileDestination>

Moves the specified file to a new location on the same drive and renames it.

**Setting parameters:**

<FileSource> String parameter, contains name and path of the file to be copied.
Wildcards (* and ?) are allowed. Therefore, specify a directory for <FileDestination>. Renaming is not possible.

<FileDestination> String parameter. Contains name and path of the new file. If no path is specified, the <FileSource> directory is used - the file is renamed.

**Usage:** Setting only
SCPI confirmed

---

**MMEMory:DELete** <FileName>

Removes the specified file/files. To delete directories, use MMEMory:RDIRectory.

**Setting parameters:**

<FileName> String parameter to specify the name and directory of the file to be removed. Wildcards (* and ?) are allowed.
If no path is defined, the current directory is used, specified with MMEMory:CDIRectory.

**Example:** `MMEM:DEL '*.CFG'`
Deletes all CFG files from the current directory.

**Usage:** Setting only
SCPI confirmed

---

**MMEMory:DATA** <FileName>, <Data>
**MMEMory:DATA?** <FileName>

Stores data in the specified file to the storage location specified using MMEMory:CDIRectory.

**Parameters:**

<Data> <block>

488.2 block data format. The delimiter EOI must be selected to achieve correct data transfer.
The block begins with character '#'. The next digit is the length of the length information, followed by this given number of digits providing the number of bytes in the following binary data.

**Parameters for setting and query:**

<FileName> String parameter, the name of the file the data is stored to.

**Example:** `MMEM:DATA 'abc.txt', #216This is the file`
#2: the length information has two digits.
16: the binary data has 16 bytes.

**Example:** `MMEM:DATA? 'abc.txt'`
Returns the data from file `abc.txt`.

---

**MMEMory:ATTRibute** <FileName>, <Attributes>
**MMEMory:ATTRibute?** <FileName>

Sets file attributes for the specified file/files. The command can be used for files only.

**Setting parameters:**

<Attributes> String with attributes and setting information.
'+' before the attribute: sets the attribute
'-' before the attribute: deletes the attribute
'R': read only
'H': hidden file

**Parameters for setting and query:**

<FileName> String parameter, contains name and path of the file. Wildcards (* and ?) are allowed.

**Return values:**

<FileEntry> String containing: "<file_name>,<file_attributes>"

---

**MMEMory:SAV** <FileDestination>

Stores the current instrument settings to the specified file.

Waveform generator settings are not included.

This command has the same effect as the combination of `*SAV` and `MMEMory:STORe:STATe`.

**Parameters:**

<FileDestination> String parameter specifying path and filename of the target file. Wildcards are not allowed.

**Usage:** Event

**MMEMory:RCL** <FileSource>

Restores the instrument settings from the specified file.

The stored instrument settings do not include waveform generator settings.

This command has the same effect as the combination of `MMEMory:LOAD:STATe` and `*RCL`.

**Parameters:**

'<FileSource>' String parameter specifying the path and filename of the source file. Wildcards are not allowed.

**Usage:** Event

**MMEMory:STORe:STATe** <MemoryNumber>, <FileName>

Stores the instrument settings from the specified internal memory to the specified file. To store the current instrument settings to the internal memory, use `*SAV` first.

**Setting parameters:**

<MemoryNumber> Number of the internal memory

Range: 1 to 99

<FileName> String parameter specifying the complete path and filename of the source file.

**Usage:** Setting only

**MMEMory:LOAD:STATe** <MemoryNumber>, <FileName>

Loads the instrument settings from the specified file to the specified internal memory. After the file has been loaded, the settings must be activated using a `*RCL` command.

**Setting parameters:**

<MemoryNumber> Number of the internal memory

Range: 1 to 99

<FileName> String parameter specifying the complete path and filename of the source file.

**Usage:** Setting only

**MMEMory:GENerator:RCL** <FilePath>

Restores the waveform generator settings from the specified file.

**Parameters:**

<FilePath> String parameter specifying the path and filename of the settings file.

**Usage:** Asynchronous command

**MMEMory:GENerator:SAV** <FilePath>

Stores the current waveform generator settings to the specified file.

**Parameters:**

<FilePath> String parameter specifying path and filename of the settings file.

**Usage:** Asynchronous command

### 18.13.2 Saveset

**SAVeset:CONFig:PREView** <IncludePreview>

If set to `OFF`, the saveset is stored without the preview image to reduce the file size.

Use the command each time before you save a saveset.

**Parameters:**

<IncludePreview> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**SAVeset:CONFig:UPReferences** <UserSettings>

If ON, the user-specific display settings for the toolbar, waveform colors and diagram presentation are included in the saveset file. The setting affects the saving and the recall actions.

**Parameters:**

<UserSettings> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

### 18.13.3 Generator saveset

**GENerator:RCL** <filePath>

Restores the waveform generator from the specified file.

**Parameters:**

<filePath> String parameter specifying the path and filename of the source file.

**Usage:** Asynchronous command

**GENerator:SAV** <filePath>

Stores the current waveform generator settings to the specified file.

**Parameters:**
<filePath> String parameter specifying path and filename of the target file.

**Usage:** Asynchronous command

## 18.13.4 Waveform data: export to file

EXPort:WAVeform:ABORt ... 1127
EXPort:WAVeform:AUTonaming:NAME ... 1127
EXPort:WAVeform:AUTonaming:PATH ... 1127
EXPort:WAVeform:AUTonaming:TYPE ... 1128
EXPort:WAVeform:CURSorset ... 1128
EXPort:WAVeform:GATE ... 1128
EXPort:WAVeform:NAME ... 1128
EXPort:WAVeform:SAVE ... 1129
EXPort:WAVeform:SCOPe ... 1129
EXPort:WAVeform:SOURce ... 1130
EXPort:WAVeform:STARt ... 1131
EXPort:WAVeform:STOP ... 1132

**EXPort:WAVeform:ABORt**

Aborts a running waveform export, which was started with `EXPort:WAVeform:SAVE`.

**Usage:** Event
Asynchronous command

**EXPort:WAVeform:AUTonaming:NAME** <FileName>

Sets a name for the waveform file, without extension. The name is extended with a time stamp when the file is saved.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation. It has **no** effect on `EXPort:WAVeform:SAVE`.

**Parameters:**
<FileName> String with the filename.

**Usage:** Asynchronous command

**Manual operation:** See "File name base" on page 459

**EXPort:WAVeform:AUTonaming:PATH** <FolderPath>

Sets the directory where the waveform file is saved. For local storage, the path is always `/home/storage/userData`.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation. It has **no** effect on `EXPort:WAVeform:SAVE`.

**Parameters:**
<FolderPath> String with the path.

**Usage:** Asynchronous command

**Manual operation:** See "<Folder>" on page 459

---

**EXPort:WAVeform:AUTonaming:TYPE** <FileExtension>

Sets the file type of the waveform file.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation. It has **no** effect on `EXPort:WAVeform:SAVE`.

**Parameters:**

<FileExtension> CSV | REF | ZIP | H5

*RST: REF

**Usage:** Asynchronous command

**Manual operation:** See "File extension" on page 459

---

**EXPort:WAVeform:CURSorset** <Cursorset>

Sets the cursor set to be used for limited data export if `EXPort:WAVeform:SCOPe` is set to `CURSor`.

**Parameters:**

<Cursorset> CURSor1 | CURSor2 | CURSor3 | CURSor4

**Usage:** Asynchronous command

**Manual operation:** See "Export mode" on page 458

---

**EXPort:WAVeform:GATE** <Gate>

Selects the gate to be used for limited data export if `EXPort:WAVeform:SCOPe` is set to `GATE`.

**Parameters:**

<Gate> Number of the gate to be applied to exported data.

**Usage:** Asynchronous command

**Manual operation:** See "Export mode" on page 458

---

**EXPort:WAVeform:NAME** <name>

Sets the path, the filename and the file format of the export file. The setting is used for save-as operations that do not use the autonaming settings.

**Parameters:**

<name> String with path and filename with extension `.ref`, `.csv`, `.zip`, or `.h5`. Extension `.ref` is provided for single waveform export only, while `.zip` is for export of multiple waveforms in REF format. CSV can be used for export of single waveforms, and multiple analog channels.
For local storage, the path is always `/home/storage/userData`.

**Example:**

```
EXP:WAV:NAME '/home/storage/userData/Export_C1.csv'
EXP:WAV:SCOP DISP
EXP:WAV:SAVE
```

Saves the visible waveform data of channel 1 in xml format to `/home/storage/userData/Export_C1.csv`.

**Usage:** Asynchronous command

**Manual operation:** See "Save As" on page 459

---

**EXPort:WAVeform:SAVE**

Saves the waveform to the file specified with `EXPort:WAVeform:NAME`.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save As" on page 459

---

**EXPort:WAVeform:SCOPe** <Scope>

Defines the part of the waveform record that has to be stored.

**Parameters:**

<Scope> DISPlay | ALL | CURSor | GATE | MANual

**DISPlay**
Waveform data that is displayed in the diagram.

**ALL**
Complete waveform, which is usually longer than the displayed waveform.

**CURSor**
Data between the cursor lines if a cursor measurement is defined for the source waveform.

**GATE**
Data included in the measurement gate if a gated measurement is defined for the source waveform.

**MANual**
Saves the data between user-defined start and stop values to be set with `EXPort:WAVeform:STARt` and `EXPort:WAVeform:STOP`.

*RST: DISPlay

**Example:** See Section 18.5.4, "Data transfer in roll mode", on page 946.

**Usage:** Asynchronous command

**Manual operation:** See "Export mode" on page 458

---

**EXPort:WAVeform:SOURce** <Sources>

Selects the waveform or waveforms to be exported to file.

**Parameters:**

<Sources> Comma-separated list of waveforms

Possible waveform sources are:
Analog signals: C1,C2,...,C8
Digital signals: D0,D1,D2,...,D15
Math waveforms: M1,M2,...,M8
Reference waveforms: R1,R2,...,R8
Spectrum
traces: SPECMAXH1,SPECMINH1,SPECNORM1,SPECAVER1,SPECMAXH2,SPECMINH2,SPECNORM2,SPECAVER2,SPECMAXH3,SPECMINH3,SPECNORM3,SPECAVER3,SPECMAXH4,SPECMINH4,SPECNORM4,SPECAVER4
Tracks: TRK1,TRK2,TRK3, ...,TRK16
Power waveforms: PA1QPOWER | PA2QPOWER | PA3QPOWER | PA1HPOWER1 | PA2HPOWER1 | PA3HPOWER1| | PA1SPOWER | PA2SPOWER | PA3SPOWER | PA1IPOWER | PA2IPOWER | PA3IPOWER | PA1OPOWER1 | PA2OPOWER1 | PA3OPOWER1 | PA1OPOWER2 | PA2OPOWER2 | PA3OPOWER2 | PA1OPOWER3 | PA2OPOWER3 | PA3OPOWER3 | PA1TOPOWER | PA2TOPOWER | PA3TOPOWER | PA1SOA | PA2SOA | PA3SOA

3-phase power waveforms: PTHREE1QPOWER1 | PTHREE2QPOWER1 | PTHREE3QPOWER1 | PTHREE1QPOWER2 | PTHREE2QPOWER2 | PTHREE3QPOWER2 | PTHREE1QPOWER3 | PTHREE2QPOWER3 | PTHREE3QPOWER3 | PTHREE1QTPOWER | PTHREE2QTPOWER | PTHREE3QTPOWER | PTHREE1HPOWER1 | PTHREE2HPOWER1 | PTHREE3HPOWER1 | PTHREE1HPOWER2 | PTHREE2HPOWER2 | PTHREE3HPOWER2 | PTHREE1HPOWER3 | PTHREE2HPOWER3 | PTHREE3HPOWER3 | PTHREE1HTPOWER | PTHREE2HTPOWER | PTHREE3HTPOWER | PTHREE1EIP1| PTHREE2EIP1| PTHREE3EIP1| PTHREE4EIP1| PTHREE5EIP1| PTHREE6EIP1| PTHREE1EIP2| PTHREE2EIP2| PTHREE3EIP2| PTHREE4EIP2| PTHREE5EIP2| PTHREE6EIP2| PTHREE1EIP3| PTHREE2EIP3| PTHREE3EIP3| PTHREE4EIP3| PTHREE5EIP3| PTHREE6EIP3| PTHREE1EITP| PTHREE2EITP| PTHREE3EITP| PTHREE4EITP| PTHREE5EITP| PTHREE6EITP | PTHREE1EOP1| PTHREE2EOP1| PTHREE3EOP1| PTHREE4EOP1| PTHREE5EOP1| PTHREE6EOP1| PTHREE1EOP2| PTHREE2EOP2| PTHREE3EOP2| PTHREE4EOP2| PTHREE5EOP2| PTHREE6EOP2| PTHREE1EOP3| PTHREE2EOP3| PTHREE3EOP3| PTHREE4EOP3| PTHREE5EOP3| PTHREE6EOP3| PTHREE1EOTP| PTHREE2EOTP| PTHREE3EOTP| PTHREE4EOTP| PTHREE5EOTP| PTHREE6EOTP

**Example:**

```
EXP:WAV:NAME '/home/storage/userData/Export_C2.csv'
EXPort:WAVeform:SOURce C2
EXP:WAV:SCOP DISP
EXP:WAV:SAVE
```

Exports the data of channel 2 to a CSV file.

**Example:**

```
EXP:WAV:NAME '/home/storage/userData/Export_multi.zip'
EXPort:WAVeform:SOURce C1,C2,SPECNORM1,SPECAVER1
EXP:WAV:SCOP DISP
EXP:WAV:SAVE
```

Exports the data of channel 1, channel 2, normal and average spectrum traces to a zip file.

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 457

---

**EXPort:WAVeform:STARt** <Start>

Sets the start time value of the waveform section for export, if `EXPort:WAVeform:SCOPe` on page 1129 is set to `Manual`.

**Parameters:**

<Start>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0.01
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Export mode" on page 458

---

**EXPort:WAVeform:STOP** <Stop>

Sets the end time value of the waveform section for export, if `EXPort:WAVeform:SCOPe` on page 1129 is set to `Manual`.

**Parameters:**

<Stop>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0.01
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Export mode" on page 458

### 18.13.5 Histogram data: export to file

You can export the data of waveform histograms, and the data of histograms on measurement results. One histogram per file can be exported.

EXPort:HMEasurement<m>:NORMalize ..... 1132
EXPort:HISTogram<m>:NORMalize ..... 1132
EXPort:HMEasurement<m>:NAME ..... 1133
EXPort:HISTogram<m>:NAME ..... 1133
EXPort:HMEasurement<m>:SAVE ..... 1133
EXPort:HISTogram<m>:SAVE ..... 1133

---

**EXPort:HMEasurement<m>:NORMalize** <Normalization>
**EXPort:HISTogram<m>:NORMalize** <Normalization>

If normalization is off, the number of samples in a given bin is exported as integer values. With normalization, the ratio of samples in a given bin to the sum of all samples is exported as float value. The sum of all normalized values is 1.

The command affects all histogram exports.

**Suffix:**

<m> Irrelevant, omit the suffix.

**Parameters:**

<Normalization> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Normalization" on page 475

---

**EXPort:HMEasurement<m>:NAME** <SaveAsPath>
**EXPort:HISTogram<m>:NAME** <SaveAsPath>

Sets the path, the filename and the file format of the export file for histogram. The command affects all histogram exports.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Parameters:**
<SaveAsPath> String with path and file name with extension `.csv`.

**Usage:** Asynchronous command

**Manual operation:** See "Save settings" on page 475

---

**EXPort:HMEasurement<m>:SAVE**

Saves the histogram on the results of the specified measurement to file. The target file is set using `EXPort:HMEasurement<m>:NAME`.

**Suffix:**
<m> 1 to 16, index of the measurement

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save settings" on page 475

---

**EXPort:HISTogram<m>:SAVE**

Saves the specified histogram to file. The target file is set using `EXPort:HISTogram<m>:NAME` on page 1133.

**Suffix:**
<m> 1...8, index of the histogram

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save settings" on page 475

### 18.13.6 Sessions

SESSion:LOAD[:EXECute] ..... 1134
SESSion:SAVE:ABORt ..... 1134
SESSion:SAVE:CHANnel ..... 1134

SESSion:SAVE:REFerence ..... 1134
SESSion:SAVE[:EXECute] ..... 1135
SESSion:USERpref ..... 1135

**SESSion:LOAD[:EXECute]** <Filepath>

Loads a session file.

To reload also user-specific display settings (if included in the session file), set `SESSion:USERpref ON`.

**Setting parameters:**

<Filepath> String parameter specifying path and filename of the session file.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 481

**SESSion:SAVE:ABORt**

Stops the saving process of a session file.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save settings" on page 480

**SESSion:SAVE:CHANnel** <ChWfms>

Includes the channel waveform data in the session file.

**Parameters:**

<ChWfms> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Channel waveforms" on page 479

**SESSion:SAVE:REFerence** <RefWfms>

Includes the reference waveform data in the session file.

**Parameters:**

<RefWfms> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Reference waveforms" on page 479

**SESSion:SAVE[:EXECute]** <Filepath>

Saves the current session with selected content to the specified file.

**Setting parameters:**

<Filepath> String parameter specifying path and filename of the target file.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save settings" on page 480

**SESSion:USERpref** <UserSettings>

If ON, the user-specific display settings for the toolbar, waveform colors and diagram presentation are included in the session file. The setting affects the saving and the recall actions.

**Parameters:**

<UserSettings> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Toolbar, color and diagram settings" on page 479

## 18.13.7 Results

EXPort:RESult:AUTonaming:NAME....1135
EXPort:RESult:AUTonaming:PATH....1136
EXPort:RESult:AUTonaming:TYPE....1136
EXPort:RESult:NAME....1136
EXPort:RESult:SAVE....1136
EXPort:RESult:SELect:CURSor....1136
EXPort:RESult:SELect:MEASurement....1137
EXPort:RESult:SELect:PLISt....1137

**EXPort:RESult:AUTonaming:NAME** <FileName>

Sets a name for the result file, without extension. The name is extended with a time stamp when the file is saved.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation. It has **no** effect on `EXPort:RESult:SAVE`.

**Parameters:**

<FileName> String with the filename.

**Usage:** Asynchronous command

**Manual operation:** See "File name base" on page 477

**EXPort:RESult:AUTonaming:PATH** <FolderPath>

Sets the directory where the waveform file is saved. For local storage, the path is always `/home/storage/userData`.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation. It has **no** effect on `EXPort:RESult:SAVE`.

**Parameters:**
<FolderPath> String with the path.

**Usage:** Asynchronous command

**Manual operation:** See "<Folder>" on page 476

**EXPort:RESult:AUTonaming:TYPE** <FileExtension>

Sets the file type of the waveform file.

The setting is used for automatic saving actions, for example, saving on trigger or mask violation. It has **no** effect on `EXPort:RESult:SAVE`.

**Parameters:**
<FileExtension> CSV

**Usage:** Asynchronous command

**Manual operation:** See "File extension" on page 477

**EXPort:RESult:NAME** <FilePath>

Sets the path, the filename and the file format of the export file.

**Parameters:**
<FilePath> String with path and file name with extension `.csv`.

**Usage:** Asynchronous command

**Manual operation:** See "Save As" on page 477

**EXPort:RESult:SAVE**

Saves the results to file. The target file is set using `EXPort:RESult:NAME`. To select the results to be exported, use `EXPort:RESult:SELect:CURSor` and `EXPort:RESult:SELect:MEASurement`.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save As" on page 477

**EXPort:RESult:SELect:CURSor** <CursorResult>

Includes the current cursor results in the export file.

**Parameters:**
<CursorResult> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Save results name" on page 476

---

**EXPort:RESult:SELect:MEASurement** <MeasResult>

Includes the current automatic measurement results in the export file.

**Parameters:**
<MeasResult> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Save results name" on page 476

---

**EXPort:RESult:SELect:PLISt** <SpectrumPeakList>

Includes the peak list results in the export file.

**Parameters:**
<SpectrumPeakList> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Save results name" on page 476

## 18.13.8 Screenshots

The `HCOPy` subsystem and some other commands control the output of display information for documentation purposes. The instrument allows two independent output configurations which can be set separately with the suffix.

Note that the remote mode is intended for maximum performance. Therefore, the display does not follow the remote commands constantly. To get a correct screenshot, turn the display on using `SYSTem:DISPlay:UPDate`.

HCOPy:DESTination<m> 1138
MMEMory:NAME 1138
HCOPy:DEVice<m>:LANGuage 1138
HCOPy:DEVice<m>:INVerse 1138
HCOPy:IMMediate<m>:NEXT 1139
HCOPy:IMMediate<m> 1139
HCOPy:ISBA 1140
HCOPy:SSD 1140
HCOPy:WBKG 1140
HCOPy:DATA? 1140

**HCOPy:DESTination<m>** <medium>

Selects the output medium: file or clipboard.

**Suffix:**
<m> 1..2
Selects the output configuration.

**Parameters:**
<medium> MMEM | CLIPBOARD
`MMEM`: saves image to a file.
`CLIPBOARD`: directs the image to the clipboard.
*RST: MMEM

**Usage:** Asynchronous command

**Manual operation:** See "<Folder>" on page 485

**MMEMory:NAME** <FileName>

Defines the filename for a screenshot that is stored to a file.

**Parameters:**
<FileName> String parameter specifying path and filename of the screenshot.

**Example:** See Section 18.5.3, "Saving screenshots to file", on page 945.

**Usage:** SCPI confirmed
Asynchronous command

**Manual operation:** See "File name base" on page 485

**HCOPy:DEVice<m>:LANGuage** <FileFormat>

Defines the file format for output of the screenshot to file.

To set the output to file, use `HCOPy:DESTination<m>` with parameter `MMEM`.

**Suffix:**
<m> 1..2
Selects the output configuration.

**Parameters:**
<FileFormat> PNG | JPG
*RST: PNG

**Usage:** Asynchronous command

**Manual operation:** See "File extension" on page 485

**HCOPy:DEVice<m>:INVerse** <InverseColor>

Inverts the colors of the output, i.e. a dark waveform is shown on a white background.

See also `HCOPy:WBKG` and White background.

**Suffix:**
<m> 1..2
Selects the output configuration.

**Parameters:**
<InverseColor> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Inverse color" on page 484

---

**HCOPy:IMMediate<m>:NEXT**

Starts the output of the next display image, depending on the `HCOPy:DESTination<m>` destination setting.

If the screenshot is saved to a file, the file name used in the last saving process is automatically counted up to the next unused name. To define the file name, use `MMEMory:NAME`.

**Suffix:**
<m> 1..2
Selects the output configuration.

**Example:** See Section 18.5.3, "Saving screenshots to file", on page 945.

**Usage:** Asynchronous command

**Manual operation:** See "Save" on page 485

---

**HCOPy:IMMediate<m>**

Starts the immediate output of the display image, depending on the `HCOPy:DESTination<m>` destination setting.

To define the file name, use `MMEMory:NAME`. Existing files are overwritten by the `HCOP:IMMM` command.

To get a correct screenshot of the diagrams, results, and dialog boxes, turn on the display using `SYSTem:DISPlay:UPDate`.

**Suffix:**
<m> 1..2
Selects the output configuration.

**Example:** See Section 18.5.3, "Saving screenshots to file", on page 945.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save" on page 485

**HCOPy:ISBA** <IncludeSignBarScpi>

If enabled, the screenshot shows the signal bar below the diagram area.

**Parameters:**
<IncludeSignBarScpi>OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Include signal bar" on page 484

**HCOPy:SSD** <ShowSetupDialog>

If enabled, the currently open dialog box is included in the screenshot.

**Parameters:**
<ShowSetupDialog> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show setup dialog" on page 484

**HCOPy:WBKG** <WhiteBackground>

Inverts the background color, so you can picture waveforms with normal waveform colors on white background.

If both `HCOPy:WBKG` and `HCOPy:DEVice<m>:INVerse` are ON, the instrument inverts the background twice, and it appears black.

**Parameters:**
<WhiteBackground> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "White background" on page 484

**HCOPy:DATA?**

Creates a PNG screenshot and returns the data of the image file in a binary data stream. When receiving the data, write them into a PNG file which you can open later.

**Usage:** Query only
Asynchronous command

## 18.13.9 Data transfer to remote computer

The commands in this section return the data of waveform points or histogram bins for transmission from the instrument to the controlling computer. The data can be used in MATLAB, for example.

Usually, for each waveform type a `...:HEADer` command and a `...:VALue` command are available. For some waveforms, additional commands provided. For fast export of several waveforms at once, you can use `EXPort:WAVeform:DATA[:VALues]?`.

**<Offset> and <Length> parameters in ...DATA[:VALues] commands**

Without parameters, the complete waveform is retrieved. Using the offset and length parameters, data can be retrieved in smaller portions, which makes the command faster. If you send only one parameter, it is interpreted as offset, and the data is retrieved from offset to the end of the waveform.

- <Offset>:
  Number of offset waveform points to be skipped. The range is 0 to m, and the limit is *n + m <= record length*.
- <Length>:
  Number of waveform points to be retrieved. The range is 0 to n, and the limit is *n + m <= record length*.

CHANnel<ch>:DATA:HEADer? 1142
CHANnel<ch>:DATA[:VALues]? 1142
REFCurve<rc>:DATA:HEADer? 1143
REFCurve<rc>:DATA[:VALues]? 1144
REFCurve<rc>:DATA:STYPe? 1144
CALCulate:MATH<m>:DATA:HEADer? 1144
CALCulate:MATH<m>:DATA[:VALues]? 1145
CALCulate:MATH<m>:DATA:STYPe? 1145
CALCulate:SPECtrum<sp>:WAVeform:AVERage:DATA:HEADer? 1145
CALCulate:SPECtrum<sp>:WAVeform:MAXimum:DATA:HEADer? 1145
CALCulate:SPECtrum<sp>:WAVeform:MINimum:DATA:HEADer? 1145
CALCulate:SPECtrum<sp>:WAVeform:NORMal:DATA:HEADer? 1145
CALCulate:SPECtrum<sp>:WAVeform:AVERage:DATA[:VALues]? 1146
CALCulate:SPECtrum<sp>:WAVeform:MAXimum:DATA[:VALues]? 1146
CALCulate:SPECtrum<sp>:WAVeform:MINimum:DATA[:VALues]? 1146
CALCulate:SPECtrum<sp>:WAVeform:NORMal:DATA[:VALues]? 1146
CALCulate:SPECtrum<sp>:WAVeform:NORMal:DATA:FRAMe<fr>[:VALues]? 1146
DIGital<m>:DATA:HEADer? 1147
DIGital<m>:DATA[:VALues]? 1147
PBUS<pb>:DATA:FORMat 1148
PBUS<pb>:DATA:HEADer? 1148
PBUS<pb>:DATA[:VALues]? 1148
EXPort:HMEasurement<m>:DATA:HEADer? 1149
EXPort:HISTogram<m>:DATA:HEADer? 1149
EXPort:HMEasurement<m>:DATA[:VALues]? 1150
EXPort:HISTogram<m>:DATA[:VALues]? 1150
EXPort:WAVeform:DATA[:VALues]? 1150

**CHANnel<ch>:DATA:HEADer?**

Returns the header of channel waveform data, the attributes of the waveform.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Return values:**
<XStart> 1. header value: time of the first sample in s

<XStop> 2. header value: time of the last sample in s

<RecordLength> 3. header value: record length, number of samples

<ValuesPerSample> 4. header value: number of values per sample. For most waveforms, the result is 1. For peak detect and envelope waveforms, it is 2. If the number is 2, the number of returned values is twice the number of samples (record length).

**Example:**
```
CHAN1:DATA:HEAD?
-1E-07,9.980000000000001E-08,1000,1
```
Start time of the data is -1E-07 = 100 ns, and end time of the data is 9.980000000000001E-08 = 99.8 ns. The data stream has 1000 values with one value per sample.

**Usage:** Query only
Asynchronous command

**CHANnel<ch>:DATA[:VALues]?** [<Offset>[,<Length>]]

Returns the data of the specified channel waveform.

To set the export format, use `FORMat[:DATA]` on page 950.

**Suffix:**
<ch> 1 to 8, index of the analog channel

**Query parameters:**
<Offset>, <Length> See "<Offset> and <Length> parameters in ...DATA[:VALues] commands" on page 1141.

**Return values:**
<Data> List of values according to the format and content settings.

**Example:** Retrieve the complete channel 1 waveform, only Y-values:
```
FORM ASC
CHAN1:DATA?
<-- -0.125000,-0.123016,-0.123016,-0.123016,-0.123016,-0.123016,...
```

**Example:** Retrieve the first 10 values of the waveform:
```
CHANnel:DATA:VALues? 0,10
<-- -0.10079051554203,-0.098814234137535,-0.098814234137535,
    -0.096837945282459,-0.094861663877964,-0.094861663877964,
    -0.092885382473469,-0.090909093618393,-0.090909093618393,
    -0.088932812213898
```

**Example:** Skip 5 samples and retrieve the next 5 samples:

```
CHANnel:DATA:VALues? 5,5
<-- -0.094861663877964,-0.092885382473469,-0.090909093618393,
    -0.090909093618393,-0.088932812213898
```

**Usage:** Query only
Asynchronous command

---

**REFCurve<rc>:DATA:HEADer?**

Returns header information on the reference waveform, the attributes of the waveform.

The information depends on the waveform domain, it is different for time domain and frequency domain reference waveforms. See:

- `CHANnel<ch>:DATA:HEADer?` on page 1142
- `CALCulate:SPECtrum<sp>:WAVeform:NORMal:DATA:HEADer?` on page 1145

**Suffix:**

<rc> 1 to 8, index of the reference waveform

**Return values:**

<XStart> 1. header value: time of the first sample in s for a time domain signal, or start frequency of the first spectrum bin in Hz for a frequency domain signal

<XStop> 2. header value: time of the last sample in s for a time domain signal, or start frequency of the last spectrum bin in Hz for a frequency domain signal

<RecordLength> 3. header value: record length of the waveform in samples or bins

<ValuesPerSample> 4. header value: number of values per sample or bin. The number depends on the source waveform from which the reference waveform was created.

**Example:** Time domain signal:

```
REFC:DATA:HEAD?
-1E-07,9.980000000000001E-08,1000,1
```

Start time of the data is -1E-07= -100 ns and stop time of the data is 9.980000000000001E-08= 99.8 ns. The data stream has 1000 values with one value per sample.

**Example:** Frequency domain signal:

```
REFC:DATA:HEAD?
549316.1734194756,1999450683.360919,2621,1
```

Start frequency of the first spectrum bin is 549316.1734194756 = ~ 549.3 kHz and start frequency of the last spectrum bin is 1999450683.360919 = ~ 1.999 GHz. The data stream has 2621 values with one value per bin.

**Usage:** Query only
Asynchronous command

---

**REFCurve<rc>:DATA[:VALues]?** [<Offset>[,<Length>]]

Returns the data of the specified reference waveform.

To set the export format, use `FORMat[:DATA]` on page 950.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Query parameters:**
<Offset>, <Length> See "<Offset> and <Length> parameters in ...DATA[:VALues] commands" on page 1141.

**Return values:**
<Data> List of values according to the format and content settings.

**Usage:** Query only
Asynchronous command

---

**REFCurve<rc>:DATA:STYPe?**

Returns the type of the source signal from which the reference waveform was created.

**Suffix:**
<rc> 1 to 8, index of the reference waveform

**Return values:**
<SignalType> NONE | CHANNEL | MATH | TRACK | REFERENCE | SPECTRUM | FRA_REF

**Usage:** Query only
Asynchronous command

---

**CALCulate:MATH<m>:DATA:HEADer?**

Returns the header of math waveform data, the attributes of the waveform.

**Suffix:**
<m> 1 to 8, index of the math waveform

**Return values:**
<XStart> 1. header value: time of the first sample in s

<XStop> 2. header value: time of the last sample in s

<RecordLength> 3. header value: record length of the waveform in samples

<ValuesPerSample> 4. header value: number of values per sample. For most waveforms, the result is 1. For peak detect and envelope waveforms, it is 2. If the number is 2, the number of returned values is twice the number of samples (record length).

**Example:** `CALC:MATH1:DATA:HEAD?`
`-1E-07,9.980000000000001E-08,1000,1`
Start time of the data is -1E-07 = 100 ns, and end time of the data is 9.980000000000001E-08 = 99.8 ns. The data stream has 1000 values with one value per sample.

**Usage:** Query only
Asynchronous command

---

**CALCulate:MATH<m>:DATA[:VALues]?** [<Offset>[,<Length>]]

Returns the data of the specified math waveform.

To set the export format, use `FORMat[:DATA]`.

**Suffix:**
<m> 1 to 8, index of the math waveform

**Query parameters:**
<Offset>, <Length> See "<Offset> and <Length> parameters in ...DATA[:VALues] commands" on page 1141.

<Length> Number of waveform points to be retrieved.

**Example:** `:CALC:MATH1:DATA:VAL? 100,10`
Returns 10 points from the start index 100.

**Usage:** Query only
Asynchronous command

---

**CALCulate:MATH<m>:DATA:STYPe?**

Returns the signal type of the math waveform.

**Suffix:**
<m> 1 to 8, index of the math waveform

**Return values:**
<SignalType> MATH

**Usage:** Query only
Asynchronous command

---

**CALCulate:SPECtrum<sp>:WAVeform:AVERage:DATA:HEADer?**
**CALCulate:SPECtrum<sp>:WAVeform:MAXimum:DATA:HEADer?**
**CALCulate:SPECtrum<sp>:WAVeform:MINimum:DATA:HEADer?**
**CALCulate:SPECtrum<sp>:WAVeform:NORMal:DATA:HEADer?**

Returns the header of spectrum data, the attributes of the waveform.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Return values:**
<XStart> 1. header value: start frequency of the first spectrum bin in Hz

<XStop> 2. header value: start frequency of the last spectrum bin in Hz

<RecordLength> 3. header value: record length of the waveform in bins

<ValuesPerSample> 4. header value: the number of values per bin is always = 1.

**Example:**
```
CALC:SPEC1:WAV:NORM:DATA:HEAD?
549316.1734194756,1999450683.360919,2621,1
```
Start frequency of the first spectrum bin is 549316.1734194756 = ~ 549.3 kHz. Start frequency of the last spectrum bin is 1999450683.360919 = ~ 1.999 GHz. The data stream has 2621 values with one value per bin.

**Usage:** Query only
Asynchronous command

---

**CALCulate:SPECtrum<sp>:WAVeform:AVERage:DATA[:VALues]?**
[<Offset>[,<Length>]]
**CALCulate:SPECtrum<sp>:WAVeform:MAXimum:DATA[:VALues]?**
[<Offset>[,<Length>]]
**CALCulate:SPECtrum<sp>:WAVeform:MINimum:DATA[:VALues]?**
[<Offset>[,<Length>]]
**CALCulate:SPECtrum<sp>:WAVeform:NORMal:DATA[:VALues]?**
[<Offset>[,<Length>]]

Returns the data of the specified spectrum trace.

To set the export format, use `FORMat[:DATA]` on page 950.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Query parameters:**
<Offset>, <Length> See "<Offset> and <Length> parameters in ...DATA[:VALues] commands" on page 1141.

**Return values:**
<Data> List of values according to the format and content settings.

**Usage:** Query only
Asynchronous command

---

**CALCulate:SPECtrum<sp>:WAVeform:NORMal:DATA:FRAMe<fr>[:VALues]?**
[<Offset>[,<Length>]]

Returns the data of the normal spectrum trace of the specified frame.

To set the export format, use `FORMat[:DATA]` on page 950.

**Suffix:**
<sp> 1 to 4, index of the spectrum

<fr> 1..*
Index of the frame

**Query parameters:**

<Offset>, <Length> See "<Offset> and <Length> parameters in ...DATA[:VALues] commands" on page 1141.

**Return values:**

<Data> List of values according to the format and content settings.

**Usage:** Query only
Asynchronous command

---

**DIGital<m>:DATA:HEADer?**

Returns the header of digital channel data, the attributes of the waveform.

**Suffix:**

<m> 0 to 15, index of the logic channel

**Return values:**

<XStart> 1. header value: time of the first sample in s

<XStop> 2. header value: time of the last sample in s

<RecordLength> 3. header value: record length of the waveform in samples

<ValuesPerSample> 4. header value: number of values per sample. For digital data, the result is always 1.

**Example:**
```
DIG3:DATA:HEAD?
-1E-07,9.980000000000001E-08,1000,1
```
Start time of the data is -1E-07 = -100 ns, and end time of the data is 9.980000000000001E-08= 99.8 ns. The data stream has 1000 values with one value per sample.

**Usage:** Query only
Asynchronous command

---

**DIGital<m>:DATA[:VALues]?** [<Offset>[,<Length>]]

Returns the data of the specified digital channel.

**Suffix:**

<m> 0..15
Selects the digital channel.

**Query parameters:**

<Offset>, <Length> See "<Offset> and <Length> parameters in ...DATA[:VALues] commands" on page 1141.

**Return values:**

<Data> List of values according to the format and content settings.

**Usage:** Query only
Asynchronous command

**PBUS<pb>:DATA:FORMat** <DataFormat>

Sets the data format of bus values, which are displayed in the decode table and on the comb bus display.

It also sets the format for the number representation for remote data transfer with `PBUS<pb>:DATA[:VALues]?`.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<DataFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG
*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Data format" on page 895

**PBUS<pb>:DATA:HEADer?**

Returns the header data of the indicated bus, the attributes of the data.

See also `DIGital<m>:DATA:HEADer?`.

**Suffix:**
<pb> 1...4, index of the logic group

**Return values:**
<XStart> 1. header value: time of the first sample in s

<XStop> 2. header value: time of the last sample in s

<RecordLength> 3. header value: record length of the waveform in samples

<ValuesPerSample> 4. header value: number of values per sample. For digital data, the result is always 1.

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Data format" on page 895

**PBUS<pb>:DATA[:VALues]?** [<Offset>[,<Length>]]

Returns the data of the specified logic bus.

Without parameters, the complete waveform is retrieved. Using the offset and length parameters, data can be retrieved in smaller portions, which makes the command faster. If you send only one parameter, it is interpreted as offset, and the data is retrieved from offset to the end of the waveform.

Requirements:

- `PBUS<pb>:STATe` is set to `ON`.
- `PBUS<pb>:DISPlay:SHBU` is set to `ON`.

- A number format is set with `PBUS<pb>:DATA:FORMat`.

**Suffix:**
<pb> 1...4, index of the logic group

**Query parameters:**
<Offset>, <Length> See "<Offset> and <Length> parameters in ...DATA[:VALues] commands" on page 1141.

**Return values:**
<Data> List of values according to the format and content settings.

**Example:**
```
PBUS:STAT ON
PBUS:DISP:SHBU ON
PBUS:DISP:BTYP COMB
PBUS:DATA:FORMat HEX
PBUS:DATA:VAL?
```

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Data format" on page 895

---

**EXPort:HMEasurement<m>:DATA:HEADer?**
**EXPort:HISTogram<m>:DATA:HEADer?**

Returns the header of the histogram data, the attributes of the waveform histogram or of the histogram on measurement results.

**Suffix:**
<m> 1 to 16, index of the measurement
or 1...8, index of the histogram

**Return values:**
<Start> 1. header value: start value. For measurement results in the measurement unit, for vertical waveform histogram in the waveform unit, for horizontal waveform histogram in s.

<End> 2. header value: end value, same units as for the start value.

<HistogrLength> 3. header value: number of histogram bins

**Example:**
```
EXP:HIST3:DATA:HEAD?
-0.250244,0.250122,1024
```
Start value of the histogram is about -250 mV, and end value is about 250 mV. The data stream has 1024 values, one value for each bin.

**Usage:** Query only
Asynchronous command

**EXPort:HMEasurement<m>:DATA[:VALues]?**
**EXPort:HISTogram<m>:DATA[:VALues]?**

Returns the data of the specified waveform histogram or of the histogram on measurement results.

To set the export format, use `FORMat[:DATA]`. For histogram data, only ASCii and REAL,32 and REAL,64 are supported. The normalization setting is considered: `EXPort:HISTogram<m>:NORMalize` or `EXPort:HMEasurement<m>:NORMalize`.

**Suffix:**
<m> 1 to 16, index of the measurement
or 1...8, index of the histogram

**Return values:**
<Data> Comma-separated list of values according to the format setting.

**Usage:** Query only
Asynchronous command

**EXPort:WAVeform:DATA[:VALues]?**

Starts a fast export of the selected waveforms for use in automated analysis systems. There is no corresponding `...:DATA:HEADer` command. Offset and length parameters are not available.

Select the waveforms for export with `EXPort:WAVeform:SOURce`, and the data scope with `EXPort:WAVeform:SCOPe`.

The data stream contains the pure data of the exported sources. The order of the sources can be read with `EXPort:WAVeform:SOURce?`. The order is predefined and independent of the setting order to ensure the same order for the same selection of waveforms.

Fast export is possible at low record length, depending on the instrument and active functionality. If fast export is not possible, you can export the single signals.

**Parameters:**
<Values> Binary format: stream of bytes, without delimiter
ASCII: comma-separated value list

**Example:** The following waveforms are active and acquired: C1, C2 and SPECNORM1.
Determine the number of values per waveform:

```
CHANnel1:WAVeform:DATA:HEADer? --> 1000 values
CHANnel2:WAVeform:DATA:HEADer?      --> 1000 values
CALCulate:SPECtrum1:WAVeform:NORMal:DATA:HEADer? --> 2867 values
```

Fast export of all waveforms at once as float values:

```
FORMat:DATA REAL, 32
EXPort:WAVeform:SOURce C2,C1,SPECNORM1
EXPort:WAVeform:SOURce?      --> C1,C2,SPECNORM1
EXPort:WAVeform:DATA:VALues?
--> data stream of 1000 values C1, then 1000 values C2, then 2867 values
```

If the fast export fails, use the export of single waveforms:

```
CHANnel1:WAVeform:DATA:VALues?
CHANnel2:WAVeform:DATA:VALues?
CALCulate:SPECtrum1:WAVeform:NORMal:DATA:VALues?
```

**Usage:** Query only
Asynchronous command

## 18.14 Automatic measurements

This section contains all remote commands to set up automatic measurements and to analyze the measurement results.

**Measurement suffix <mg>**

The suffix <mg> indicates the number of the measurement for which the command takes effect.

### 18.14.1 General settings

MEASurement<mg>:ACTive....1152
MEASurement<mg>:COUNt?....1152
MEASurement<mg>[:ENABle]....1152
MEASurement<mg>:SOURce....1152
MEASurement<mg>:FSRC....1153
MEASurement<mg>:SSRC....1154
MEASurement<mg>:MAIN....1154
MEASurement<mg>:GATE....1155
MEASurement<mg>:STATistics:RESet....1155
MEASurement<mg>:CLEar....1155
MEASurement<mg>:DISPlay:RESults....1155
MEASurement<mg>:DISPlay:REFLevel<rl>:LEVels....1155
MEASurement<mg>:ENVSelect....1156

**MEASurement<mg>:ACTive** <First>

Activates or deactivates the indicated measurement.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<First> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Add" on page 276

**MEASurement<mg>:COUNt?**

Returns the maximum number of measurements, which is the maximum value for the <mg> suffix.

**Suffix:**
<mg> Irrelevant, omit the suffix.

**Return values:**
<Count> Maximum number of measurements

**Usage:** Query only
Asynchronous command

**MEASurement<mg>[:ENABle]** <First>

Selects the indicated measurement.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<First> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Add" on page 276

**MEASurement<mg>:SOURce** <SignalSource>,[<SignalSource2>]

Sets the source of the measurement.

The source must fit the selected measurement type. For example, histogram measurements require a waveform histogram as source, and protocol measurements require a serial bus.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**

<SignalSource> NONE | C1 | C2 | ... | C8 | D0 | D1 | D2 | ... | D14 | D15 | M1 | M2 | ... | M8 | TRK1 | TRK2 | TRK3 | ... | TRK16 | HWAVeform1 | HWAVeform2 | ... | HWAVeform8 | PA1QPOWER | PA2QPOWER | PA3QPOWER | PA1HPOWER1 | PA2HPOWER1 | PA3HPOWER1| | PA1SPOWER | PA2SPOWER | PA3SPOWER | PA1SOA | PA2SOA | PA3SOA | PA1IPOWER | PA2IPOWER | PA3IPOWER | PA1OPOWER1 | PA2OPOWER1 | PA3OPOWER1 | PA1OPOWER2 | PA2OPOWER2 | PA3OPOWER2 | PA1OPOWER3 | PA2OPOWER3 | PA3OPOWER3 | PTHREE1QPOWER1 | PTHREE2QPOWER1 | PTHREE3QPOWER1 | PTHREE1QPOWER2 | PTHREE2QPOWER2 | PTHREE3QPOWER2 | PTHREE1QPOWER3 | PTHREE2QPOWER3 | PTHREE3QPOWER3 | PTHREE1QTPOWER | PTHREE2QTPOWER | PTHREE3QTPOWER | PTHREE1HPOWER1 | PTHREE2HPOWER1 | PTHREE3HPOWER1 | PTHREE1HPOWER2 | PTHREE2HPOWER2 | PTHREE3HPOWER2 | PTHREE1HPOWER3 | PTHREE2HPOWER3 | PTHREE3HPOWER3 | PTHREE1HTPOWER | PTHREE2HTPOWER | PTHREE3HTPOWER | PTHREE1EIP1| PTHREE2EIP1| PTHREE3EIP1| PTHREE4EIP1| PTHREE5EIP1| PTHREE6EIP1| PTHREE1EIP2| PTHREE2EIP2| PTHREE3EIP2| PTHREE4EIP2| PTHREE5EIP2| PTHREE6EIP2| PTHREE1EIP3| PTHREE2EIP3| PTHREE3EIP3| PTHREE4EIP3| PTHREE5EIP3| PTHREE6EIP3| PTHREE1EITP| PTHREE2EITP| PTHREE3EITP| PTHREE4EITP| PTHREE5EITP| PTHREE6EITP | PTHREE1EOP1| PTHREE2EOP1| PTHREE3EOP1| PTHREE4EOP1| PTHREE5EOP1| PTHREE6EOP1| PTHREE1EOP2| PTHREE2EOP2| PTHREE3EOP2| PTHREE4EOP2| PTHREE5EOP2| PTHREE6EOP2| PTHREE1EOP3| PTHREE2EOP3| PTHREE3EOP3| PTHREE4EOP3| PTHREE5EOP3| PTHREE6EOP3| PTHREE1EOTP| PTHREE2EOTP| PTHREE3EOTP| PTHREE4EOTP| PTHREE5EOTP| PTHREE6EOTP | PA1TOPOWER | PA2TOPOWER | PA3TOPOWER

<SignalSource2> Same as <SignalSource>

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 275

---

**MEASurement<mg>:FSRC** <Source>

Defines the first measurement source.

The command is an alternative to `MEASurement<mg>:SOURce`.

**Suffix:**

<mg> 1 to 16, index of the measurement

**Parameters:**
<Source> See MEASurement<mg>:SOURce on page 1152.

**Usage:** Asynchronous command

---

**MEASurement<mg>:SSRC** <Source2>

Defines the second measurement source.

The command is an alternative to MEASurement<mg>:SOURce.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<Source2> See MEASurement<mg>:SOURce on page 1152.

**Usage:** Asynchronous command

**Manual operation:** See "Source 2, Clock source" on page 287

---

**MEASurement<mg>:MAIN** <MeasType>

Defines the measurement type that is assigned to the indicated measurement. To query the result, use MEASurement<mg>:RESult[:ACTual]? on page 1166.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<MeasType>

**Amplitude/time measurements**
HIGH | LOW | AMPLitude | MAXimum | MINimum | PDELta | MEAN | RMS | STDDev | CRESt | POVershoot | NOVershoot | AREA | RTIMe | FTIMe | PPULse | NPULse | PERiod | FREQuency | PDCYcle | NDCYcle | CYCarea | CYCMean | CYCRms | CYCStddev | CYCCrest | CAMPlitude | CMAXimum | CMINimum | CPDelta | PULCnt | DELay | PHASe | BWIDth | EDGecount | SETup | HOLD | SHT | SHR | DTOTrigger | SLERising | SLEFalling

**Jitter measurements**
CCJitter | NCJitter | CCWidth | CCDutycycle | TIE | UINTerval | DRATe | SKWDelay | SKWPhase
Require option R&S MXO3-K12.

**Measurements on waveform histograms**
WCOunt | HSAMples | HPEak | PEAK | UPEakvalue | LPEakvalue | HMAXimum | HMINimum | MEDian | MAXMin | HMEan | HSTDdev | M1STddev | M2STddev | M3STddev | MKPositive | MKNegative

**Measurements for serial protocols**
F2F | T2F | F2T | FLDValue | MBITrate | SBITrate | BIDLe | GAP | FCNT | FEC | FER | CFER
Require option R&S MXO3-K500.

**Usage:** Asynchronous command

**Manual operation:** See "Select measurements" on page 276

---

**MEASurement<mg>:GATE** <Gate>

Sets the gate of the indicated measurement. Enable a gate before you assign a measurement to it (`GATE<g>:ENABle` =`ON`).

The query returns 0, if no gate is assigned.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<Gate> Number of the gate to be used

**Usage:** Asynchronous command

**Manual operation:** See "Gate" on page 278

---

**MEASurement<mg>:STATistics:RESet**
**MEASurement<mg>:CLEar**

Deletes the results of the specified measurement.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Clear results" on page 277

---

**MEASurement<mg>:DISPlay:RESults** <ResultLines>

Enables the measurement annotations for the selected measurement. These annotations are, for example, periods, maximum and minimum values, relevant reference levels, and more.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<ResultLines> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Display result lines" on page 278

---

**MEASurement<mg>:DISPlay:REFLevel<rl>:LEVels** <DisplayLevels>

Displays the reference levels of the indicated measurement.

**Suffix:**
<mg> 1 to 16, index of the measurement

<rl> Irrelevant, omit the suffix.
Reference level set is assigned to the measurement.

**Parameters:**
<DisplayLevels> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

---

**MEASurement<mg>:ENVSelect** <EnvelopeCurve>

Relevant only for measurements on envelope waveforms. It selects the envelope to be used for measurement.

Prerequisites:

- `ACQuire:TYPE` is set to `ENVElope`.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<EnvelopeCurve> MIN | MAX | BOTH

**MIN**
Measures on the lower envelope.

**MAX**
Measures on the upper envelope.

**BOTH**
The envelope is ignored, and the waveform is measured as usual.

*RST: BOTH

**Usage:** Asynchronous command

**Manual operation:** See "Envelope" on page 285

## 18.14.2 Settings for amplitude/time measurements

MEASurement<mg>:AMPTime:CSLope....1157
MEASurement<mg>:AMPTime:ESLope....1157
MEASurement<mg>:AMPTime:PTCount....1157
MEASurement<mg>:AMPTime:PSLope....1158
MEASurement<mg>:AMPTime:DELay<n>:DIRection....1158
MEASurement<mg>:AMPTime:DTOTrigger<n>:SLOPe....1158
MEASurement<mg>:AMPTime:DELay<n>:SLOPe....1159
MEASurement<mg>:AMPTime:PHASe:UNIT....1159
MEASurement<mg>:REFLevel<rl>:REFerence....1159

**MEASurement<mg>:AMPTime:CSLope** <SetHoldClkSlp>

Sets the edge of the clock from which the setup and hold times are measured.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<SetHoldClkSlp> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Clock slope" on page 287

**MEASurement<mg>:AMPTime:ESLope** <EdgesSlope>

Sets the edge direction to be counted: rising edges, falling edges, or both. The setting is only relevant for edge count measurement `MEASurement<mg>:MAIN` is set to `EDGecount`.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<EdgesSlope> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Edges slope" on page 286

**MEASurement<mg>:AMPTime:PTCount** <PulseCount>

Sets the number of positive pulses for the pulse train measurement. It measures the duration of N positive pulses from the rising edge of the first pulse to the falling edge of the N-th pulse.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<PulseCount> Range: 1 to 2147483647
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Pulse count" on page 285

**MEASurement<mg>:AMPTime:PSLope** <PulsesSlope>

Sets the first slope of the pulses to be counted. The setting is only relevant for pulse count and pulse train measurements (`MEASurement<mg>:MAIN` is set to `PULCnt` or `PULSetrain`.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<PulsesSlope> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Pulse slope" on page 286

**MEASurement<mg>:AMPTime:DELay<n>:DIRection** <EdgeCountDir>

Selects the direction for counting slopes for each source: from the beginning of the waveform, or from the end.

**Suffix:**
<mg> 1 to 16, index of the measurement

<n> 1..2
Selects the source number.

**Parameters:**
<EdgeCountDir> FRFI | FRLA
`FRFI` - FRom FIrst, counting starts with the first edge of the waveform.
`FRLA` - FRom LAst, counting starts with the last edge of the waveform.
*RST: FRFI

**Usage:** Asynchronous command

**Manual operation:** See "Direction" on page 288

**MEASurement<mg>:AMPTime:DTOTrigger<n>:SLOPe** <DlyTrigSlp>

Sets the edge direction to be used for delay measurement.

**Suffix:**
<mg> 1 to 16, index of the measurement

<n> 1..2
Selects the source number.

**Parameters:**
<DlyTrigSlp> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Delay to trigger measurement settings" on page 288

---

**MEASurement<mg>:AMPTime:DELay<n>:SLOPe** <Slope>

Sets the edge of each source, between which the delay is measured.

**Suffix:**
<mg> 1 to 16, index of the measurement

<n> 1..2
Selects the source number.

**Parameters:**
<Slope> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 288

---

**MEASurement<mg>:AMPTime:PHASe:UNIT** <PhaseMode>

Selects the unit of the oase: degrees or radians.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<PhaseMode> DEGRees | RADians
*RST: DEGRees

**Usage:** Asynchronous command

**Manual operation:** See "Phase unit" on page 288

---

**MEASurement<mg>:REFLevel<rl>:REFerence** <RefLevSet>

Selects the set of reference levels that is used for the measurement and for the indicated measurement source. Each source of the measurement can have its own reference level set.

**Suffix:**
<mg> 1 to 16, index of the measurement

<rl> 1..2
Measurement source, to which the reference level set is applied. For most measurements, one source is needed. For delay measurements, for example, two measurement sources are required, and each source can have its own reference levels.

**Parameters:**

<RefLevSet> Number of the reference level set. Define the reference level set before you use it.

**Example:**

```
MEASurement1:MAIN DELay
MEASurement1:SOURce C1,C2
MEASurement1:REFLevel1:REFerence 1
MEASurement1:REFLevel2:REFerence 2
```

Measurement 1 is a delay measurement on channel 1 and channel 2. C1 is the first source and uses reference level set 1 (RL1). C2 is the second source and uses reference level set 2 (RL2).

**Usage:** Asynchronous command

**Manual operation:** See "Reference levels" on page 277

### 18.14.3 Settings for histogram measurements

MEASurement<mg>:HISTogram:PROBability:LIMit....1160
MEASurement<mg>:HISTogram:PROBability:TYPE....1160

**MEASurement<mg>:HISTogram:PROBability:LIMit** <Limit>

Defines a range around the probability marker.

The setting is relevant hor histogram measurements: `MEASurement<mg>:MAIN` on page 1154 is set to `MKPositive` or `MKNegative`.

**Suffix:**

<mg> 1 to 16, index of the measurement

**Parameters:**

<Limit>

Range: 0 to 100
Increment: 10
*RST: 10
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Probability limit" on page 292

**MEASurement<mg>:HISTogram:PROBability:TYPE** <Marker>

Defines the marker position in the probability domain.

The setting is relevant hor histogram measurements: `MEASurement<mg>:MAIN` on page 1154 is set to `MKPositive` or `MKNegative`.

**Suffix:**

<mg> 1 to 16, index of the measurement

**Parameters:**

<Marker> PEAK | UPPK | LWPK | MAXimum | MINimum | MEDian | MEAN

**PEAK**
The y-value with the maximum sample value in the histogram

**UPPK**
Upper peak, the y-value at the maximum sample value in the upper half of the histogram

**LWPK**
Lower peak: the y-value at the maximum sample value in the lower half of the histogram

**MAXimum**
The highest y-value with a probability > 0

**MINimum**
The lowest y-value with a probability > 0

**MEDian**
The lowest y-value with a probability > 0

**MEAN**
The weighted arithmetic average of the histogram

*RST: PEAK

**Usage:** Asynchronous command

**Manual operation:** See "Marker" on page 292

### 18.14.4 Settings for jitter measurements

Jitter measurements require option R&S MXO3-K12.

MEASurement<mg>:JITTer:NCYCles ..... 1161
MEASurement<mg>:JITTer:POLarity ..... 1162
MEASurement<mg>:JITTer:REFLevel<rl> ..... 1162
MEASurement<mg>:JITTer:RELPolarity ..... 1163
MEASurement<mg>:JITTer:SLOPe ..... 1163
MEASurement<mg>:JITTer:TREF ..... 1164
MEASurement<mg>:JITTer:UNIT ..... 1164

---

**MEASurement<mg>:JITTer:NCYCles** <NumberCycles>

Sets the number of periods (cycles) that are accumulated to measure the N-cycle jitter.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<NumberCycles> Range: 1 to 4294967295
Increment: 1
*RST: 2

**Usage:** Asynchronous command

**Manual operation:** See "Number of cycles" on page 296

**MEASurement<mg>:JITTer:POLarity** <Polarity>

For cycle-cycle width and the cycle-cycle duty cycle measurements, the command sets the polarity of pulses for which the pulse width is measured: POSitive or NEGative. `MEASurement<mg>:MAIN` is set to measurements CCWidth | CCDutycycle.

For skew delay and skew phase measurements, the command sets the edge of the first waveform from which the measurements starts: POSitive, NEGative or EITHer. `MEASurement<mg>:MAIN` is set to measurements SKWDelay | SKWPhase.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<Polarity> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Pulse polarity" on page 295

**MEASurement<mg>:JITTer:REFLevel<rl>** <ReferenceLevel>

Selects the reference level that is used for the measurement and for the indicated measurement source. Each source of the measurement can have its own reference level .

**Suffix:**
<mg> 1 to 16, index of the measurement

<rl> 1..2
Measurement source, to which the reference level set is applied. For most measurements, one source is needed. For delay measurements, for example, two measurement sources are required, and each source can have its own reference levels.

**Parameters:**
<ReferenceLevel> UPPer | MIDDle | LOWer
*RST: MIDDle

**Example:**
```
MEASurement1:MAIN SKWDelay
MEASurement1:SOURce C1,C2
MEASurement1:JITTer:REFLevel1 MIDDle
MEASurement1:JITTer:REFLevel2 UPPer
```

Measurement 1 is a skew delay measurement on channel 1 and channel 2. C1 is the first source and uses the middle reference level. C2 is the second source and uses the upper reference level.

**Usage:** Asynchronous command

**Manual operation:** See "Reference level" on page 296

**MEASurement<mg>:JITTer:RELPolarity** <RelPolarity>

Sets the edge of the second waveform relative to the first waveform.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<RelPolarity> MATChing | INVerse

**MATChing**
Measures from positive to positive edge or from negative to negative edge.

**INVerse**
Measures from positive to negative edge or from negative to positive edge.

*RST: MATChing

**Usage:** Asynchronous command

**Manual operation:** See "Relative polarity" on page 296

**MEASurement<mg>:JITTer:SLOPe** <Slope>

For cycle-cycle jitter, N-cycle jitter, and cycle-cycle duty cycle measurements, the setting selects the slope at which the periods and thus the jitter is measured: FIRSt, POSitive, NEGative or EITHer. `MEASurement<mg>:MAIN` is set to measurements CCJitter | NCJitter | CCDutycycle.

For time-interval error measurements, the command sets the edges of the data signal that are used for measurements: POSitive, NEGative or EITHer. `MEASurement<mg>:MAIN` is set to TIE.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<Slope> FIRSt | POSitive | NEGative | EITHer

*RST: FIRSt

<Slope> FIRSt | POSitive | NEGative | EITHer

**FIRSt**
Measures the period from the first edge that is found, no matter of its direction.

**POSitive**
Measures the period at positive going edges.

**NEGative**
Measures the period at negative going edges.

**EITHer**
Measures the period at both positive and negative going edges. This option is useful, for example, to check the clock stability of a double data rate clock.

*RST: FIRSt

**Usage:** Asynchronous command

**Manual operation:** See "Cycle begin, Slope" on page 295

---

**MEASurement<mg>:JITTer:TREF** <TimingReference>

Selects the timing reference, which is one of the available clock configurations. The timing reference must be defined before it can be used.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<TimingReference> Range: 1 to 4

**Usage:** Asynchronous command

**Manual operation:** See "Timing reference" on page 298

---

**MEASurement<mg>:JITTer:UNIT** <Unit>

Sets the unit for data rate measurements.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<Unit> BIT_S | HZ

*RST: BIT_S

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 298

## 18.14.5 Settings for protocol measurements

Protocol measurements require option R&S MXO3-K500, and at least one serial protocol option.

MEASurement<mg>:PROTocol:FNAMe....1164
MEASurement<mg>:PROTocol:FDName....1165
MEASurement<mg>:PROTocol:FDValue....1165
MEASurement<mg>:PROTocol:F2Name....1165
MEASurement<mg>:PROTocol:FD2Name....1166
MEASurement<mg>:PROTocol:FD2Value....1166

---

**MEASurement<mg>:PROTocol:FNAMe** <FrameName>

Sets or queries the name of the frame or the frame type, at which the oscilloscope executes or starts the measurement.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<FrameName>

**Usage:** Asynchronous command

**Manual operation:** See "Frame Type" on page 301

---

**MEASurement<mg>:PROTocol:FDName** <FieldName>

Sets or queries the name of the field or the field type, at which the oscilloscope executes or starts the measurement.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<FieldName>

**Usage:** Asynchronous command

**Manual operation:** See "Field Type" on page 301

---

**MEASurement<mg>:PROTocol:FDValue** <FieldValue>

Sets or queries the one or more values of the field, at which the oscilloscope executes or starts the measurement.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<FieldValue> List of comma separated values

**Usage:** Asynchronous command

**Manual operation:** See "Field Value" on page 301

---

**MEASurement<mg>:PROTocol:F2Name** <Frame2Name>

Sets or queries the name of the frame or the frame type, at which the oscilloscope ends the measurement in a "From" - "To" condition.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<Frame2Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame Type" on page 301

**MEASurement<mg>:PROTocol:FD2Name** <Field2Name>

Sets or queries the name of the field or the field type, at which the oscilloscope ends the measurement in a "From" - "To" condition.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<Field2Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field Type" on page 301

**MEASurement<mg>:PROTocol:FD2Value** <Field2Value>

Sets or queries the one or more values of the field, at which the oscilloscope ends the measurement in a "From" - "To" condition.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<Field2Value> List of comma separated values

**Usage:** Asynchronous command

**Manual operation:** See "Field Value" on page 302

### 18.14.6 Results

MEASurement<mg>:RESult[:ACTual]? 1166
MEASurement<mg>:RESult:AVG? 1166
MEASurement<mg>:RESult:EVTCount? 1166
MEASurement<mg>:RESult:NPEak? 1166
MEASurement<mg>:RESult:PPEak? 1167
MEASurement<mg>:RESult:RMS? 1167
MEASurement<mg>:RESult:WFMCount? 1167
MEASurement<mg>:RESult:RSTDdev? 1167
MEASurement<mg>:RESult:STDDev? 1167
MEASurement<mg>:RESult:STARt? 1167
MEASurement<mg>:RESult:STOP? 1167
MEASurement<mg>:RESult:EVENts:COUNt? 1167
MEASurement<mg>:RESult:EVENts:STARt? 1168
MEASurement<mg>:RESult:EVENts:STOP? 1168
MEASurement<mg>:RESult:EVENts:VALue? 1168

**MEASurement<mg>:RESult[:ACTual]?**
**MEASurement<mg>:RESult:AVG?**
**MEASurement<mg>:RESult:EVTCount?**
**MEASurement<mg>:RESult:NPEak?**

**MEASurement<mg>:RESult:PPEak?**
**MEASurement<mg>:RESult:RMS?**
**MEASurement<mg>:RESult:WFMCount?**
**MEASurement<mg>:RESult:RSTDdev?**
**MEASurement<mg>:RESult:STDDev?**

Return the statistic results of the specified measurement. The measurement type is defined with `MEASurement<mg>:MAIN`.

- [:ACTual]: current measurement result
- AVG: arithmetic mean of the measurement results
- EVTCount: number of measurement results in the measurement
- NPEak: negative peak value of the measurement results
- PPEak: positive peak value of the measurement results
- RMS: RMS value of the measurement results
- STDDev: standard deviation of the measurement results
- RSTDdev: relative standard deviation (*STDDev / AVG*)
- WFMCount: number of measured waveforms

**Suffix:**
<mg> 1 to 16, index of the measurement

**Return values:**
<Result> Numeric result value

**Usage:** Query only
Asynchronous command

---

**MEASurement<mg>:RESult:STARt?**
**MEASurement<mg>:RESult:STOP?**

Return the start and stop times of the specified measurement. The parameter defines the measurement. If no parameter is specified, the result of the main measurement is returned. The main measurement is defined using `MEASurement<mg>:MAIN`.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Return values:**
<Stop> Range: -1E+26 to 1E+26
Increment: 0
*RST: 0

**Usage:** Query only
Asynchronous command

---

**MEASurement<mg>:RESult:EVENts:COUNt?**

Returns the number of measured events in one acquisition.

The command is relevant for measurements of all events, see `MEASurement<mg>:MULTiple`.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Return values:**
<Count> Number of events

**Usage:** Query only
Asynchronous command

---

**MEASurement<mg>:RESult:EVENts:STARt?** <EventIndex>

Returns the start time of the indicated measured event.

The command is relevant for measurements of all events, see `MEASurement<mg>:MULTiple`.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Setting parameters:**
<EventIndex> Index number of the measured event.

**Return values:**
<EventStart> Range: -1E+26 to 1E+26
Increment: 0
*RST: 0

**Usage:** Query only
Asynchronous command

---

**MEASurement<mg>:RESult:EVENts:STOP?** <EventIndex>

Returns the end time of the indicated measured event.

The command is relevant for measurements of all events, see `MEASurement<mg>:MULTiple`.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Setting parameters:**
<EventIndex> Index number of the measured event.

**Return values:**
<EventStop> Range: -1E+26 to 1E+26
Increment: 0
*RST: 0

**Usage:** Query only
Asynchronous command

---

**MEASurement<mg>:RESult:EVENts:VALue?** <MeasResEvtIdx>

Returns the measured value of the indicated measured event.

The command is relevant for measurements of all events, see MEASurement<mg>:MULTiple.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Setting parameters:**
<EventIndex> Index number of the measured event.

**Return values:**
<EventValue> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

## 18.14.7 Statistics

MEASurement<mg>:STATistics:ARESet....1169
MEASurement<mg>:STATistics[:ENABle]....1169
MEASurement<mg>:MULTiple....1170
MEASurement<mg>:MNOMeas....1170

**MEASurement<mg>:STATistics:ARESet**

Resets the statistics for all measurements.

**Suffix:**
<mg> Irrelevant, omit the suffix.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Clear results" on page 305

**MEASurement<mg>:STATistics[:ENABle]** <GlobalEnable>

Enables statistics calculation for all measurements.

**Suffix:**
<mg> Irrelevant, omit the suffix.

**Parameters:**
<GlobalEnable> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Statistics" on page 305

**MEASurement<mg>:MULTiple** <GlobalMeassAll>

If ON, the measurement is performed repeatedly if the measured parameter occurs several times inside the acquisition or defined gate. All results are included in evaluation, e.g. in statistics. To set the number of results to be considered, use `MEASurement<mg>:MNOMeas`.

**Suffix:**
<mg> Irrelevant, omit the suffix.

**Parameters:**
<GlobalMeassAll> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Measure all events" on page 305

**MEASurement<mg>:MNOMeas** <MaxMeasPerAcq>

Sets the maximum number of measurements per acquisition if `MEASurement<mg>:MULTiple` is on. The setting affects all measurements.

**Suffix:**
<mg> Irrelevant, omit the suffix.

**Parameters:**
<MaxMeasPerAcq> Range: 2 to 1E6
Increment: 1
*RST: 1E6

**Usage:** Asynchronous command

**Manual operation:** See "Max. No. of events" on page 305

### 18.14.8 Tracks

MEASurement<mg>:TRACk[:STATe] ..... 1170
MEASurement<mg>:TRACk:AUTO ..... 1171
MEASurement<mg>:TRACk:CONTiunous ..... 1171
MEASurement<mg>:TRACk:OFFSet ..... 1171
MEASurement<mg>:TRACk:SCALe ..... 1172
MEASurement<mg>:TRACk:DATA:HEADer? ..... 1172
MEASurement<mg>:TRACk:DATA[:VALues]? ..... 1173

**MEASurement<mg>:TRACk[:STATe]** <State>

Enables or disables the track for the selected measurement.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 307

---

**MEASurement<mg>:TRACk:AUTO**

Sets the vertical scale and the offset of the track, so that the complete height of the diagram is used.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Usage:** Event
Asynchronous command

**Manual operation:** See "Fit to screen" on page 307

---

**MEASurement<mg>:TRACk:CONTiunous** <AutoScale>

Performs an automatic scaling whenever the track does not fit in the diagram during the measurement period.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<AutoScale> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Auto scale" on page 307

---

**MEASurement<mg>:TRACk:OFFSet** <VerticalOffset>

Sets or queries the offset of the track waveform.

If `MEASurement<mg>:TRACk:CONTiunous` is ON, use the command to query the current value.

If `MEASurement<mg>:TRACk:CONTiunous` is OFF, the command sets the offset.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<VerticalOffset> Range: -1E+26 to 1E+26
Increment: 1E-06
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Meas offset" on page 307

---

**MEASurement<mg>:TRACk:SCALe** <VerticalScale>

Sets or queries the vertical scale of the track diagram.

If `MEASurement<mg>:TRACk:CONTiunous` is ON, use the command to query the current value.

If `MEASurement<mg>:TRACk:CONTiunous` is OFF, the command sets the scale.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<VerticalScale>
Range: 1E-26 to 1E+26
Increment: 1E-05
*RST: 0.5
Default unit: V/div

**Usage:** Asynchronous command

**Manual operation:** See "Meas scale" on page 307

---

**MEASurement<mg>:TRACk:DATA:HEADer?**

Returns the header of track data, the attributes of the waveform.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Return values:**
<XStart> 1. header value: time of the first track value in s

<XStop> 2. header value: time of the last track value in s

<RecordLength> 3. header value: number of measured events = number of points in the track waveform

<ValuesPerSample> 4. header value: number of values per sample. For tracks, the result is always 1.

**Example:**
```
MEAS1:TRAC:DATA:HEAD?
-1E-07,9.980000000000001E-08,1000,1
```
Start time of the data is -1E-07 = -100 ns, and end time of the data is 9.980000000000001E-08= 99.8 ns. The data stream has 1000 values with one value per sample, inn other words, the track waveform has 1000 points.

**Usage:** Query only
Asynchronous command

**MEASurement<mg>:TRACk:DATA[:VALues]?** [<Offset>[,<Length>]]

Returns the data of the indicated track waveform for transmission from the instrument to the controlling computer. The data can be used in MATLAB, for example.

Without parameters, the complete waveform is retrieved. Using the offset and length parameters, data can be retrieved in smaller portions, which makes the command faster. If you send only one parameter, it is interpreted as offset, and the data is retrieved from offset to the end of the waveform.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Query parameters:**
<Offset> Number of offset track points, which are ignored in the data transfer.

[<Length>] Number of track points to be retrieved.

**Return values:**
<Data> List of values according to the format and content settings.

**Usage:** Query only
Asynchronous command

## 18.14.9 Histograms on measurement results

To transfer the histogram data to a remote computer, use:

- `EXPort:HMEasurement<m>:DATA:HEADer?` on page 1149
- `EXPort:HMEasurement<m>:DATA[:VALues]?` on page 1150

For export of histogram data to file, see Section 18.13.5, "Histogram data: export to file", on page 1132.

MEASurement<mg>:STATistics:HISTogram:AUToscale ..... 1173
MEASurement<mg>:STATistics:HISTogram[:ENABle] ..... 1174
MEASurement<mg>:STATistics:HISTogram:HBINs ..... 1174
MEASurement<mg>:STATistics:HISTogram:HORizontal:OFFSet ..... 1174
MEASurement<mg>:STATistics:HISTogram:HORizontal:SCALe ..... 1175
MEASurement<mg>:STATistics:HISTogram:RESet ..... 1175

**MEASurement<mg>:STATistics:HISTogram:AUToscale** <AutoScaleMode>

If ON, the range on the x-axis is continuously adjusted to the measured values. If OFF, you can set the offset and the scale of the histogram with `MEASurement<mg>:STATistics:HISTogram:HORizontal:OFFSet` and `MEASurement<mg>:STATistics:HISTogram:HORizontal:SCALe`.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<AutoScaleMode> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Auto scale" on page 312

---

**MEASurement<mg>:STATistics:HISTogram[:ENABle]** <State>

Enables the histogram for the specified measurement.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 312

---

**MEASurement<mg>:STATistics:HISTogram:HBINs** <NrBinsUser>

Sets the number of bins to be displayed in the histogram of the specified measurement.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<NrBinsUser> Range: Possible values: 10, 25, 50, 100, 250, 500, 1000, 2000
*RST: 250

**Usage:** Asynchronous command

**Manual operation:** See "Number of bins" on page 312

---

**MEASurement<mg>:STATistics:HISTogram:HORizontal:OFFSet** <Center>

Sets the center value of the histogram on the x-axis, which is the measurement result in the middle of the histogram.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<Center> Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0
Default unit: Specify the unit with the value.

**Example:** `MEASurement:STATistics:HISTogram:HORizontal:OFFSet 3mV`

**Usage:** Asynchronous command

**Manual operation:** See "Offset" on page 312

---

**MEASurement<mg>:STATistics:HISTogram:HORizontal:SCALe** <Scale>

Sets the horizontal scale of the histogram.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Parameters:**
<Scale>
Range: 0 to INF
Increment: 0.1
*RST: 1
Default unit: Specify the unit with the value.

**Usage:** Asynchronous command

**Manual operation:** See "Scale" on page 313

---

**MEASurement<mg>:STATistics:HISTogram:RESet**

Resets the cumulation of measurement results, all bins are set to zero.

**Suffix:**
<mg> 1 to 16, index of the measurement

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Reset" on page 312

## 18.15 Cursor measurements

Some of the commands in the following section are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, section "Command sequence and synchronization"

- Cursor setup....1176
- Cursor results....1186
- Peak search using cursors....1187
- Cursor appearance....1189

## 18.15.1 Cursor setup

CURSor<cu>:AOFF....1176
CURSor<cu>:COUNt?....1176
CURSor<cu>:STATe....1177
CURSor<cu>:FUNCtion....1177
CURSor<cu>:SMODe....1177
CURSor<cu>:SOURce....1178
CURSor<cu>:USSource....1180
CURSor<cu>:SSOurce....1180
CURSor<cu>:X1Position....1180
CURSor<cu>:X2Position....1181
CURSor<cu>:Y1Position....1181
CURSor<cu>:Y2Position....1182
CURSor<cu>:SSCReen....1182
CURSor<cu>:TRACking[:STATe]....1183
CURSor<cu>:LABel....1183
CURSor<cu>:HORizontal<n>:LABel....1183
CURSor<cu>:VERTical<n>:LABel....1184
CURSor<cu>:DISPlay:VALues....1184
CURSor<cu>:SIAD....1184
CURSor<cu>:XCOupling....1185
CURSor<cu>:YCOupling....1185
CURSor<cu>:X1ENvelope....1185
CURSor<cu>:X2ENvelope....1185

**CURSor<cu>:AOFF**

Switches all cursors off.

**Suffix:**
<cu> Irrelevant, omit the suffix.

**Usage:** Setting only
Asynchronous command

**CURSor<cu>:COUNt?**

Returns the maximum number of cursor sets, which is the maximum value for the cursor suffix.

**Suffix:**
<cu> Irrelevant, omit the suffix.

**Return values:**
<Count> Maximum number of cursor sets

**Usage:** Query only
Asynchronous command

---

**CURSor<cu>:STATe** <State>

Enables the selected cursor measurement.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show cursor" on page 264

---

**CURSor<cu>:FUNCtion** <Type>

Defines the cursor type to be used for the measurement.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<Type> VERTical | HORizontal | PAIRed

**HORizontal**
A pair of horizontal cursor lines. Not available for measurements on multiple sources.

**VERTical**
A pair of vertical cursor lines. Not available for measurements on multiple sources.

**PAIRed**
Both vertical and horizontal cursor line pairs.

*RST: PAIRed

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 265

---

**CURSor<cu>:SMODe** <SourceMode>

Selects the number of sources that you want to measure.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<SourceMode> SINGle | SECond | MSOurce

**SINGle**
The cursor lines are set on one waveform. Select the source with `CURSor<cu>:SOURce`.

**SECond**
The cursor lines are set on two waveforms. To set the second souce, use `CURSor<cu>:SSOurce`.

**MSOurce**
Multiple waveforms are selected to be measured with one cursor set. Activates `CURSor<cu>:TRACking[:STATe]` and sets `CURSor<cu>:FUNCtion` to PAIRed. Select the sources with `CURSor<cu>:SOURce`.

*RST: SINGle

**Usage:** Asynchronous command

**Manual operation:** See "Source mode" on page 265

---

**CURSor<cu>:SOURce** <Signal>

Selects the cursor source or multiple sources, which are the waveforms to be measured. The query returns the waveforms sorted by category and number.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**

<Signal> C1,C2,...,C8,D0,D1,D2,...,D15,M1,M2,...,M8,R1,R2,...,R8,SPECMAXH1,SPECMINH1,SPECNORM1,SPECAVER1,SPECMAXH2,SPECMINH2,SPECNORM2,SPECAVER2,SPECMAXH3,SPECMINH3,SPECNORM3,SPECAVER3,SPECMAXH4,SPECMINH4,SPECNORM4,SPECAVER4,
TRK1,TRK2,TRK3, ...,TRK16,PBUS1,PBUS2,PBUS3,PBUS4,XY1,XY2,XY3,XY4 | PA1QPOWER,PA2QPOWER,PA3QPOWER,PA1HPOWER1,PA2HPOWER1,PA3HPOWER1,PA1SPOWER,PA2SPOWER,PA3SPOWER,PA1SOA,PA2SOA,PA3SOA,PA1IPOWER,PA2IPOWER,PA3IPOWER,PA1OPOWER1,PA2OPOWER1,PA3OPOWER1,PA1OPOWER2,PA2OPOWER2,PA3OPOWER2,PA1OPOWER3,PA2OPOWER3,PA3OPOWER3,PTHREE1HPOWER1,PTHREE2HPOWER1,PTHREE3HPOWER1,
PTHREE1HPOWER2,PTHREE2HPOWER2,PTHREE3HPOWER2, PTHREE1HPOWER3,PTHREE2HPOWER3,PTHREE3HPOWER3,
PTHREE1HTPOWER,PTHREE2HTPOWER,PTHREE3HTPOWER, PTHREE1QPOWER1,PTHREE2QPOWER1,PTHREE3QPOWER1, PTHREE1QPOWER2,PTHREE2QPOWER2,PTHREE3QPOWER2,
PTHREE1QPOWER3,PTHREE2QPOWER3,PTHREE3QPOWER3,
PTHREE1QTPOWER,PTHREE2QTPOWER,PTHREE3QTPOWER,
PTHREE1EIPOWER1,PTHREE2EIPOWER1,...,PTHREE6EIPOWER1,
PTHREE1EIPOWER2,PTHREE2EIPOWER2,...,PTHREE6EIPOWER2,
PTHREE1EIPOWER3,PTHREE2EIPOWER3,...,PTHREE6EIPOWER3, PTHREE1EITPOWER,PTHREE2EITPOWER,...,PTHREE6EITPOWER,PTHREE1EOPOWER1,PTHREE2EOPOWER1,...,PTHREE6EOPOWER1,
PTHREE1EOPOWER2,PTHREE2EOPOWER2,...,PTHREE6EOPOWER2,
PTHREE1EOPOWER3,PTHREE2EOPOWER3,...,PTHREE6EOPOWER3, PTHREE1EOTPOWER,PTHREE2EOTPOWER,...,PTHREE6EOTPOWER | PA1TOPOWER,PA2TOPOWER,PA3TOPOWER

Comma-separated list of waveforms.

**Example:** `CURSor:SOURce C1`

Sets channel 1 as cursor source and changes `CURSor<cu>:SMODe` to SINGle if MSOurce was set before.

**Example:** `CURSor:SOURce C1,M1,C3`
`CURSor:SOURce? --> C1,C3,M1`
Sets channel 1, 3 and math 1 as cursor sources and changes `CURSor<cu>:SMODe` to MSOurce if SINGle was set before.
Activates `CURSor<cu>:TRACking[:STATe]` for the specified cursor set.

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 265

---

**CURSor<cu>:USSource** <UseSource2>

Enables the second cursor source. To select the second source, use `CURSor<cu>:SSOurce`.

The command has the same effect as `CURSor<cu>:SMODe SECond`.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<UseSource2> OFF | ON

**Usage:** Asynchronous command

---

**CURSor<cu>:SSOurce** <Source2>

Selects the second cursor source.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<Source2> One of the possible cursor sources, see `CURSor<cu>:SOURce`.

**Usage:** Asynchronous command

**Manual operation:** See "Source 2" on page 265

---

**CURSor<cu>:X1Position** <X1Position>

Defines the position of the left vertical cursor line.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<X1Position>
Range: -1E+26 to 1E+26
Increment: 0.1
*RST: depends on time scale, at 25% of the time axis
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "X1 position, X2 position" on page 266

---

**CURSor<cu>:X2Position** <X2Position>

Defines the position of the right vertical cursor line.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<X2Position>
Range: -1E+26 to 1E+26
Increment: 0.1
*RST: depends on time scale, at 75% of the time axis
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "X1 position, X2 position" on page 266

---

**CURSor<cu>:Y1Position** [<Signal>]

The command usage depends on the setting of `CURSor<cu>:SMODe`.

In single source and second source mode, the command sets or queries the position of the lower horizontal cursor line. The <Signal> parameter is irrelevant. If `CURSor<cu>:TRACking[:STATe]` is enabled, the Y-positions are set automatically, and the query returns the measurement result.

In multiple source mode, tracking is always on. The command is used as query only, and you specify the signal for which you want to get the value.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<Value> Y-position of the first horizontal cursor line.

**Query parameters:**
<Signal> Only relevant for `CURSor<cu>:SMODe MSource`. One of the sources that are specified with `CURSor<cu>:SOURce`.

**Example:**
```
CURSor2:SMODe SINGle
CURSor2:FUNCtion PAIRed
CURSor2:TRACking ON
CURSor2:Y1Position?
--> 0.123859
```

In single source mode with active tracking, the command returns the Y-value.

**Example:**
```
CURSor:SMODe SINGle
CURSor:FUNCtion PAIRed
CURSor:TRACking OFF
CURSor:Y1Position 0.15
```
In single source mode with disabled tracking, the command sets the Y-value.

**Example:**
```
CURSor2:SMODe MSOurce
CURSor2:Y1Position? C1
--> 0.123859
```
In multiple source mode, tracking is always active. The command returns the Y-value for the specified source.

**Usage:** Asynchronous command

**Manual operation:** See "Y1 position, Y2 position" on page 266

---

**CURSor<cu>:Y2Position** [<Signal>]

Defines or queries the position of the upper horizontal cursor line.The command works in the same way as `CURSor<cu>:Y1Position`.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<Value> Y-position of the second horizontal cursor line.

**Query parameters:**
<Signal> Only relevant for `CURSor<cu>:SMODe MSource`. One of the sources that are specified with `CURSor<cu>:SOURce`.

**Usage:** Asynchronous command

**Manual operation:** See "Y1 position, Y2 position" on page 266

---

**CURSor<cu>:SSCReen**

Resets the cursors to their initial positions. Reset is helpful if the cursors have disappeared from the display or need to be moved for a larger distance.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Place on display" on page 267

**CURSor<cu>:TRACking[:STATe]** <TrackCurve>

If enabled, the horizontal cursor lines follow the waveform. The function is available if both horizontal and vertical cursors are displayed (`CURSor<cu>:FUNCtion PAIRed`).

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<TrackCurve> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Track waveform" on page 266

**CURSor<cu>:LABel** <ShowLabel>

Shows the cursor labels in the diagram.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<ShowLabel> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show label" on page 268

**CURSor<cu>:HORizontal<n>:LABel** <Label>

Defines the label to be displayed with the horizontal cursor lines. By default, the cursors are labeled as Cu1.1, Cu1.2, Cu2.1, ...

**Suffix:**
<cu> 1 to 4, index of the cursor set

<n> 1..2
Number of the cursor line

**Parameters:**
<Label> String with the cursor label

**Example:**
```
CURSor1:HORizontal1:LABel 'Cu1H1'
CURSor1:HORizontal2:LABel "Cu1H2"
```

**Usage:** Asynchronous command

**Manual operation:** See "Horizontal cursor 1, Horizontal cursor 2" on page 268

**CURSor<cu>:VERTical<n>:LABel** <Label>

Defines the label to be displayed with the vertical cursor lines. By default, the cursors are labeled as Cu1.1, Cu1.2, Cu2.1, ...

**Suffix:**
<cu> 1 to 4, index of the cursor set

<n> 1..2
Number of the cursor line

**Parameters:**
<Label> String with the cursor label

**Example:**
```
CURSor2:VERTical1:LABel 'Cu2V1'
CURSor2:VERTical2:LABel "Cu2V2"
```

**Usage:** Asynchronous command

**Manual operation:** See "Vertical cursor 1, Vertical cursor 2" on page 268

**CURSor<cu>:DISPlay:VALues** <AddValuesLabel>

Shows the measured values in the cursor labels.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<AddValuesLabel> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show values" on page 268

**CURSor<cu>:SIAD** <ShwInAllDiags>

Shows the enabled cursor measurements in all active diagrams of the time domain.

In the spectrum domain, the setting is disabled. The cursors are shown only on the source spectrum of the measurement.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<ShwInAllDiags> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Show in all diagrams" on page 267

**CURSor<cu>:XCOupling** <Coupling>

Defines the positioning mode of the vertical cursor.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<Coupling> OFF | ON

**ON**
Moving one cursor line moves the other cursor line too. The cursor lines always remain a fixed distance.

**OFF**
Each cursor line is positioned independently.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "X, Y" on page 267

**CURSor<cu>:YCOupling** <Coupling>

Defines the positioning mode of the horizontal cursor. If the horizontal cursor lines track the waveform, the y coupling is irrelevant (`CURSor<cu>:TRACking[:STATe]` is `ON`).

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<Coupling> OFF | ON

**ON**
Moving one cursor line moves the other cursor line too. The cursor lines always remain a fixed distance.

**OFF**
Each cursor line is positioned independently.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "X, Y" on page 267

**CURSor<cu>:X1ENvelope** <EnvelopeSelect>
**CURSor<cu>:X2ENvelope** <EnvelopeSelect>

Define which horizontal cursor is positioned to the maximum and which to the minimum envelope values.

Prerequisites:

- `ACQuire:TYPE` is set to `ENVElope` or `PDETect`.
- `CURSor<cu>:TRACking[:STATe]` is set to `ON`.
- `CURSor<cu>:FUNCtion` is set to `PAIRed`.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<EnvelopeSelect> MIN | MAX

**MIN**
The horizontal cursor is set to the crossing point of the vertical cursor with the minimum waveform envelope.

**MAX**
The horizontal cursor is set to the crossing point of the vertical cursor with the maximum waveform envelope.

*RST: MAX

**Usage:** Asynchronous command

**Manual operation:** See "Envelope 1, Envelope 2" on page 266

## 18.15.2 Cursor results

CURSor<cu>:XDELta[:VALue]? ..... 1186
CURSor<cu>:XDELta:INVerse? ..... 1186
CURSor<cu>:YDELta[:VALue]? ..... 1187
CURSor<cu>:YDELta:SLOPe? ..... 1187

**CURSor<cu>:XDELta[:VALue]?**

Queries the delta value (distance) of two vertical cursor lines.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Return values:**
<Delta> Range: -1E+26 to 1E+26
Increment: 0.1
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

**CURSor<cu>:XDELta:INVerse?**

Queries the inverse value of the delta value (distance) of the two vertical cursor lines.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Return values:**
<DeltaInverse> Range: -1E+26 to 1E+26
Increment: 0.1
*RST: 0
Default unit: Hz

**Usage:** Query only
Asynchronous command

---

**CURSor<cu>:YDELta[:VALue]?** [<SignalValue>]

Returns the delta value (distance) of the two horizontal cursor lines.

In multiple source mode, specify the signal for which you want to get the value.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Query parameters:**
<Signal> Only relevant for `CURSor<cu>:SMODe MSource`. One of the sources that are specified with `CURSor<cu>:SOURce`.

**Return values:**
<Value> Inverse value of the delta value of the two horizontal cursor lines.

**Usage:** Query only
Asynchronous command

---

**CURSor<cu>:YDELta:SLOPe?** [<SignalValue>]

Returns the inverse value of the voltage difference, the reciprocal of the vertical distance of two horizontal cursor lines: *1/ΔV*.

In multiple source mode, specify the signal for which you want to get the value.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Query parameters:**
<Signal> Only relevant for `CURSor<cu>:SMODe MSource`. One of the sources that are specified with `CURSor<cu>:SOURce`.

**Return values:**
<Value> Inverse value of the delta value of the two cursor lines.

**Usage:** Query only
Asynchronous command

### 18.15.3 Peak search using cursors

CURSor<cu>:FFT:SETCenter ..... 1188
CURSor<cu>:FFT:TOCenter ..... 1188
CURSor<cu>:MAXimum[:PEAK] ..... 1188
CURSor<cu>:MAXimum:LEFT ..... 1188
CURSor<cu>:MAXimum:RIGHt ..... 1188
CURSor<cu>:MAXimum:NEXT ..... 1189
CURSor<cu>:PEXCursion ..... 1189

**CURSor<cu>:FFT:SETCenter**

Sets the center frequency to the frequency value that is measured at cursor line Cu1.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Set center frequency to" on page 269

**CURSor<cu>:FFT:TOCenter**

Sets the vertical cursor line Cu1 to the center frequency.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Center frequency" on page 269

**CURSor<cu>:MAXimum[:PEAK]**

Sets both cursors to the absolute peak value.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Usage:** Event
Asynchronous command

**Manual operation:** See "Absolute peak" on page 269

**CURSor<cu>:MAXimum:LEFT**

Cursor 2 is set to the next peak to the left of the current position.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Usage:** Event
Asynchronous command

**Manual operation:** See "Next peak left" on page 269

**CURSor<cu>:MAXimum:RIGHt**

Cursor 2 is set to the next peak to the right of the current position.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Usage:** Event
Asynchronous command

**Manual operation:** See "Next peak right" on page 269

---

**CURSor<cu>:MAXimum:NEXT**

Cursor 2 is set to the next smaller absolute peak from the current position.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Usage:** Event
Asynchronous command

**Manual operation:** See "Next peak" on page 269

---

**CURSor<cu>:PEXCursion** <Value>

Sets the minimum level by which the waveform must rise or fall so that it will be identified as a maximum or a minimum by the search functions.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<Value>

**Usage:** Asynchronous command

**Manual operation:** See "Peak excursion" on page 269

### 18.15.4 Cursor appearance

---

**CURSor<cu>:STYLe** <Style>

Defines how the cursor is displayed in the diagram.

**Suffix:**
<cu> 1 to 4, index of the cursor set

**Parameters:**
<Style> LINes | LRHombus | VLRHombus | RHOMbus

**LINes**
The cursors are displayed as lines.

**LRHombus**
The cursors are displayed as lines. The intersections of the cursors with the waveforms are displayed by rhombus-shaped points.

**VLRHombus**
The cursors are displayed only as vertical lines. The intersections of the cursors with the waveforms are displayed by rhombus-shaped points.

**RHOMbus**
The intersections of the cursors with the waveforms are displayed by rhombus-shaped points.

*RST: LINes

**Usage:** Asynchronous command

**Manual operation:** See "Cursor style" on page 264

# 18.16 Spectrum analysis

## 18.16.1 Spectrum setup

CALCulate:SPECtrum<sp>:COUPling[:ENABle] 1191
CALCulate:SPECtrum<sp>:COUPling:FROM 1191
CALCulate:SPECtrum<sp>:COUPling:MODE 1191
CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution]:ADJusted? 1192
CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution]:AUTO 1192
CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution]:RATio 1192
CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution][:VALue] 1193
CALCulate:SPECtrum<sp>:FREQuency:CENTer 1193
CALCulate:SPECtrum<sp>:FREQuency:SCALe 1193
CALCulate:SPECtrum<sp>:FREQuency:SPAN 1194
CALCulate:SPECtrum<sp>:FREQuency:STARt 1194
CALCulate:SPECtrum<sp>:FREQuency:STOP 1194
CALCulate:SPECtrum<sp>:FREQuency:WINDow:TYPE 1195
CALCulate:SPECtrum<sp>:MAGNitude:LEVel 1196
CALCulate:SPECtrum<sp>:MAGNitude:RANGe 1196
CALCulate:SPECtrum<sp>:MAGNitude:SCALe 1196
CALCulate:SPECtrum<sp>:PRESet 1196
CALCulate:SPECtrum<sp>:SOURce 1197
CALCulate:SPECtrum<sp>:STATe 1198
CALCulate:SPECtrum<sp>:THReshold 1198
CALCulate:SPECtrum<sp>:PEXCursion 1198
CALCulate:SPECtrum<sp>:WAVeform:AVERage:COUNt 1199
CALCulate:SPECtrum<sp>:WAVeform:AVERage:ENABle 1199
CALCulate:SPECtrum<sp>:WAVeform:MAXimum:ENABle 1199
CALCulate:SPECtrum<sp>:WAVeform:MINimum:ENABle 1199
CALCulate:SPECtrum<sp>:WAVeform:NORMal[:ENABle] 1200

**CALCulate:SPECtrum<sp>:COUPling[:ENABle]** <Enable>

Activates the settings coupling for the specified spectrum.

`CALCulate:SPECtrum<sp>:COUPling:FROM` selects the spectrum whose settings you want to use.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Enable> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Coupling: Enable, Couple from" on page 323

**CALCulate:SPECtrum<sp>:COUPling:FROM** <Spectrum>

Selects the spectrum whose settings are used for the spectrum specified in the suffix if `CALCulate:SPECtrum<sp>:COUPling[:ENABle]` is ON.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Spectrum> Number of the source spectrum

**Usage:** Asynchronous command

**Manual operation:** See "Coupling: Enable, Couple from" on page 323

**CALCulate:SPECtrum<sp>:COUPling:MODE** <ScaleMode>

By default, the vertical scale is adapted to the current measurement results automatically to provide an optimal display. However, if necessary, you can define scaling values manually to suit your requirements.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<ScaleMode> MANual | AUTO

**AUTO**
Vertical scale is set automatically.

**MANual**
Set the vertical scale with `CALCulate:SPECtrum<sp>:MAGNitude:SCALe` and `CALCulate:SPECtrum<sp>:MAGNitude:RANGe`.

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Scale mode" on page 322

---

**CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution]:ADJusted?**

Queries the effective resolution bandwidth.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Return values:**
<AdjResBW> Range: 0 to 200E+9
Increment: 0.01
*RST: 0
Default unit: Hz

**Usage:** Query only
Asynchronous command

---

**CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution]:AUTO**
<AutoRBW>

Couples the frequency span to the "RBW" setting.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<AutoRBW> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Auto RBW" on page 320

---

**CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution]:RATio**
<SpanRBWRatio>

Defines the coupling ratio for Span/RBW.

Available, if `CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution]:AUTO` is set to `ON`.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<SpanRBWRatio> Range: 10 to 10000
Increment: 1
*RST: 1000

**Usage:** Asynchronous command

**Manual operation:** See "Span/RBW" on page 320

**CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution][:VALue]** <RBW>

Queries or defines the used resolution bandwidth.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<RBW> Range: 0.0002 to 2000000
Increment: 0.01
*RST: 2000000
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "RBW" on page 321

**CALCulate:SPECtrum<sp>:FREQuency:CENTer** <Center>

Defines the position of the displayed frequency range, which is (Center - Span/2) to (Center + Span/2). The width of the range is defined using the frequency span setting.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Center> Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0
Default unit: HZ

**Usage:** Asynchronous command

**Manual operation:** See "[Position]" on page 47

**CALCulate:SPECtrum<sp>:FREQuency:SCALe** <XAxisMode>

Defines the scaling method for the frequency axis (x-axis) of the spectrogram.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<XAxisMode> LIN | LOG

`LIN`: linear scaling
`LOG`: logarithmic scaling

*RST: LIN

**Usage:** Asynchronous command

**Manual operation:** See "Frequency axis" on page 323

**CALCulate:SPECtrum<sp>:FREQuency:SPAN** <Span>

The span is specified in Hertz and defines the width of the displayed frequency range, which is (Center - Span/2) to (Center + Span/2). The position of the span is defined using the "Center" setting.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Span>
Range: 0 to 1E+26
Increment: 0.01
*RST: 0.02
Default unit: HZ

**Usage:** Asynchronous command

**Manual operation:** See "[Scale]" on page 47

**CALCulate:SPECtrum<sp>:FREQuency:STARt** <Start>

Defines the start frequency of the displayed frequency span.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Start>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: -0.01
Default unit: HZ

**Usage:** Asynchronous command

**Manual operation:** See "Start" on page 320

**CALCulate:SPECtrum<sp>:FREQuency:STOP** <Stop>

Sets the stop frequency of the displayed frequency span.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Stop>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0.01
Default unit: HZ

**Usage:** Asynchronous command

**Manual operation:** See "Stop" on page 320

**CALCulate:SPECtrum<sp>:FREQuency:WINDow:TYPE** <WindowFunction>

Selects the window type. Windowing helps minimize the discontinuities at the end of the measured signal interval and thus reduces the effect of spectral leakage, increasing the frequency resolution.

Various different window functions are provided in the MXO 3 to suit different input signals. Each of the window functions has specific characteristics, including some advantages and some trade-offs. Consider these characteristics carefully to find the optimum solution for the measurement task.

**Suffix:**

<sp> 1 to 4, index of the spectrum

**Parameters:**

<WindowFunction> RECTangular | HAMMing | HANN | BLACkharris | GAUSsian | FLATTOP2 | KAISerbessel

**RECTangular**
The rectangular window has the best frequency resolution, but a poor amplitude accuracy and is recommended for separating two tones with almost equal amplitudes and a small frequency distance.

**HAMMing**
The Hamming window is bell shaped and has a good frequency resolution and fair amplitude accuracy. It is recommended for frequency response measurements and sine waves, periodic signals and narrowband noise.

**HANN**
The Hann window is bell shaped and has a slightly worse frequency resolution but smaller sidelobe level than the Hamming window. The applications are the same.

**BLACkharris**
The Blackman window is bell shaped and has a poor frequency resolution, but very good amplitude accuracy. It is recommended mainly for signals with single frequencies to detect harmonics.

**GAUSsian**
Good frequency resolution and best magnitude resolution, recommended for weak signals and short duration

**FLATTOP2**
The flat top window has a poor frequency resolution, but the best amplitude accuracy and the sharpest sidelobe. It is recommended for accurate single tone amplitude measurements.

**KAISerbessel**
The kaiser-bessel window has a fair frequency resolution and good amplitude accuracy, and a very high sidelobe level. It is recommended for separating two tones with differing amplitudes and a small frequency distance.

*RST: BLACkharris

**Usage:** Asynchronous command

**Manual operation:** See "Window type" on page 321

---

**CALCulate:SPECtrum<sp>:MAGNitude:LEVel** <VerticalMax>

Sets the maximum displayed value on the vertical scale.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<VerticalMax> Range: -1E+26 to 1000000000000000
Increment: 1
*RST: 10

**Usage:** Asynchronous command

**Manual operation:** See "Vertical maximum" on page 323

---

**CALCulate:SPECtrum<sp>:MAGNitude:RANGe** <VerticalRange>

Sets the range of the spectrum values to be displayed.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<VerticalRange> Range: 0 to 1000000000000000
Increment: 0.01
*RST: 100

**Usage:** Asynchronous command

**Manual operation:** See "Vertical range" on page 323

---

**CALCulate:SPECtrum<sp>:MAGNitude:SCALe** <Unit>

Sets the unit for the y-axis.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Unit> LINear | DB | DBM | DBV | DBUV | DBS | DBHZ
*RST: DBM

**Usage:** Asynchronous command

**Manual operation:** See "Magnitude unit" on page 323

---

**CALCulate:SPECtrum<sp>:PRESet**

Presets the spectrum measurement.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Spectrum preset" on page 322

---

**CALCulate:SPECtrum<sp>:SOURce** <Source>

Selects the source for the spectrum.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Source> C1 | C2 | ... | C8 | M1 | M2 | ... | M8 | R1 | R2 | ... | R8 | TRK1 | TRK2 | TRK3 | ... | TRK16 | PA1QPOWER | PA2QPOWER | PA3QPOWER | PA1SPOWER | PA2SPOWER | PA3SPOWER | PTHREE1QPOWER1 | PTHREE2QPOWER1 | PTHREE3QPOWER1 | PTHREE1QPOWER2 | PTHREE2QPOWER2 | PTHREE3QPOWER2 | PTHREE1QPOWER3 | PTHREE2QPOWER3 | PTHREE3QPOWER3 | PTHREE1QTPOWER | PTHREE2QTPOWER | PTHREE3QTPOWER | PTHREE1HPOWER1 | PTHREE2HPOWER1 | PTHREE3HPOWER1 | PTHREE1HPOWER2 | PTHREE2HPOWER2 | PTHREE3HPOWER2 | PTHREE1HPOWER3 | PTHREE2HPOWER3 | PTHREE3HPOWER3 | PTHREE1HTPOWER | PTHREE2HTPOWER | PTHREE3HTPOWER | PTHREE1EIP1| PTHREE2EIP1| PTHREE3EIP1| PTHREE4EIP1| PTHREE5EIP1| PTHREE6EIP1| PTHREE1EIP2| PTHREE2EIP2| PTHREE3EIP2| PTHREE4EIP2| PTHREE5EIP2| PTHREE6EIP2| PTHREE1EIP3| PTHREE2EIP3| PTHREE3EIP3| PTHREE4EIP3| PTHREE5EIP3| PTHREE6EIP3| PTHREE1EITP| PTHREE2EITP| PTHREE3EITP| PTHREE4EITP| PTHREE5EITP| PTHREE6EITP | PTHREE1EOP1| PTHREE2EOP1| PTHREE3EOP1| PTHREE4EOP1| PTHREE5EOP1| PTHREE6EOP1| PTHREE1EOP2| PTHREE2EOP2| PTHREE3EOP2| PTHREE4EOP2| PTHREE5EOP2| PTHREE6EOP2| PTHREE1EOP3| PTHREE2EOP3| PTHREE3EOP3| PTHREE4EOP3| PTHREE5EOP3| PTHREE6EOP3| PTHREE1EOTP| PTHREE2EOTP| PTHREE3EOTP| PTHREE4EOTP| PTHREE5EOTP| PTHREE6EOTP

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 319

**CALCulate:SPECtrum<sp>:STATe** <State>

Enables the spectrum.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Display" on page 319

**CALCulate:SPECtrum<sp>:THReshold** <Threshold>

Sets an absolute threshold as an additional condition for the peak search. Only peaks that exceed the threshold are detected.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Threshold> Range: -500 to 500
Increment: 1
*RST: -70
Default unit: dBm

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 269

**CALCulate:SPECtrum<sp>:PEXCursion** <PeakExcursion>

Defines a minimum level value by which the waveform must drop left and right of the local maximum to be listed as a peak. Enter a peak excursion value to omit close by peaks and list just the highest peak.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<PeakExcursion> Range: 0 to 100
Increment: 1
*RST: 5
Default unit: dB

**Usage:** Asynchronous command

**Manual operation:** See "Peak excursion" on page 269

**CALCulate:SPECtrum<sp>:WAVeform:AVERage:COUNt** <AverageCount>

Sets the number of segments used for the averaging of the spectrum.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<AverageCount> Range: 2 to 65534
Increment: 1
*RST: 1000

**Usage:** Asynchronous command

**Manual operation:** See "Traces" on page 321

**CALCulate:SPECtrum<sp>:WAVeform:AVERage:ENABle** <Enable>

Enables the average trace.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Enable> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Traces" on page 321

**CALCulate:SPECtrum<sp>:WAVeform:MAXimum:ENABle** <Enable>

Enables the maximum trace.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Enable> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Traces" on page 321

**CALCulate:SPECtrum<sp>:WAVeform:MINimum:ENABle** <Enable>

Enables the minimum trace.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Enable> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Traces" on page 321

---

**CALCulate:SPECtrum<sp>:WAVeform:NORMal[:ENABle]** <Enable>

Enables the normal spectrum trace.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Enable> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Traces" on page 321

## 18.16.2 Spectrum gate

CALCulate:SPECtrum<sp>:GATE:POSition....1200
CALCulate:SPECtrum<sp>:GATE:STARt....1200
CALCulate:SPECtrum<sp>:GATE:STOP....1201
CALCulate:SPECtrum<sp>:GATE:WIDTh....1201

---

**CALCulate:SPECtrum<sp>:GATE:POSition** <Center>

Sets the position of the displayed frequency range.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Center>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0
Default unit: HZ

**Usage:** Asynchronous command

**Manual operation:** See "Position" on page 324

---

**CALCulate:SPECtrum<sp>:GATE:STARt** <Start>

Sets the starting value for the gate.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Start>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: -0.01
Default unit: HZ

**Usage:** Asynchronous command

**Manual operation:** See "Start" on page 324

---

**CALCulate:SPECtrum<sp>:GATE:STOP** <Stop>

Sets the end value for the gate.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Stop>
Range: -1E+26 to 1E+26
Increment: 0.01
*RST: 0.01
Default unit: HZ

**Usage:** Asynchronous command

**Manual operation:** See "Stop" on page 324

---

**CALCulate:SPECtrum<sp>:GATE:WIDTh** <Span>

Defines the width of the displayed gate.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Span>
Range: 0 to 1E+26
Increment: 0.01
*RST: 0.02
Default unit: HZ

**Usage:** Asynchronous command

**Manual operation:** See "Width" on page 324

## 18.16.3 Peak list

CALCulate:SPECtrum<sp>:PLISt:COUNt? 1202
CALCulate:SPECtrum<sp>:PLISt[:STATe] 1202
CALCulate:SPECtrum<sp>:PLISt:MAXCount 1202
CALCulate:SPECtrum<sp>:PLISt:MODE 1202
CALCulate:SPECtrum<sp>:PLISt:SOURce 1203
CALCulate:SPECtrum<sp>:PLISt:SPOW 1203
CALCulate:SPECtrum<sp>:PLISt:RESult[:VALue]? 1203
CALCulate:SPECtrum<sp>:PLISt:SORT:COLumn 1204
CALCulate:SPECtrum<sp>:PLISt:SORT:ORDer 1204
CALCulate:SPECtrum<sp>:PLISt:LABel:FREQuency[:STATe] 1204
CALCulate:SPECtrum<sp>:PLISt:LABel:MAXCount 1204

CALCulate:SPECtrum<sp>:PLISt:LABel:INVert....1205
CALCulate:SPECtrum<sp>:PLISt:LABel:MAXCount....1205
CALCulate:SPECtrum<sp>:PLISt:LABel[:STATe]....1205

**CALCulate:SPECtrum<sp>:PLISt:COUNt?**

Returns the current number of peak list results.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**CALCulate:SPECtrum<sp>:PLISt[:STATe]** <ShowTable>

Enables the display of the peak table.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<ShowTable> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Peak table" on page 326

**CALCulate:SPECtrum<sp>:PLISt:MAXCount** <MaxNoRess>

Sets the maximum number of measurement results that are listed in the result table.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<MaxNoRess> Range: 1 to 1000
Increment: 1
*RST: 10

**Usage:** Asynchronous command

**Manual operation:** See "Max results" on page 327

**CALCulate:SPECtrum<sp>:PLISt:MODE** <ResultMode>

Selects how the measurement results are displayed.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<ResultMode> ABS | REL
*RST: ABS

**Usage:** Asynchronous command

**Manual operation:** See "Result mode" on page 327

---

**CALCulate:SPECtrum<sp>:PLISt:SOURce** <Source>

Selects the source of the peak table. You can select one of the traces that is enabled with Traces.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Source> SPECMAXH1 | SPECMINH1 | SPECNORM1 | SPECAVER1 | SPECMAXH2 | SPECMINH2 | SPECNORM2 | SPECAVER2 | SPECMAXH3 | SPECMINH3 | SPECNORM3 | SPECAVER3 | SPECMAXH4 | SPECMINH4 | SPECNORM4 | SPECAVER4

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 328

---

**CALCulate:SPECtrum<sp>:PLISt:SPOW** <ShowPeaks>

Displays a box with a description for each detected peak in the spectrum, including the magnitude.

If `CALCulate:SPECtrum<sp>:PLISt:LABel:FREQuency[:STATe]` is `ON`, the frequency values are also displayed.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<ShowPeaks> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Peak on waveform" on page 326

---

**CALCulate:SPECtrum<sp>:PLISt:RESult[:VALue]?**

Returns the current peak list measurement results.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Return values:**
<Value> Comma-separated list of results

**Usage:** Query only
Asynchronous command

---

**CALCulate:SPECtrum<sp>:PLISt:SORT:COLumn** <ResultColumn>

Sorts the results in the spectrum peak list table either according to the frequency or according to the value.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<ResultColumn> FREQ | VAL
*RST: FREQ

**Usage:** Asynchronous command

---

**CALCulate:SPECtrum<sp>:PLISt:SORT:ORDer** <ResultOrder>

Defines if the spectrum peak list results are sorted in an ascending (increasing) or descending (decreasing) order.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<ResultOrder> ASC | DESC
`ASC`: ascending
`DESC`: descending
*RST: ASC

**Usage:** Asynchronous command

---

**CALCulate:SPECtrum<sp>:PLISt:LABel:FREQuency[:STATe]** <ShowFrequency>

Includes the frequency of the detected peak in the diagram labels.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<ShowFrequency> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Show frequency" on page 328

---

**CALCulate:SPECtrum<sp>:PLISt:LABel:MAXCount** <MaxNumberPeaks>

Sets the maximum number of measurement results that are listed in the result table.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<MaxNumberPeaks> Range: 1 to 100
Increment: 1
*RST: 10

**Usage:** Asynchronous command

---

**CALCulate:SPECtrum<sp>:PLISt:LABel:INVert** <Inverse>

Inverts the colors of the peak list labels, the peak boxes are shown with a white background.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<Inverse> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

---

**CALCulate:SPECtrum<sp>:PLISt:LABel:MAXCount** <MaxNumberPeaks>

Sets the maximum number of measurement results that are listed in the result table.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<MaxNumberPeaks> Range: 1 to 100
Increment: 1
*RST: 10

**Usage:** Asynchronous command

---

**CALCulate:SPECtrum<sp>:PLISt:LABel[:STATe]** <ShowLabels>

Displays the labels in the peak list diagram.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<ShowLabels> OFF | ON
*RST: ON

**Usage:** Asynchronous command

### 18.16.4 Spectrum with frames and overlap

CALCulate:SPECtrum<sp>:OVERlap:ENABle....1206
CALCulate:SPECtrum<sp>:OVERlap:FACTor:VALue....1206
CALCulate:SPECtrum<sp>:OVERlap:FACTor:ADJusted?....1206
CALCulate:SPECtrum<sp>:OVERlap:FCOunt:LIMit:ENABle....1207
CALCulate:SPECtrum<sp>:OVERlap:FCOunt:VALue....1207
CALCulate:SPECtrum<sp>:OVERlap:FCOunt:ADJusted?....1207
CALCulate:SPECtrum<sp>:OVERlap:FWIDth?....1208

**CALCulate:SPECtrum<sp>:OVERlap:ENABle** <State>

Activates the use of frames and overlap to create the FFT spectrum.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 328

**CALCulate:SPECtrum<sp>:OVERlap:FACTor:VALue** <OverlapFactor>

Defines the minimum factor by which two neighboring frames overlap.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Parameters:**
<OverlapFactor> Range: 0 to 90
Increment: 1
*RST: 20
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Overlap factor" on page 329

**CALCulate:SPECtrum<sp>:OVERlap:FACTor:ADJusted?**

Returns the adjusted overlap factor that is used for calculation. It is based on the given minimum overlap factor and the current settings.

**Suffix:**
<sp> 1 to 4, index of the spectrum

**Return values:**

<AdjOverlapFact> Range: 0 to 100
Increment: 1
*RST: 20
Default unit: %

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Overlap factor" on page 329

---

**CALCulate:SPECtrum<sp>:OVERlap:FCOunt:LIMit:ENABle** <LimitNumberFFT>

Limits the number of FFTs that are done per acquisition. To set the number of FFTs, use `CALCulate:SPECtrum<sp>:OVERlap:FCOunt:VALue`.

**Suffix:**

<sp> 1 to 4, index of the spectrum

**Parameters:**

<LimitNumberFFT> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Limit # of FFTs" on page 328

---

**CALCulate:SPECtrum<sp>:OVERlap:FCOunt:VALue** <FFTPerAcq>

Sets the number of frames to be used for FFT with `CALCulate:SPECtrum<sp>:OVERlap:FCOunt:LIMit:ENABle` set ON.

**Suffix:**

<sp> 1 to 4, index of the spectrum

**Parameters:**

<FFTPerAcq> Range: 1 to 4294967295
Increment: 1
*RST: 1000

**Usage:** Asynchronous command

**Manual operation:** See "FFTs per acq." on page 329

---

**CALCulate:SPECtrum<sp>:OVERlap:FCOunt:ADJusted?**

Returns the adjusted number of frames that is used for FFT calculation.

**Suffix:**

<sp> 1 to 4, index of the spectrum

**Return values:**

<AdjFFTPerAcq> Range: 1 to 4294967295
Increment: 1
*RST: 1000

**Usage:** Query only
Asynchronous command

**Manual operation:** See "FFTs per acq." on page 329

---

**CALCulate:SPECtrum<sp>:OVERlap:FWIDth?**

Returns the frame width.

**Suffix:**

<sp> 1 to 4, index of the spectrum

**Return values:**

<FrameWidth> Range: 1E-16 to 10000
Increment: 1
*RST: 1
Default unit: s

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Frame width" on page 329

### 18.16.5 Data export of spectrum waveforms

For each of the spectrum traces, a dedicated command is available. You find the commands in Section 18.13.9, "Data transfer to remote computer", on page 1141. For a description of the traces, see "Traces" on page 321.

For fast export of several waveforms at once, use `EXPort:WAVeform:DATA[:VALues]?` on page 1150.

For export of spectrum data to file, see Section 18.13.4, "Waveform data: export to file", on page 1127.

## 18.17 Applications

### 18.17.1 Mask testing

Some of the commands in the following section are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, section "Command sequence and synchronization"

- Mask test definition ..... 1209
- Polygon definition ..... 1212
- Auto mask definition ..... 1215
- Actions ..... 1218
- Results ..... 1220

### 18.17.1.1 Mask test definition

MTESt<m>:ADD ..... 1209
MTESt<m>:COUNt? ..... 1209
MTESt<m>:DIAGram ..... 1210
MTESt<m>:IMEXport:NAME ..... 1210
MTESt<m>:IMEXport:OPEN ..... 1210
MTESt<m>:IMEXport:SAVE ..... 1210
MTESt<m>:REMove ..... 1211
MTESt<m>:SOURce ..... 1211
MTESt<m>:STATe ..... 1211
MTESt<m>:VISible ..... 1211

**MTESt<m>:ADD**

Creates a mask test with the indicated index.

**Suffix:**
<m> 1 to 8, index of the mask

**Example:** See Section 18.5.2, "Creating a mask", on page 945.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add mask" on page 333

**MTESt<m>:COUNt?**

Returns the number of masks.

`MTESt:COUNt? MAX` returns the maximum number of masks that can be created.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Return values:**
<Count> Number of defined masks

**Usage:** Query only
Asynchronous command

**MTESt<m>:DIAGram** <DiagramKey>

Sets the layout and the diagram where the mask is located and the test runs.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<DiagramKey> String with the layout-diagram key.

**Example:**
```
:MTESt:DIAGram?
-->"L1_D1"
```

**Usage:** Asynchronous command

---

**MTESt<m>:IMEXport:NAME** <Name>

Sets the path, the filename and the file format of the mask file.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<Name> String with path and file name with extension `.xml`.

**Example:**
```
MTES2:IMEX:NAME
'/home/storage/userData/Masks/MyMask2.xml'
```
File and path definition for mask test 2.

**Usage:** Asynchronous command

**Manual operation:** See "Save mask" on page 334

---

**MTESt<m>:IMEXport:OPEN**

Opens and loads the mask selected by `MTESt<m>:IMEXport:NAME`.

**Suffix:**
<m> 1 to 8, index of the mask

**Usage:** Event
Asynchronous command

**Manual operation:** See "Open" on page 334

---

**MTESt<m>:IMEXport:SAVE**

Saves the mask test to the file selected by `MTESt<m>:IMEXport:NAME`.

**Suffix:**
<m> 1 to 8, index of the mask

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save mask" on page 334

**MTESt<m>:REMove**

Deletes the selected mask.

**Suffix:**
<m> 1 to 8, index of the mask

**Usage:** Setting only
Asynchronous command

**MTESt<m>:SOURce** <Source>

Selects the waveform to be tested against the mask. For auto mask definition, it is also the waveform from which the mask is created.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<Source> C1 | C2 | ... | C8 | M1 | M2 | ... | M8 | XY1 | XY2 | XY3 | XY4 | SPECNORM1 | SPECNORM2 | SPECNORM3 | SPECNORM4

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 334

**MTESt<m>:STATe** <State>

Activates or deactivates the mask test.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<State> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable test" on page 333

**MTESt<m>:VISible** <DisplayState>

Displays all mask segments of the selected mask in the diagrams, or hides them.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<DisplayState> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Display" on page 333

### 18.17.1.2 Polygon definition

MTESt<m>:SEGMent<n>:ADD ..... 1212
MTESt<m>:SEGMent<n>:COUNt? ..... 1212
MTESt<m>:SEGMent<n>:POINt<o>:ADD ..... 1212
MTESt<m>:SEGMent<n>:POINt<o>:COUNt? ..... 1213
MTESt<m>:SEGMent<n>:POINt<o>:REMove ..... 1213
MTESt<m>:SEGMent<n>:POINt<o>:VALid? ..... 1213
MTESt<m>:SEGMent<n>:POINt<o>:X ..... 1214
MTESt<m>:SEGMent<n>:POINt<o>:Y ..... 1214
MTESt<m>:SEGMent<n>:REMove ..... 1215
MTESt<m>:SEGMent<n>:VALid? ..... 1215

**MTESt<m>:SEGMent<n>:ADD**

Adds a new segment to the selected mask. The segment has no points, use `MTESt<m>:SEGMent<n>:POINt<o>:ADD` to add the points.

**Suffix:**

<m> 1 to 8, index of the mask

<n> 1 to 8, index of the segment
Omit the suffix to use the next free suffix.

**Example:** See Section 18.5.2, "Creating a mask", on page 945.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add segment" on page 335

**MTESt<m>:SEGMent<n>:COUNt?**

Returns the number of segments that belong to the indicated mask.

`MTESt<m>:SEGMent:COUNt? MAX` returns the maximum number of segments that can be created.

**Suffix:**

<m> 1 to 8, index of the mask

<n> Irrelevant, omit the suffix.

**Return values:**

<Count> Number of defined masks

**Usage:** Query only
Asynchronous command

**MTESt<m>:SEGMent<n>:POINt<o>:ADD**

Adds a corner point to the selected mask segment at the next free suffix. The new point has the coordinates 0;0.

**Suffix:**

<m> 1 to 8, index of the mask

<n> 1 to 8, index of the segment

<o> 1 to *, index of the segment point

**Example:** See Section 18.5.2, "Creating a mask", on page 945.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add +" on page 336

---

**MTESt<m>:SEGMent<n>:POINt<o>:COUNt?**

Returns the number of points that was added to the indicated mask segment.

You can query the maximum value with `<command>? MAX`.

**Suffix:**

<m> 1 to 8, index of the mask

<n> 1 to 8, index of the segment

<o> Irrelevant, omit the suffix.

**Parameters:**

<Count> Number of defined points

**Example:** See Section 18.5.2, "Creating a mask", on page 945.

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Add +" on page 336

---

**MTESt<m>:SEGMent<n>:POINt<o>:REMove**

Deletes the indicated corner point of the mask segment.

**Suffix:**

<m> 1 to 8, index of the mask

<n> 1 to 8, index of the segment

<o> 1 to *, index of the segment point

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Delete, Delete all" on page 337

---

**MTESt<m>:SEGMent<n>:POINt<o>:VALid?**

Checks the validity of the selected point. See "Polygon masks" on page 331.

**Suffix:**
<m> 1 to 8, index of the mask
<n> 1 to 8, index of the segment
<o> 1 to *, index of the segment point

**Return values:**
<Valid> OFF | ON
ON = valid
OFF = invalid
*RST: ON

**Usage:** Query only
Asynchronous command

---

**MTESt<m>:SEGMent<n>:POINt<o>:X** <X>

Sets the horizontal position of the selected corner point.

**Suffix:**
<m> 1 to 8, index of the mask
<n> 1 to 8, index of the segment
<o> 1 to *, index of the segment point

**Parameters:**
<X> Range: -1E+26 to 1E+26
Increment: 1E-06
*RST: 0
Default unit: s

**Example:** See Section 18.5.2, "Creating a mask", on page 945.

**Usage:** Asynchronous command

**Manual operation:** See "Point, X, Y" on page 336

---

**MTESt<m>:SEGMent<n>:POINt<o>:Y** <Y>

Sets the vertical position of the selected corner point.

**Suffix:**
<m> 1 to 8, index of the mask
<n> 1 to 8, index of the segment
<o> 1 to *, index of the segment point

**Parameters:**
<Y> Range: -1E+26 to 1E+26
Increment: 1E-06
*RST: 0
Default unit: V

**Example:** See Section 18.5.2, "Creating a mask", on page 945.

**Usage:** Asynchronous command

**Manual operation:** See "Point, X, Y" on page 336

---

**MTESt<m>:SEGMent<n>:REMove**

Deletes the specified mask segment.

**Suffix:**
<m> 1 to 8, index of the mask

<n> 1 to 8, index of the segment

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Delete, Delete all" on page 337

---

**MTESt<m>:SEGMent<n>:VALid?**

Checks the validity of the indicated segment. The segment is invalid if one of its points is invalid. See "Polygon masks" on page 331.

**Suffix:**
<m> 1 to 8, index of the mask

<n> 1 to 8, index of the segment

**Return values:**
<Valid> OFF | ON

ON = valid
OFF = invalid

*RST: ON

**Usage:** Query only
Asynchronous command

### 18.17.1.3 Auto mask definition

MTESt<m>:AUTomask:UPDate ..... 1216
MTESt<m>:AUTomask:XTOLerance:DIVisions ..... 1216
MTESt<m>:AUTomask:XTOLerance:PHYSical ..... 1216
MTESt<m>:AUTomask:YTOLerance:DIVisions ..... 1217
MTESt<m>:AUTomask:YTOLerance:PHYSical ..... 1217
MTESt<m>:AUTomask:XTUNit ..... 1217
MTESt<m>:AUTomask:YTUNit ..... 1217
MTESt<m>:AUTomask:VALues:BOTTom? ..... 1218
MTESt<m>:AUTomask:VALues:TOP? ..... 1218

**MTESt<m>:AUTomask:UPDate**

Creates or updates the upper and lower mask limit from the envelope of the selected source by copying the waveform data. See also: `MTESt<m>:SOURce` on page 1211.

**Suffix:**
<m> 1 to 8, index of the mask

**Usage:** Event
Asynchronous command

**Manual operation:** See "Create/Update" on page 337

**MTESt<m>:AUTomask:XTOLerance:DIVisions** <HorizTol>

Sets the width of the mask in horizontal direction in divisions, and sets `MTESt<m>:AUTomask:XTUNit` to `DIVisions`.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<HorizTol> Tolerance value in divisions. The overall mask width is twice the specified value.

Range: 0 to 10
Increment: 0.01
*RST: 0.2

**Usage:** Asynchronous command

**Manual operation:** See "Tolerance (Horizontal)" on page 338

**MTESt<m>:AUTomask:XTOLerance:PHYSical** <HorizTol>

Sets the width of the mask in horizontal direction in the unit of the horizontal axis, and sets `MTESt<m>:AUTomask:XTUNit` to `PHYSical`.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<HorizTol> Tolerance value in the physical unit of the x-axis. The overall mask width is twice the specified value.

Range: 0 to 10
Increment: 0.01
*RST: 0.2

**Usage:** Asynchronous command

**Manual operation:** See "Tolerance (Horizontal)" on page 338

**MTESt<m>:AUTomask:YTOLerance:DIVisions** <VertTol>

Sets the height of the mask in vertical direction in divisions, and sets `MTESt<m>:AUTomask:YTUNit` to `DIVisions`.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<VertTol> Tolerance value in divisions. The overall mask height is twice the specified value.

Range: 0 to 10
Increment: 0.01
*RST: 0.2

**Usage:** Asynchronous command

**Manual operation:** See "Tolerance (Vertical)" on page 338

**MTESt<m>:AUTomask:YTOLerance:PHYSical** <VertTol>

Sets the height of the mask in vertical direction in the unit of the vertical axis, and sets `MTESt<m>:AUTomask:YTUNit` to `PHYSical`.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<VertTol> Tolerance value in the physical unit of the y-axis. The overall mask height is twice the specified value.

Range: 0 to 10
Increment: 0.01
*RST: 0.2

**Usage:** Asynchronous command

**Manual operation:** See "Tolerance (Vertical)" on page 338

**MTESt<m>:AUTomask:XTUNit** <HorizTolUnt>
**MTESt<m>:AUTomask:YTUNit** <VertTolUnt>

Set the unit of the mask tolerance in horizontal and vertical direction: divisions or in the unit of the corresponding axis.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<VertTolUnt> DIVisions | PHYSical

*RST: DIVisions

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 338

**MTESt<m>:AUTomask:VALues:BOTTom?**

Returns the points of the lower mask limit.

**Suffix:**
<m> 1 to 8, index of the mask

**Return values:**
<Points> Comma-separated list of values
Range: -INF to INF
Increment: 0.01
*RST: 0

**Usage:** Query only
Asynchronous command

**MTESt<m>:AUTomask:VALues:TOP?**

Returns the points of the upper mask limit.

**Suffix:**
<m> 1 to 8, index of the mask

**Return values:**
<Points> Comma-separated list of values
Range: -INF to INF
Increment: 0.01
*RST: 0

**Usage:** Query only
Asynchronous command

### 18.17.1.4 Actions

MTESt<m>:ONViolation:BEEP ..... 1218
MTESt<m>:ONViolation:SCReenshot ..... 1219
MTESt<m>:ONViolation:SRESults ..... 1219
MTESt<m>:ONViolation:STOP ..... 1219
MTESt<m>:ONViolation:TRIGgerout ..... 1220
MTESt<m>:ONViolation:WFMSave ..... 1220

**MTESt<m>:ONViolation:BEEP** <Beep>

Generates a beep sound if the command is set to `SUCCess` or `VIOLation`.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<Beep> NOACtion | SUCCess | VIOLation
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Beep" on page 339

---

**MTESt<m>:ONViolation:SCReenshot** <SaveScreenshot>

Saves the waveform data to file if the command is set to `SUCCess` or `VIOLation`.

To configure the screenshot settings, use the commands described in Section 18.13.8, "Screenshots", on page 1137.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<SaveScreenshot> NOACtion | SUCCess | VIOLation
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Screenshot" on page 340

---

**MTESt<m>:ONViolation:SRESults** <SaveResults>

Saves the selected results to file if the command is set to `SUCCess` or `VIOLation`.

To define the path and file names, use the EXPort:RESult:AUTonaming:* commands:

- `EXPort:RESult:AUTonaming:NAME` on page 1135
- `EXPort:RESult:AUTonaming:PATH` on page 1136
- `EXPort:RESult:AUTonaming:TYPE` on page 1136

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<SaveResults> NOACtion | SUCCess | VIOLation
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Save results" on page 340

---

**MTESt<m>:ONViolation:STOP** <StopAcq>

Stops the running acquisition if the command is set to `SUCCess` or `VIOLation`.

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<StopAcq> NOACtion | SUCCess | VIOLation
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Stop acq" on page 339

**MTESt<m>:ONViolation:TRIGgerout** <TrigOutPls>

Sends an outgoing pulse to the [Trigger Out] connector if the command is set to `SUCCess` or `VIOLation`.

To configure the pulse, user the following commands:

- `TRIGger:ACTions:OUT:SOURce` on page 1063
- `TRIGger:ACTions:OUT:POLarity` on page 1064
- `TRIGger:ACTions:OUT:DELay` on page 1063
- `TRIGger:ACTions:OUT:PLENgth` on page 1064

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<TrigOutPls> NOACtion | SUCCess | VIOLation
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Trigger out pulse" on page 340

**MTESt<m>:ONViolation:WFMSave** <SaveWfm>

Saves the waveform data to file if the command is set to `SUCCess` or `VIOLation`.

To define the path and file names, use the EXPort:WAVeform:AUTonaming:* commands:

- `EXPort:WAVeform:AUTonaming:NAME` on page 1127
- `EXPort:WAVeform:AUTonaming:PATH` on page 1127
- `EXPort:WAVeform:AUTonaming:TYPE` on page 1128

**Suffix:**
<m> 1 to 8, index of the mask

**Parameters:**
<SaveWfm> NOACtion | SUCCess | VIOLation
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Save wfm" on page 339

### 18.17.1.5 Results

MTESt<m>:RESult:COUNt:FWAVeforms? 1221
MTESt<m>:RESult:COUNt:PWAVeforms? 1221
MTESt<m>:RESult:COUNt:WAVeforms? 1221
MTESt<m>:RESult:FRATe? 1221
MTESt<m>:RESult[:RESult]? 1222

**MTESt<m>:RESult:COUNt:FWAVeforms?**

Returns the number of failed acquisitions.

**Suffix:**
<m> 1 to 8, index of the mask

**Return values:**
<AcqsFailed> Range: 0 to 10000000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Failed acq." on page 332

**MTESt<m>:RESult:COUNt:PWAVeforms?**

Returns the number of acquisitions that passed the test successfully.

**Suffix:**
<m> 1 to 8, index of the mask

**Return values:**
<AcqsPassed> Range: 0 to 10000000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Passed acq." on page 332

**MTESt<m>:RESult:COUNt:WAVeforms?**

Returns the number of tested acquisitions.

**Suffix:**
<m> 1 to 8, index of the mask

**Return values:**
<AcqsCompleted> Range: 0 to 10000000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Total acq." on page 331

**MTESt<m>:RESult:FRATe?**

Returns the fail rate, the ratio of failed acquisitions to the number of tested acquisitions.

**Suffix:**
<m> 1 to 8, index of the mask

**Return values:**
<FailRate> Range: -1E+26 to 1E+26
Increment: 0
*RST: 0
Default unit: %

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Fail rate" on page 332

---

**MTESt<m>:RESult[:RESult]?**

Returns the overall test result.

**Suffix:**
<m> 1 to 8, index of the mask

**Return values:**
<TestResult> PASS | FAIL
*RST: PASS

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Result" on page 332

## 18.17.2 Frequency response and impedance analyses (options R&S MXO3-K36/K337)


- Settings....1222
- Generator settings....1229
- Amplitude profile....1230
- Diagram settings....1232
- Calibration settings for frequency response analysis....1235
- Analysis results....1236
- Reference waveforms....1241
- Export of analysis data....1245


### 18.17.2.1 Settings


FRANalysis:AUToscale....1223
FRANalysis:AUToscale:RESult[:ENABle]....1223
FRANalysis:ENABle....1223
FRANalysis:FREQuency:MODE....1224
FRANalysis:FREQuency:STARt....1224
FRANalysis:FREQuency:STOP....1224
FRANalysis:HDEFinition[:ENABle]....1224

FRANalysis:IMPedance[:MODE] 1225
FRANalysis:INPut[:SOURce] 1225
FRANalysis:MEASurement:DELay:STATe 1225
FRANalysis:MEASurement:DELay:MODE 1225
FRANalysis:MEASurement:DELay[:TIME] 1226
FRANalysis:MEASurement:DELay:PERiod[:NUMBer] 1226
FRANalysis:MEASurement:POINt[:DISPlay] 1226
FRANalysis:MEASurement:RBW 1227
FRANalysis:MODE 1227
FRANalysis:OUTPut[:SOURce] 1227
FRANalysis:PHASe:MAXimum 1227
FRANalysis:POINts:LOGarithmic 1228
FRANalysis:POINts:MODE 1228
FRANalysis:POINts:TOTal 1228
FRANalysis:REPeat 1229
FRANalysis:RESet 1229
FRANalysis:STATe 1229

**FRANalysis:AUToscale** <AutoScale>

Enables the auto scaling function for each measurement.

**Parameters:**

<AutoScale> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Auto scale" on page 356

**FRANalysis:AUToscale:RESult[:ENABle]** <State>

If ON, the scaling of the plot diagram is adjusted when new measurement results are determined.

**Parameters:**

<State> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Auto scale result" on page 356

**FRANalysis:ENABle** <State>

Enables the frequency response analysis application.

If the frequency response analysis is disabled, the instrument does not accept any `FRANalysis` command.

You can start the analysis with `FRANalysis:STATe`.

**Parameters:**
<State> OFF | ON

**Usage:** Asynchronous command

---

**FRANalysis:FREQuency:MODE** <FrequencyMode>

Selects between single frequency and sweep analysis.

**Parameters:**
<FrequencyMode> SWEep | SINGle
*RST: SWEep

**Usage:** Asynchronous command

**Manual operation:** See "Frequency mode" on page 351

---

**FRANalysis:FREQuency:STARt** <StartFrequency>

Sets the start frequency of the sweep.

**Parameters:**
<StartFrequency> Range: 0.01 to 50E+6
Increment: 1
*RST: 100
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop" on page 351

---

**FRANalysis:FREQuency:STOP** <StopFrequency>

Sets the stop frequency of the sweep.

**Parameters:**
<StopFrequency> Range: 0.01 to 50E+6
Increment: 1
*RST: 10000
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop" on page 351

---

**FRANalysis:HDEFinition[:ENABle]** <HDMode>

Disables the HD mode, which is active by default. In particular, disable the HD mode if you analyze switching peaks.

**Parameters:**
<HDMode> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "HD mode" on page 357

---

**FRANalysis:IMPedance[:MODE]** <MeasMode>

Selects the measurement method for the impedance analysis.

**Parameters:**

<MeasMode> SHUNt | SERies | VCURrent

SHUNt: Shunt through
SERies: Series through
VCURrent: Voltage current

*RST: SERies

**Manual operation:** See "Measure type" on page 351

---

**FRANalysis:INPut[:SOURce]** <InputChannel>

Sets the channel for the input signal of the DUT.

**Parameters:**

<InputChannel> C1 | C2 | ... | C8

*RST: C1

**Usage:** Asynchronous command

**Manual operation:** See "Input" on page 350

---

**FRANalysis:MEASurement:DELay:STATe** <MeasDelay>

Enables the measurement delay.

**Parameters:**

<MeasDelay> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Meas. delay mode, Delay time, Delay period" on page 356

---

**FRANalysis:MEASurement:DELay:MODE** <MeasDelayMode>

Selects the delay mode.

**Parameters:**

<MeasDelayMode> TIME | PERiod

*RST: TIME

**Usage:** Asynchronous command

**Manual operation:** See "Meas. delay mode, Delay time, Delay period" on page 356

**FRANalysis:MEASurement:DELay[:TIME]** <MeasDelayTime>

Sets a time delay, that the system waits before measuring the next point of the plot. This is helpful in systems that need more time to adapt to the new frequency, for example if filters with significant time group delays are present.

The settings takes effect if if `FRANalysis:MEASurement:DELay:STATe` = `ON` and `FRANalysis:MEASurement:DELay:MODE` = `PERiod`.

**Parameters:**

<MeasDelayTime>
Range: 0 to 10
Increment: 0.001
*RST: 0
Default unit: s

**Example:**

```
FRANalysis:MEASurement:DELay:STATe ON
FRANalysis:MEASurement:DELay:MODE TIME
FRANalysis:MEASurement:DELay 0.005
```

**Usage:** Asynchronous command

**Manual operation:** See "Meas. delay mode, Delay time, Delay period" on page 356

**FRANalysis:MEASurement:DELay:PERiod[:NUMBer]** <MeasDelayPeriod>

Sets a period delay, which the system waits before measuring the next point of the plot.

The settings takes effect if if `FRANalysis:MEASurement:DELay:STATe` = `ON` and `FRANalysis:MEASurement:DELay:MODE` = `PERiod`.

**Parameters:**

<MeasDelayPeriod>
Range: 0.01 to 1000
Increment: 0.01
*RST: 1

**Example:**

```
FRANalysis:MEASurement:DELay:STATe ON
FRANalysis:MEASurement:DELay:MODE PERiod
FRANalysis:MEASurement:DELay:PERiod 0.05
```

**Usage:** Asynchronous command

**Manual operation:** See "Meas. delay mode, Delay time, Delay period" on page 356

**FRANalysis:MEASurement:POINt[:DISPlay]** <Points>

Enables the display of the measurement points for the frequency response analysis.

**Parameters:**

<Points>
OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Display points" on page 357

**FRANalysis:MEASurement:RBW** <RBW>

Sets the resolution bandwidth, which determines the number of measurements that are used for creating the plot.

**Parameters:**

<RBW> HIGH | MID | LOW
*RST: HIGH

**Usage:** Asynchronous command

**Manual operation:** See "RBW" on page 356

**FRANalysis:MODE** <ApplicationMd>

Selects the application mode.

- "Gain/phase": Frequency response analysis (option R&S MXO3-K36)
- "Impedance": Impedance analysis (option R&S MXO3-K337)

**Parameters:**

<ApplicationMd> GPHase | IMPedance
*RST: GPHase

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 350

**FRANalysis:OUTPut[:SOURce]** <OutputChannel>

Sets the channel for the output signal of the DUT.

**Parameters:**

<OutputChannel> C1 | C2 | ... | C8
*RST: C2

**Usage:** Asynchronous command

**Manual operation:** See "Output" on page 351

**FRANalysis:PHASe:MAXimum** <MaxPhase>

Sets the upper boundary of the vertical phase window.

The lower boundary is given by "Maximum phase" - 360°.

By default, the "Maximum phase" is set to 180° for a phase window ranging from -180° to 180° accordingly.

Only available for frequency response analysis.

**Parameters:**

<MaxPhase> Range: 0 to 360
Increment: 1
*RST: 180
Default unit: °

**Usage:** Asynchronous command

**Manual operation:** See "Maximum phase" on page 355

---

**FRANalysis:POINts:LOGarithmic** <PtsPerDecade>

Selects the number of points that are measured per decade, if `FRANalysis:POINts:MODE` is set to `DECade`.

**Parameters:**

<PtsPerDecade> Range: 10 to 500
Increment: 1
*RST: 10

**Usage:** Asynchronous command

**Manual operation:** See "Points" on page 351

---

**FRANalysis:POINts:MODE** <PointsMode>

Selects, if the number of points for the FRA are measured as total or per decade.

You can set the number of points with `FRANalysis:POINts:TOTal`/ `FRANalysis:POINts:LOGarithmic`.

**Parameters:**

<PointsMode> TOTal | DECade
*RST: DECade

**Usage:** Asynchronous command

**Manual operation:** See "Points" on page 351

---

**FRANalysis:POINts:TOTal** <TotalPoints>

Set the total number of points for the FRA analysis, if `FRANalysis:POINts:MODE` on page 1228 is set to `TOTal`.

**Parameters:**

<TotalPoints> Range: 10 to 5000
Increment: 1
*RST: 100

**Usage:** Asynchronous command

**Manual operation:** See "Points" on page 351

**FRANalysis:REPeat** <Repeat>

Repeats the measurement, using the same parameters.

**Parameters:**
<Repeat> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Repeat" on page 352

**FRANalysis:RESet**

Resets the frequency response analysis.

**Usage:** Setting only
Asynchronous command

**FRANalysis:STATe** <Value>

Starts the frequency response analysis.

**Parameters:**
<Value> RUN | STOP | OFF

**Usage:** Asynchronous command

**Manual operation:** See "Run" on page 352

### 18.17.2.2 Generator settings

FRANalysis:AMPLitude:MODE ..... 1229
FRANalysis:GENerator[:CHANnel] ..... 1230
FRANalysis:GENerator:AMPLitude ..... 1230
FRANalysis:GENerator:LOAD ..... 1230

**FRANalysis:AMPLitude:MODE** <AmplitudeMode>

Selects, if the amplitude is a constant value (`FRANalysis:GENerator:AMPLitude`) or is defined as an amplitude profile.

**Parameters:**
<AmplitudeMode> CONStant | PROFile
*RST: CONStant

**Usage:** Asynchronous command

**Manual operation:** See "Amplitude mode" on page 352

**FRANalysis:GENerator[:CHANnel]** <GenCh>

Selects the built-in generator to start a frequency sweep for a defined frequency range.

**Parameters:**

<GenCh> GEN1

BOTH: only available for impedance analysis, synchronizes the generators.

*RST: GEN1

**Usage:** Asynchronous command

**Manual operation:** See "Generator" on page 352

**FRANalysis:GENerator:AMPLitude** <GenAmplitude>

Sets a fixed amplitude for the generated signal.

**Parameters:**

<GenAmplitude> Range: 0.01 to 12
Increment: 0.1
*RST: 1
Default unit: Vpp

**Usage:** Asynchronous command

**Manual operation:** See "Amplitude" on page 353

**FRANalysis:GENerator:LOAD** <GenLoad>

Selects the generator voltage display for 50Ω or high impedance load.

**Parameters:**

<GenLoad> FIFTy | HIZ

`HIZ`: high input impedance

*RST: FIFTy

**Usage:** Asynchronous command

**Manual operation:** See "User load" on page 353

### 18.17.2.3 Amplitude profile

FRANalysis:AMPLitude:PROFile:APOint ..... 1231
FRANalysis:AMPLitude:PROFile:COUNt ..... 1231
FRANalysis:AMPLitude:PROFile:MODE ..... 1231
FRANalysis:AMPLitude:PROFile:POINt<m>:AMPLitude ..... 1231
FRANalysis:AMPLitude:PROFile:POINt<m>:FREQuency ..... 1231
FRANalysis:AMPLitude:PROFile:POINt<m>:REMove ..... 1232
FRANalysis:AMPLitude:PROFile:SORT ..... 1232

**FRANalysis:AMPLitude:PROFile:APOint**

Adds a new step to the amplitude profile.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add" on page 354

**FRANalysis:AMPLitude:PROFile:COUNt** <Value>

Sets the number of defined points for the amplitude profile.

**Parameters:**
<Value>

**Usage:** Asynchronous command

**Manual operation:** See "Add" on page 354

**FRANalysis:AMPLitude:PROFile:MODE** <VoltageChange>

Selects if the voltage change is done as a single step or as a ramp.

**Parameters:**
<VoltageChange> SINGle | RAMP
*RST: SINGle

**Usage:** Asynchronous command

**Manual operation:** See "Voltage change" on page 354

**FRANalysis:AMPLitude:PROFile:POINt<m>:AMPLitude** <Amplitude>

Sets the amplitude for the selected point.

**Suffix:**
<m> Index of the amplitude profile point

**Parameters:**
<Amplitude> Range: 0 to 5.995
Increment: 0.1
*RST: 1
Default unit: Vpp

**Usage:** Asynchronous command

**Manual operation:** See "Step start freq, Amplitude" on page 354

**FRANalysis:AMPLitude:PROFile:POINt<m>:FREQuency** <Frequency>

Sets the start frequency for the selected point.

**Suffix:**
<m> Index of the amplitude profile point

**Parameters:**
<Frequency> Range: 0.01 to 100E6
Increment: 1
*RST: 100
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Step start freq, Amplitude" on page 354

---

**FRANalysis:AMPLitude:PROFile:POINt<m>:REMove**

Removes the specified step from the amplitude profile.

**Suffix:**
<m> 1..*, step index

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Delete" on page 354

---

**FRANalysis:AMPLitude:PROFile:SORT**

Sorts the steps in the amplitude table by frequency, starting with the lowest frequency.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Sort" on page 354

### 18.17.2.4 Diagram settings

FRANalysis:AMPLitude:ENABle ..... 1233
FRANalysis:AMPLitude:OFFSet ..... 1233
FRANalysis:AMPLitude:SCALe ..... 1233
FRANalysis:GAIN:ENABle ..... 1233
FRANalysis:GAIN:OFFSet ..... 1233
FRANalysis:GAIN:SCALe ..... 1234
FRANalysis:IMPedance:ENABle ..... 1234
FRANalysis:IMPedance:SCALe ..... 1234
FRANalysis:IMPedance:STARt ..... 1234
FRANalysis:IMPedance:STOP ..... 1234
FRANalysis:PHASe:ENABle ..... 1235
FRANalysis:PHASe:OFFSet ..... 1235
FRANalysis:PHASe:SCALe ..... 1235

**FRANalysis:AMPLitude:ENABle** <State>

Enables the amplitude signal for the frequency response analysis. You can then define the amplitude profile of the signal.

**Parameters:**

<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Amplitude mode" on page 352

**FRANalysis:AMPLitude:OFFSet** <VerticalOffset>

Sets a vertical offset of the amplitude waveform.

**Parameters:**

<VerticalOffset> Range: -100 to 100
Increment: 0.01
*RST: 5
Default unit: Vpp

**Usage:** Asynchronous command

**FRANalysis:AMPLitude:SCALe** <VerticalScale>

Sets the vertical scale for the amplitude waveform.

**Parameters:**

<VerticalScale> Range: 0.01 to 10
Increment: 0.01
*RST: 0.5
Default unit: Vpp/div

**Usage:** Asynchronous command

**FRANalysis:GAIN:ENABle** <State>

Enables the gain waveform for the frequency response analysis.

**Parameters:**

<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**FRANalysis:GAIN:OFFSet** <VerticalOffset>

Sets a vertical offset of the gain waveform.

**Parameters:**

<VerticalOffset> Range: -2000 to 2000
Increment: 0.001
*RST: 10
Default unit: dB

**Usage:** Asynchronous command

**Manual operation:** See "Vertical offset" on page 358

---

**FRANalysis:GAIN:SCALe** <VerticalScale>

Sets the vertical scale for the gain waveform.

**Parameters:**

<VerticalScale> Range: 0.1 to 20
Increment: 1
*RST: 10
Default unit: dB/div

**Usage:** Asynchronous command

**Manual operation:** See "Vertical scale" on page 358

---

**FRANalysis:IMPedance:ENABle** <State>

Enables the impedance waveform for the impedance analysis.

**Parameters:**

<State> OFF | ON
*RST: OFF

---

**FRANalysis:IMPedance:SCALe** <Mode>

Sets the scale mode for the impedance trace.

**Parameters:**

<Mode> LIN | LOG
*RST: LOG

**Usage:** Asynchronous command

**Manual operation:** See "Vertical scale" on page 358

---

**FRANalysis:IMPedance:STARt** <Value>
**FRANalysis:IMPedance:STOP** <Value>

Set the lower and upper value of the vertical scale for the impedance trace.

**Parameters:**

<Value>
Range: 2E-09 to 1000000000
Increment: 1
*RST: 1000
Default unit: Ω

**Usage:** Asynchronous command

**Manual operation:** See "Vertical start, Vertical stop" on page 358

---

**FRANalysis:PHASe:ENABle** <State>

Enables the phase waveform for the frequency response analysis.

**Parameters:**

<State>
OFF | ON
*RST: OFF

**Usage:** Asynchronous command

---

**FRANalysis:PHASe:OFFSet** <VerticalOffset>

Sets a vertical offset of the phase waveform.

**Parameters:**

<VerticalOffset>
Range: -10000 to 10000
Increment: 0.001
*RST: 0
Default unit: °

**Usage:** Asynchronous command

**Manual operation:** See "Vertical offset" on page 358

---

**FRANalysis:PHASe:SCALe** <VerticalScale>

Sets the vertical scale for the phase waveform.

**Parameters:**

<VerticalScale>
Range: 1 to 180
Increment: 1
*RST: 36
Default unit: °/div

**Usage:** Asynchronous command

**Manual operation:** See "Vertical scale" on page 358

### 18.17.2.5 Calibration settings for frequency response analysis

FRANalysis:CALibration:CALibration....1236
FRANalysis:CALibration:RESult?....1236
FRANalysis:CALibration:STATe....1236

**FRANalysis:CALibration:CALibration**

Runs a calibration sequence.

**Usage:** Event

**Manual operation:** See "Start calibration" on page 361

**FRANalysis:CALibration:RESult?**

Returns the result of the calibration.

**Return values:**

<States> PASS | FAIL | RUN | NOAL

`PASS`: the calibration is successful.
`FAIL`: the calibration failed.
`RUN`: a calibration cycle is running.
`NOAL`: no active calibration.

*RST: NOAL

**Usage:** Query only

**Manual operation:** See "Start calibration" on page 361

**FRANalysis:CALibration:STATe** <UseCalData>

If ON, the user calibration data is used for the frequency response analysis.

**Parameters:**

<UseCalData> OFF | ON

*RST: OFF

**Manual operation:** See "Use calibration data" on page 361

#### 18.17.2.6 Analysis results

FRANalysis:MARGin:GAIN:FREQuency? 1237
FRANalysis:MARGin:GAIN:VALue? 1237
FRANalysis:MARGin:PHASe:FREQuency? 1237
FRANalysis:MARGin:PHASe:VALue? 1237
FRANalysis:MARGin:STATe 1237
FRANalysis:MARKer<m>:FREQuency 1238
FRANalysis:MARKer<m>:IMPedance? 1238
FRANalysis:MARKer<m>:GAIN? 1238
FRANalysis:MARKer<m>:PHASe? 1238
FRANalysis:MARKer<m>:DIFFerence:FREQuency? 1238
FRANalysis:MARKer<m>:DIFFerence:IMPedance? 1239
FRANalysis:MARKer<m>:DIFFerence:GAIN? 1239
FRANalysis:MARKer<m>:DIFFerence:PHASe? 1239
FRANalysis:MARKer<m>:INDex? 1239

FRANalysis:MARKer<m>:SSCReen ..... 1240
FRANalysis:MARKer<m>:STATe ..... 1240
FRANalysis:RESult:STATe ..... 1240

**FRANalysis:MARGin:GAIN:FREQuency?**

Returns the frequency of the gain margin.

**Return values:**
<Frequency>

**Usage:** Query only
Asynchronous command

**FRANalysis:MARGin:GAIN:VALue?**

Returns the value of the gain margin.

**Return values:**
<Phase>

**Usage:** Query only
Asynchronous command

**FRANalysis:MARGin:PHASe:FREQuency?**

Returns the frequency of the phase margin.

**Return values:**
<Frequency>

**Usage:** Query only
Asynchronous command

**FRANalysis:MARGin:PHASe:VALue?**

Returns the value of the phase margin.

**Return values:**
<Phase>

**Usage:** Query only
Asynchronous command

**FRANalysis:MARGin:STATe** <Margins>

Enables the display of the margin table for the frequency response analysis.

**Parameters:**
<Margins> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Margin" on page 357

---

**FRANalysis:MARKer<m>:FREQuency** <Frequency>

Sets the frequency for the specified marker, which defines the horizontal marker position.

**Suffix:**
<m> 1..2
Selects the marker number.

**Parameters:**
<Frequency> Numeric value in Hz

**Usage:** Asynchronous command

---

**FRANalysis:MARKer<m>:IMPedance?**

Returns the impedance for the specified marker.

**Suffix:**
<m> 1..2
Selects the marker number.

**Usage:** Query only
Asynchronous command

---

**FRANalysis:MARKer<m>:GAIN?**

Returns the gain for the specified marker.

**Suffix:**
<m> 1..2
Selects the marker number.

**Usage:** Query only
Asynchronous command

---

**FRANalysis:MARKer<m>:PHASe?**

Returns the phase value for the specified marker.

**Suffix:**
<m> 1..2
Selects the marker number.

**Usage:** Query only
Asynchronous command

---

**FRANalysis:MARKer<m>:DIFFerence:FREQuency?**

Returns the delta value of the frequency between the two markers.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Return values:**
<DeltaFrequency>

**Usage:** Query only
Asynchronous command

---

**FRANalysis:MARKer<m>:DIFFerence:IMPedance?**

Returns the delta value of the impedance between the two markers.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Return values:**
<DeltaImpedance>

**Usage:** Query only
Asynchronous command

---

**FRANalysis:MARKer<m>:DIFFerence:GAIN?**

Returns the delta value of the gain between the two markers.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Return values:**
<DeltaGain>

**Usage:** Query only
Asynchronous command

---

**FRANalysis:MARKer<m>:DIFFerence:PHASe?**

Returns the delta value of the phase between the two markers.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Return values:**
<DeltaPhase>

**Usage:** Query only
Asynchronous command

---

**FRANalysis:MARKer<m>:INDex?**

Returns the point index in the plot where the specified marker is positioned.

**Suffix:**
<m> 1..2
Selects the marker number.

**Return values:**
<Index> The lowest index is 0, which corresponds to line 1 in the result table.
*PointIndex = # in result table - 1*

**Usage:** Query only
Asynchronous command

---

**FRANalysis:MARKer<m>:SSCReen**

Resets the markers to their initial positions. Reset is helpful if the markers have disappeared from the display or need to be moved for a larger distance.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Usage:** Setting only
Asynchronous command

---

**FRANalysis:MARKer<m>:STATe** <Markers>

Activates the display of the marker table and the markers in the plot.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Parameters:**
<Markers> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Markers" on page 357

---

**FRANalysis:RESult:STATe** <Table>

Enables the display of the result table for the FRA.

**Parameters:**
<Table> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Result table" on page 357

### 18.17.2.7 Reference waveforms

FRANalysis:REFCurve<rc>:CLEar....1241
FRANalysis:REFCurve<rc>:DATA?....1241
FRANalysis:REFCurve<rc>:ENABle....1241
FRANalysis:REFCurve<rc>:NAME....1242
FRANalysis:REFCurve<rc>:OFFSet....1242
FRANalysis:REFCurve<rc>:OPEN....1242
FRANalysis:REFCurve<rc>:SAVE....1243
FRANalysis:REFCurve<rc>:SCALe....1243
FRANalysis:REFCurve<rc>:SOURce....1243
FRANalysis:REFCurve<rc>:TOORiginal....1243
FRANalysis:REFCurve<rc>:UPDate....1244
FRANalysis:REFCurve<rc>:VMODe....1244
FRANalysis:MARKer<m>:REFerence<rc>?....1244
FRANalysis:MARKer<m>:DIFFerence:REFerence<rc>?....1244

**FRANalysis:REFCurve<rc>:CLEar**

Deletes the selected reference waveform. It disappears from the display, and its memory is deleted.

**Suffix:**
<rc> 1...4, index of the reference waveform

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Clear" on page 365

**FRANalysis:REFCurve<rc>:DATA?**

Returns the data of the specified reference waveform.

**Suffix:**
<rc> 1...4, index of the reference waveform

**Return values:**
<Data> Comma-separated values

**Usage:** Query only
Asynchronous command

**FRANalysis:REFCurve<rc>:ENABle** <State>

Enables the display of the reference waveform in the diagram. Before you can display it, create the reference waveform.

**Suffix:**
<rc> 1...4, index of the reference waveform

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Display" on page 364

---

**FRANalysis:REFCurve<rc>:NAME** <Name>

Defines the name of the reference waveform file to be loaded, saved or deleted.

**Suffix:**
<rc> 1...4, index of the reference waveform

**Parameters:**
<Name> String with the file path, filename and extension.

**Usage:** Asynchronous command

**Manual operation:** See "Save as" on page 365

---

**FRANalysis:REFCurve<rc>:OFFSet** <VerticalOffset>

The vertical offset moves the reference waveform vertically. Enter a value with the unit of the waveform.

**Suffix:**
<rc> 1...4, index of the reference waveform

**Parameters:**
<VerticalOffset> Range: -100 to 100
Increment: 1
*RST: 5

**Usage:** Asynchronous command

**Manual operation:** See "Vertical offset" on page 367

---

**FRANalysis:REFCurve<rc>:OPEN**

Loads the reference waveform file selected by `FRANalysis:REFCurve<rc>:NAME`.

**Suffix:**
<rc> 1...4, index of the reference waveform

**Usage:** Event
Asynchronous command

**Manual operation:** See "Open" on page 365

**FRANalysis:REFCurve<rc>:SAVE**

Saves the FRA reference waveform to the file defined by `FRANalysis:REFCurve<rc>:NAME`.

**Suffix:**
<rc> 1...4, index of the reference waveform

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save as" on page 365

**FRANalysis:REFCurve<rc>:SCALe** <VerticalScale>

Sets the vertical scale, which defines the displayed amplitude of the selected waveform.

**Suffix:**
<rc> 1...4, index of the reference waveform

**Parameters:**
<VerticalScale> Range: -100 to 100
Increment: 1
*RST: 0.5

**Usage:** Asynchronous command

**Manual operation:** See "Vertical scale" on page 366

**FRANalysis:REFCurve<rc>:SOURce** <Source>

Selects the source waveform of the reference.

**Suffix:**
<rc> 1...4, index of the reference waveform

**Parameters:**
<Source> FGAin | FPHase | FIMP | FREF1 | FREF2 | FREF3 | FREF4

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 364

**FRANalysis:REFCurve<rc>:TOORiginal**

Restores the original vertical settings of the reference waveform.

**Suffix:**
<rc> 1...4, index of the reference waveform

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Set to original" on page 366

**FRANalysis:REFCurve<rc>:UPDate**

Copies the selected source waveform with all its settings to the memory of the reference waveform. If there is a previously defined reference waveform in this memory, it is updated by the current source waveform.

**Suffix:**
<rc> 1...4, index of the reference waveform

**Usage:** Event
Asynchronous command

**Manual operation:** See "Create/Update" on page 365

**FRANalysis:REFCurve<rc>:VMODe** <VerticalMode>

Selects the type of vertical settingsfor the reference waveform.

**Suffix:**
<rc> 1...4, index of the reference waveform

**Parameters:**
<VerticalMode> COUPled | INDependent

**COUPled**
Vertical position and scale of the source are used.

**INDependent**
Scaling and position can be set specific to the reference waveform.

*RST: COUPled

**Usage:** Asynchronous command

**Manual operation:** See "Vertical mode" on page 366

**FRANalysis:MARKer<m>:REFerence<rc>?**

Returns the vertical value of the reference waveform at the specified marker.

**Suffix:**
<m> 1..2
Selects the marker number.

<rc> 1...4, index of the reference waveform

**Usage:** Query only
Asynchronous command

**FRANalysis:MARKer<m>:DIFFerence:REFerence<rc>?**

Returns the difference between the two markers on the reference waveform. If no marker values are exist, or they are invalid, an execution erroris returned.

**Suffix:**
<m> Irrelevant, omit the suffix.

<rc> 1...4, index of the reference waveform

**Return values:**
<DeltaReference> Value of the difference

**Usage:** Query only
Asynchronous command

### 18.17.2.8 Export of analysis data

EXPort:RESult:SELect:FRA:MARGin ..... 1245
EXPort:RESult:SELect:FRA:MARKer ..... 1245
EXPort:RESult:SELect:FRA:RESult ..... 1245
FRANalysis:GAIN:DATA? ..... 1246
FRANalysis:FREQuency:DATA? ..... 1246
FRANalysis:IMPedance:DATA? ..... 1246
FRANalysis:PHASe:DATA? ..... 1247

**EXPort:RESult:SELect:FRA:MARGin** <MarginResult>

If enabled, includes the margin results in the results export file of the FRA.

**Parameters:**
<MarginResult> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "FRA results" on page 362

**EXPort:RESult:SELect:FRA:MARKer** <MarkerResult>

If enabled, includes the marker results in the results export file of the FRA.

**Parameters:**
<MarkerResult> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "FRA results" on page 362

**EXPort:RESult:SELect:FRA:RESult** <Result>

If enabled, includes the frequency response analysis results, including the frequency, gain, phase and amplitude, in the results export file of the FRA.

**Parameters:**
<Result> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "FRA results" on page 362

---

**FRANalysis:GAIN:DATA?**

Returns the data of the gain as a list of comma-separated values in dB.

**Return values:**
<Data> Comma-separated list of values

**Example:**
```
FRAN:GAIN:DATA?
```
Result:
```
0.4381956630254851,0.4593246383252179,
0.461895233375762,0.4570717993766185,
0.4623802509651942,0.4533450898899926,
0.4408194360143725,0.4221600784741697,
0.4483342212879299,0.4473558449908476,
0.4915016244058707
```

**Usage:** Query only
Asynchronous command

---

**FRANalysis:FREQuency:DATA?**

Returns the data of the frequency points for which gain and phase have been calculated as a list of comma-separated values in Hz.

**Return values:**
<Data> Comma-separated list of values

**Example:**
```
FRAN:FREQ:DATA?
```
Result:
```
-0.07463742650985239,-0.09417628145200041,
-0.1457021392091207,-0.1914790795969807,
-0.2494368612845221,-0.3415108478964939,
-0.4350150706042513,-0.5673151547214843,
-0.7363486980048108,-0.9055923620805615,
-1.0829557075417
```

**Usage:** Query only
Asynchronous command

---

**FRANalysis:IMPedance:DATA?**

Returns the impedance results as a list of comma-separated values in Ohm.

**Return values:**
<Data> Comma-separated list of values

**Usage:** Query only

**FRANalysis:PHASe:DATA?**

Returns the data of the phase as a list of comma-separated values in degree.

**Return values:**

<Data> Comma-separated list of values

**Example:**

```
FRAN:PHAS:DATA?
```

Result:

```
-0.07463742650985239,-0.09417628145200041,
-0.1457021392091207,-0.1914790795969807,
-0.2494368612845221,-0.3415108478964939,
-0.4350150706042513,-0.5673151547214843,
-0.7363486980048108,-0.9055923620805615,
-1.0829557075417
```

**Usage:** Query only
Asynchronous command

## 18.17.3 Power analysis (option R&S MXO3-K31)

Some of the commands in the following section are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, section "Command sequence and synchronization"

### Overview of statistic results

The remote commands for the power analysis measurements can retrieve statistic results. The abbreviations in the commands have the following meaning:

| Command | Description |
|---|---|
| `POWer<m>:xxx:RESult:nnn:[:ACTual]?` | Current measurement result |
| `POWer<m>:xxx:RESult:nnn::AVERage?` | Average of the measurement results |
| `POWer<m>:xxx:RESult:nnn::EVTCount?` | Number of calculated measurement results (periods) |
| `POWer<m>:xxx:RESult:nnn:RMS?` | RMS value of the measurement results |
| `POWer<m>:xxx:RESult:nnn::PPEak?` | Positive peak value of the measurement results |
| `POWer<m>:xxx:RESult:nnn::NPEak?` | Negative peak value of the measurement results |

| Command | Description |
|---|---|
| POWer<m>:xxx:RESult:xxx::STDDev? | Standard deviation of the measurement results |
| POWer<m>:xxx:RESult:nnn::WFMCount? | Number of calculated waveforms |


- General settings....1248
- Power quality....1250
- Power harmonics....1263
- Switching loss....1272
- Turn on/off time....1282
- Power efficiency....1286
- Safe operating area (SOA)....1293
- Power waveform....1303
- Reference level....1306
- Power analysis data....1312


### 18.17.3.1 General settings


POWer<m>[:ENABle]....1248
POWer<m>:TYPE....1248
POWer<m>:ASKew[:EXECute]....1249
EXPort:RESult:SELect:POWer<m>....1249
POWer<m>:EFFiciency:STATistics[:ENABle]....1249
POWer<m>:SWITching:STATistics:ENABle....1249
POWer<m>:QUALity:STATistics[:ENABle]....1249
POWer<m>:EFFiciency:STATistics:RESet....1250
POWer<m>:SWITching:STATistics:RESet....1250
POWer<m>:QUALity:STATistics:RESet....1250
POWer<m>:EFFiciency:STATistics:WFMCount?....1250
POWer<m>:SWITching:STATistics:WFMCount?....1250
POWer<m>:QUALity:STATistics:WFMCount?....1250


**POWer<m>[:ENABle]** <State>

Enables the power measurement instance.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<State> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "+ Add" on page 371

**POWer<m>:TYPE** <Type>

Sets the type for the respective power analysis measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Type> QUALity | HARMonics | SWITching | ONOFf | EFFiciency | SOA
SWITching = switching loss
ONOFf = turn on/off time
SOA = safe operating area

**Example:** `POW1:TYPE QUAL`
Sets the first power analysis measurement to quality.

**Usage:** Asynchronous command

**Manual operation:** See "+ Add" on page 371

---

**POWer<m>:ASKew[:EXECute]**

Performs auto deskew adjustment.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Auto deskew" on page 372

---

**EXPort:RESult:SELect:POWer<m>** <ExportResults>

If enabled, includes the results of selected power analysis measurement in the results export file.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ExportResults> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

---

**POWer<m>:EFFiciency:STATistics[:ENABle]** <State>
**POWer<m>:SWITching:STATistics:ENABle** <State>
**POWer<m>:QUALity:STATistics[:ENABle]** <State>

The commands activate statistical calculation for the selected power measurement. Make sure that the suffix matches the selected power measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<State> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Statistics" on page 376

---

**POWer<m>:EFFiciency:STATistics:RESet**
**POWer<m>:SWITching:STATistics:RESet**
**POWer<m>:QUALity:STATistics:RESet**

The commands restart statistical calculation for the selected power measurement. Make sure that the suffix matches the selected power measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Clear results" on page 376

---

**POWer<m>:EFFiciency:STATistics:WFMCount?**
**POWer<m>:SWITching:STATistics:WFMCount?**
**POWer<m>:QUALity:STATistics:WFMCount?**

Return the number of waveforms included in the selected power analysis. Make sure that the suffix matches the selected power measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<WaveformCount> Number of analyzed waveforms.

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Statistics" on page 376

### 18.17.3.2 Power quality


- Power quality setup....1250
- Power quality measurement display....1252
- Power quality results....1255


#### Power quality setup


POWer<m>:QUALity:SOURce:CURRent....1251
POWer<m>:QUALity:SOURce[:VOLTage]....1251
POWer<m>:QUALity:SOURce:CYCLe....1251

POWer<m>:QUALity:FREQuency[:VALue]....1251
POWer<m>:QUALity:FREQuency:USER....1252
POWer<m>:QUALity:GATE....1252

**POWer<m>:QUALity:SOURce:CURRent** <CurrentSource>

Selects the current source waveform. Analog channels and math waveforms can be used.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<CurrentSource> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Current" on page 375

**POWer<m>:QUALity:SOURce[:VOLTage]** <VoltageSource>

Selects the voltage source waveform. Analog channels and math waveforms can be used.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<VoltageSource> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Voltage" on page 375

**POWer<m>:QUALity:SOURce:CYCLe** <CycleSource>

Selects if voltage or current is used for the cycle measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<CycleSource> VOLTage | CURRent
*RST: VOLTage

**Usage:** Asynchronous command

**Manual operation:** See "Cycle source" on page 375

**POWer<m>:QUALity:FREQuency[:VALue]** <FundamFreq>

Sets the input frequency of the source signal in Hz.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<FundamFreq> F50 | F60 | F400 | USER | AUTO
*RST: F50

**Usage:** Asynchronous command

**Manual operation:** See "Fund. frequency" on page 375

---

**POWer<m>:QUALity:FREQuency:USER** <FundamentalFreq>

Sets the user-defined frequency, if `POWer<m>:QUALity:FREQuency[:VALue]` is set to `USER`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<FundamentalFreq> Range: 10 to 1000000
Increment: 1
*RST: 50
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Fund. frequency" on page 375

---

**POWer<m>:QUALity:GATE** <Gate>

Selects the gate that is used for limiting the range of the power quality measurement.

Enable the gate before you assign a measurement to it (`GATE<g>:ENABle` =`ON`).

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Gate> Number of the gate to be used

**Usage:** Asynchronous command

### Power quality measurement display

---

**POWer<m>:QUALity:DISPlay:CURRent:CREStfactor** <ShwCurrCrest>

Enables the current crest factor measurement for the power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShwCurrCrest> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**POWer<m>:QUALity:DISPlay:CURRent:PEAK** <ShwCurrPeak>

Enables the current peak measurement for the power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShwCurrPeak> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**POWer<m>:QUALity:DISPlay:CURRent:RMS** <ShowCurrentRMS>

Enables the current RMS measurement for the power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShowCurrentRMS> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**POWer<m>:QUALity:DISPlay:POWer:ACTive** <ShwActPow>

Enables the active power measurement for the power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShwActPow> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**POWer<m>:QUALity:DISPlay:POWer:APParent** <ShwApparentPow>

Enables the apparent power measurement for the power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShwApparentPow> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**POWer<m>:QUALity:DISPlay:POWer:PFACtor** <ShwPowFact>

Enables the power factor measurement for the power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShwPowFact> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**POWer<m>:QUALity:DISPlay:POWer:PHASe** <ShowPhase>

Enables the power phase measurement for the power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShowPhase> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**POWer<m>:QUALity:DISPlay:POWer:REACtive** <ShwReactivePow>

Enables the reactive power measurement for the power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShwReactivePow> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**POWer<m>:QUALity:DISPlay:VOLTage:CRESTfactor** <ShwVoltageCrest>

Enables the voltage crest factor measurement for the power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShwVoltageCrest> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**POWer<m>:QUALity:DISPlay:VOLTage:FREQuency** <ShwVoltageFreq>

Enables the voltage frequency measurement for the power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShwVoltageFreq> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**POWer<m>:QUALity:DISPlay:VOLTage:PEAK** <ShwVoltagePeak>

Enables the voltage peak value measurement for the power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShwVoltagePeak> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**POWer<m>:QUALity:DISPlay:VOLTage:RMS** <ShowVoltageRMS>

Enables the voltage RMS measurement for the power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShowVoltageRMS> OFF | ON
*RST: ON

**Usage:** Asynchronous command

### Power quality results

See "Overview of statistic results" on page 1247.

POWer<m>:QUALity:RESult:CURRent:CRESTfactor[:ACTual]? 1257
POWer<m>:QUALity:RESult:CURRent:CRESTfactor:AVERage? 1257
POWer<m>:QUALity:RESult:CURRent:CRESTfactor:EVTCount? 1257
POWer<m>:QUALity:RESult:CURRent:CRESTfactor:NPEak? 1257
POWer<m>:QUALity:RESult:CURRent:CRESTfactor:PPEak? 1257
POWer<m>:QUALity:RESult:CURRent:CRESTfactor:RMS? 1257
POWer<m>:QUALity:RESult:CURRent:CRESTfactor:WFMCount? 1257
POWer<m>:QUALity:RESult:CURRent:CRESTfactor:STDDev? 1257
POWer<m>:QUALity:RESult:CURRent:PEAK[:ACTual]? 1258
POWer<m>:QUALity:RESult:CURRent:PEAK:AVERage? 1258

POWer<m>:QUALity:RESult:CURRent:PEAK:EVTCount? 1258
POWer<m>:QUALity:RESult:CURRent:PEAK:NPEak? 1258
POWer<m>:QUALity:RESult:CURRent:PEAK:PPEak? 1258
POWer<m>:QUALity:RESult:CURRent:PEAK:RMS? 1258
POWer<m>:QUALity:RESult:CURRent:PEAK:WFMCount? 1258
POWer<m>:QUALity:RESult:CURRent:PEAK:STDDev? 1258
POWer<m>:QUALity:RESult:CURRent:RMS[:ACTual]? 1258
POWer<m>:QUALity:RESult:CURRent:RMS:AVERage? 1258
POWer<m>:QUALity:RESult:CURRent:RMS:EVTCount? 1258
POWer<m>:QUALity:RESult:CURRent:RMS:NPEak? 1258
POWer<m>:QUALity:RESult:CURRent:RMS:PPEak? 1258
POWer<m>:QUALity:RESult:CURRent:RMS:RMS? 1258
POWer<m>:QUALity:RESult:CURRent:RMS:WFMCount? 1258
POWer<m>:QUALity:RESult:CURRent:RMS:STDDev? 1258
POWer<m>:QUALity:RESult:POWer:ACTive[:ACTual]? 1259
POWer<m>:QUALity:RESult:POWer:ACTive:AVERage? 1259
POWer<m>:QUALity:RESult:POWer:ACTive:EVTCount? 1259
POWer<m>:QUALity:RESult:POWer:ACTive:NPEak? 1259
POWer<m>:QUALity:RESult:POWer:ACTive:PPEak? 1259
POWer<m>:QUALity:RESult:POWer:ACTive:RMS? 1259
POWer<m>:QUALity:RESult:POWer:ACTive:WFMCount? 1259
POWer<m>:QUALity:RESult:POWer:ACTive:STDDev? 1259
POWer<m>:QUALity:RESult:POWer:APParent[:ACTual]? 1259
POWer<m>:QUALity:RESult:POWer:APParent:AVERage? 1259
POWer<m>:QUALity:RESult:POWer:APParent:EVTCount? 1259
POWer<m>:QUALity:RESult:POWer:APParent:NPEak? 1259
POWer<m>:QUALity:RESult:POWer:APParent:PPEak? 1259
POWer<m>:QUALity:RESult:POWer:APParent:RMS? 1259
POWer<m>:QUALity:RESult:POWer:APParent:WFMCount? 1259
POWer<m>:QUALity:RESult:POWer:APParent:STDDev? 1259
POWer<m>:QUALity:RESult:POWer:PFACtor[:ACTual]? 1260
POWer<m>:QUALity:RESult:POWer:PFACtor:AVERage? 1260
POWer<m>:QUALity:RESult:POWer:PFACtor:EVTCount? 1260
POWer<m>:QUALity:RESult:POWer:PFACtor:NPEak? 1260
POWer<m>:QUALity:RESult:POWer:PFACtor:PPEak? 1260
POWer<m>:QUALity:RESult:POWer:PFACtor:RMS? 1260
POWer<m>:QUALity:RESult:POWer:PFACtor:WFMCount? 1260
POWer<m>:QUALity:RESult:POWer:PFACtor:STDDev? 1260
POWer<m>:QUALity:RESult:POWer:PHASe[:ACTual]? 1260
POWer<m>:QUALity:RESult:POWer:PHASe:AVERage? 1260
POWer<m>:QUALity:RESult:POWer:PHASe:EVTCount? 1260
POWer<m>:QUALity:RESult:POWer:PHASe:NPEak? 1260
POWer<m>:QUALity:RESult:POWer:PHASe:PPEak? 1260
POWer<m>:QUALity:RESult:POWer:PHASe:RMS? 1260
POWer<m>:QUALity:RESult:POWer:PHASe:WFMCount? 1260
POWer<m>:QUALity:RESult:POWer:PHASe:STDDev? 1260
POWer<m>:QUALity:RESult:POWer:REACtive[:ACTual]? 1261
POWer<m>:QUALity:RESult:POWer:REACtive:AVERage? 1261
POWer<m>:QUALity:RESult:POWer:REACtive:EVTCount? 1261
POWer<m>:QUALity:RESult:POWer:REACtive:NPEak? 1261

POWer<m>:QUALity:RESult:POWer:REACtive:PPEak? 1261
POWer<m>:QUALity:RESult:POWer:REACtive:RMS? 1261
POWer<m>:QUALity:RESult:POWer:REACtive:WFMCount? 1261
POWer<m>:QUALity:RESult:POWer:REACtive:STDDev? 1261
POWer<m>:QUALity:RESult:VOLTage:CRESt factor[:ACTual]? 1261
POWer<m>:QUALity:RESult:VOLTage:CRESt factor:AVERage? 1261
POWer<m>:QUALity:RESult:VOLTage:CRESt factor:EVTCount? 1261
POWer<m>:QUALity:RESult:VOLTage:CRESt factor:NPEak? 1261
POWer<m>:QUALity:RESult:VOLTage:CRESt factor:PPEak? 1261
POWer<m>:QUALity:RESult:VOLTage:CRESt factor:RMS? 1261
POWer<m>:QUALity:RESult:VOLTage:CRESt factor:WFMCount? 1261
POWer<m>:QUALity:RESult:VOLTage:CRESt factor:STDDev? 1261
POWer<m>:QUALity:RESult:VOLTage:FREQuency[:ACTual]? 1261
POWer<m>:QUALity:RESult:VOLTage:FREQuency:AVERage? 1261
POWer<m>:QUALity:RESult:VOLTage:FREQuency:EVTCount? 1261
POWer<m>:QUALity:RESult:VOLTage:FREQuency:NPEak? 1261
POWer<m>:QUALity:RESult:VOLTage:FREQuency:PPEak? 1262
POWer<m>:QUALity:RESult:VOLTage:FREQuency:RMS? 1262
POWer<m>:QUALity:RESult:VOLTage:FREQuency:WFMCount? 1262
POWer<m>:QUALity:RESult:VOLTage:FREQuency:STDDev? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK[:ACTual]? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK:AVERage? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK:EVTCount? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK:NPEak? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK:PPEak? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK:RMS? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK:WFMCount? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK:STDDev? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS[:ACTual]? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS:AVERage? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS:EVTCount? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS:NPEak? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS:PPEak? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS:RMS? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS:WFMCount? 1263
POWer<m>:QUALity:RESult:VOLTage:RMS:STDDev? 1263

---

**POWer<m>:QUALity:RESult:CURRent:CRESt factor[:ACTual]?**
**POWer<m>:QUALity:RESult:CURRent:CRESt factor:AVERage?**
**POWer<m>:QUALity:RESult:CURRent:CRESt factor:EVTCount?**
**POWer<m>:QUALity:RESult:CURRent:CRESt factor:NPEak?**
**POWer<m>:QUALity:RESult:CURRent:CRESt factor:PPEak?**
**POWer<m>:QUALity:RESult:CURRent:CRESt factor:RMS?**
**POWer<m>:QUALity:RESult:CURRent:CRESt factor:WFMCount?**
**POWer<m>:QUALity:RESult:CURRent:CRESt factor:STDDev?**

Returns the current crest factor, the *Peak value / RMS value* for the power quality analysis.

For details on the statistics, see "Overview of statistic results" on page 1247.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:QUALity:RESult:CURRent:PEAK[:ACTual]?**
**POWer<m>:QUALity:RESult:CURRent:PEAK:AVERage?**
**POWer<m>:QUALity:RESult:CURRent:PEAK:EVTCount?**
**POWer<m>:QUALity:RESult:CURRent:PEAK:NPEak?**
**POWer<m>:QUALity:RESult:CURRent:PEAK:PPEak?**
**POWer<m>:QUALity:RESult:CURRent:PEAK:RMS?**
**POWer<m>:QUALity:RESult:CURRent:PEAK:WFMCount?**
**POWer<m>:QUALity:RESult:CURRent:PEAK:STDDev?**

Returns the current peak value, the highest measured magnitude value for the power quality analysis.

For details on the statistics, see "Overview of statistic results" on page 1247.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:QUALity:RESult:CURRent:RMS[:ACTual]?**
**POWer<m>:QUALity:RESult:CURRent:RMS:AVERage?**
**POWer<m>:QUALity:RESult:CURRent:RMS:EVTCount?**
**POWer<m>:QUALity:RESult:CURRent:RMS:NPEak?**
**POWer<m>:QUALity:RESult:CURRent:RMS:PPEak?**
**POWer<m>:QUALity:RESult:CURRent:RMS:RMS?**
**POWer<m>:QUALity:RESult:CURRent:RMS:WFMCount?**
**POWer<m>:QUALity:RESult:CURRent:RMS:STDDev?**

Returns the root mean square of the current for the power quality analysis.

For details on the statistics, see "Overview of statistic results" on page 1247.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**

<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:QUALity:RESult:POWer:ACTive[:ACTual]?**
**POWer<m>:QUALity:RESult:POWer:ACTive:AVERage?**
**POWer<m>:QUALity:RESult:POWer:ACTive:EVTCount?**
**POWer<m>:QUALity:RESult:POWer:ACTive:NPEak?**
**POWer<m>:QUALity:RESult:POWer:ACTive:PPEak?**
**POWer<m>:QUALity:RESult:POWer:ACTive:RMS?**
**POWer<m>:QUALity:RESult:POWer:ACTive:WFMCount?**
**POWer<m>:QUALity:RESult:POWer:ACTive:STDDev?**

Returns the active power for the power quality analysis.

For details on the statistics, see "Overview of statistic results" on page 1247.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

**Return values:**

<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:QUALity:RESult:POWer:APParent[:ACTual]?**
**POWer<m>:QUALity:RESult:POWer:APParent:AVERage?**
**POWer<m>:QUALity:RESult:POWer:APParent:EVTCount?**
**POWer<m>:QUALity:RESult:POWer:APParent:NPEak?**
**POWer<m>:QUALity:RESult:POWer:APParent:PPEak?**
**POWer<m>:QUALity:RESult:POWer:APParent:RMS?**
**POWer<m>:QUALity:RESult:POWer:APParent:WFMCount?**
**POWer<m>:QUALity:RESult:POWer:APParent:STDDev?**

Returns the apparent power for the power quality analysis.

For details on the statistics, see "Overview of statistic results" on page 1247.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

**Return values:**

<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:QUALity:RESult:POWer:PFACtor[:ACTual]?**
**POWer<m>:QUALity:RESult:POWer:PFACtor:AVERage?**
**POWer<m>:QUALity:RESult:POWer:PFACtor:EVTCount?**
**POWer<m>:QUALity:RESult:POWer:PFACtor:NPEak?**
**POWer<m>:QUALity:RESult:POWer:PFACtor:PPEak?**
**POWer<m>:QUALity:RESult:POWer:PFACtor:RMS?**
**POWer<m>:QUALity:RESult:POWer:PFACtor:WFMCount?**
**POWer<m>:QUALity:RESult:POWer:PFACtor:STDDev?**

Returns the power factor for the power quality analysis.

For details on the statistics, see "Overview of statistic results" on page 1247.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:QUALity:RESult:POWer:PHASe[:ACTual]?**
**POWer<m>:QUALity:RESult:POWer:PHASe:AVERage?**
**POWer<m>:QUALity:RESult:POWer:PHASe:EVTCount?**
**POWer<m>:QUALity:RESult:POWer:PHASe:NPEak?**
**POWer<m>:QUALity:RESult:POWer:PHASe:PPEak?**
**POWer<m>:QUALity:RESult:POWer:PHASe:RMS?**
**POWer<m>:QUALity:RESult:POWer:PHASe:WFMCount?**
**POWer<m>:QUALity:RESult:POWer:PHASe:STDDev?**

Returns the phase for the power quality analysis.

For details on the statistics, see "Overview of statistic results" on page 1247.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**POWer<m>:QUALity:RESult:POWer:REACtive[:ACTual]?**
**POWer<m>:QUALity:RESult:POWer:REACtive:AVERage?**
**POWer<m>:QUALity:RESult:POWer:REACtive:EVTCount?**
**POWer<m>:QUALity:RESult:POWer:REACtive:NPEak?**
**POWer<m>:QUALity:RESult:POWer:REACtive:PPEak?**
**POWer<m>:QUALity:RESult:POWer:REACtive:RMS?**
**POWer<m>:QUALity:RESult:POWer:REACtive:WFMCount?**
**POWer<m>:QUALity:RESult:POWer:REACtive:STDDev?**

Returns the reactive power for the power quality analysis.

For details on the statistics, see "Overview of statistic results" on page 1247.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:QUALity:RESult:VOLTage:CRESTfactor[:ACTual]?**
**POWer<m>:QUALity:RESult:VOLTage:CRESTfactor:AVERage?**
**POWer<m>:QUALity:RESult:VOLTage:CRESTfactor:EVTCount?**
**POWer<m>:QUALity:RESult:VOLTage:CRESTfactor:NPEak?**
**POWer<m>:QUALity:RESult:VOLTage:CRESTfactor:PPEak?**
**POWer<m>:QUALity:RESult:VOLTage:CRESTfactor:RMS?**
**POWer<m>:QUALity:RESult:VOLTage:CRESTfactor:WFMCount?**
**POWer<m>:QUALity:RESult:VOLTage:CRESTfactor:STDDev?**

Returns the voltage crest factor, the *Peak value / RMS value* for the power quality analysis.

For details on the statistics, see "Overview of statistic results" on page 1247.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:QUALity:RESult:VOLTage:FREQuency[:ACTual]?**
**POWer<m>:QUALity:RESult:VOLTage:FREQuency:AVERage?**
**POWer<m>:QUALity:RESult:VOLTage:FREQuency:EVTCount?**
**POWer<m>:QUALity:RESult:VOLTage:FREQuency:NPEak?**

**POWer<m>:QUALity:RESult:VOLTage:FREQuency:PPEak?**
**POWer<m>:QUALity:RESult:VOLTage:FREQuency:RMS?**
**POWer<m>:QUALity:RESult:VOLTage:FREQuency:WFMCount?**
**POWer<m>:QUALity:RESult:VOLTage:FREQuency:STDDev?**

Returns the voltage frequency value for the power quality analysis.

For details on the statistics, see "Overview of statistic results" on page 1247.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:QUALity:RESult:VOLTage:PEAK[:ACTual]?**
**POWer<m>:QUALity:RESult:VOLTage:PEAK:AVERage?**
**POWer<m>:QUALity:RESult:VOLTage:PEAK:EVTCount?**
**POWer<m>:QUALity:RESult:VOLTage:PEAK:NPEak?**
**POWer<m>:QUALity:RESult:VOLTage:PEAK:PPEak?**
**POWer<m>:QUALity:RESult:VOLTage:PEAK:RMS?**
**POWer<m>:QUALity:RESult:VOLTage:PEAK:WFMCount?**
**POWer<m>:QUALity:RESult:VOLTage:PEAK:STDDev?**

Returns the voltage peak value, the highest measured magnitude value for the power quality analysis.

For details on the statistics, see "Overview of statistic results" on page 1247.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:QUALity:RESult:VOLTage:RMS[:ACTual]?**
**POWer<m>:QUALity:RESult:VOLTage:RMS:AVERage?**
**POWer<m>:QUALity:RESult:VOLTage:RMS:EVTCount?**
**POWer<m>:QUALity:RESult:VOLTage:RMS:NPEak?**
**POWer<m>:QUALity:RESult:VOLTage:RMS:PPEak?**
**POWer<m>:QUALity:RESult:VOLTage:RMS:RMS?**

**POWer<m>:QUALity:RESult:VOLTage:RMS:WFMCount?**
**POWer<m>:QUALity:RESult:VOLTage:RMS:STDDev?**

Returns the root mean square of the voltage for the power quality analysis.

For details on the statistics, see "Overview of statistic results" on page 1247.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

### 18.17.3.3 Power harmonics


- Power harmonics setup....1263
- Power harmonics results....1268


#### Power harmonics setup


POWer<m>:HARMonics:SOURce:CURRent....1263
POWer<m>:HARMonics:SOURce[:VOLTage]....1264
POWer<m>:HARMonics:STANdard....1264
POWer<m>:HARMonics:REVision....1264
POWer<m>:HARMonics:FREQuency:EN....1265
POWer<m>:HARMonics:FREQuency:MIL....1265
POWer<m>:HARMonics:FREQuency:RTCA....1265
POWer<m>:HARMonics:DISPlay:HARMonics[:DISPlay]....1266
POWer<m>:HARMonics:DISPlay:BARGraph:VERTical:SCALe:TYPE....1266
POWer<m>:HARMonics:DISPlay:HARMonics:STARt[:VALue]....1266
POWer<m>:HARMonics:DISPlay:HARMonics:STOP[:VALue]....1266
POWer<m>:HARMonics:RPOWer[:MODE]....1267
POWer<m>:HARMonics:RPOWer:USER....1267
POWer<m>:HARMonics:PFACtor:USER....1267
POWer<m>:HARMonics:PFACtor[:MODE]....1268


---

**POWer<m>:HARMonics:SOURce:CURRent** <CurrentSource>

Sets the channel for the current source input of the power harmonics analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<CurrentSource> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Current" on page 382

**POWer<m>:HARMonics:SOURce[:VOLTage]** <VoltageSource>

Sets the channel for the voltage source input of the power harmonics analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<VoltageSource> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Voltage" on page 382

**POWer<m>:HARMonics:STANdard** <Standard>

Sets a standard for the current harmonic measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Standard> ENA | ENB | ENC | END | MIL | RTCA

`ENA`: EN 61000-3-2 Class A
`ENB`: EN 61000-3-2 Class B
`ENC`: EN 61000-3-2 Class C
`END`: EN 61000-3-2 Class D
`MIL`: MIL-STD-1399
`RTCA`: RTCA DO-160

*RST: ENA

**Usage:** Asynchronous command

**Manual operation:** See "Standard" on page 382

**POWer<m>:HARMonics:REVision** <Revision>

Selects the revision of the EN61000 standard, if `POWer<m>:HARMonics:STANdard` is set to `ENA / ENB / ENC / END`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Revision> REV2011 | REV2019

*RST: REV2019

**Usage:** Asynchronous command

**Manual operation:** See "Standard" on page 382

**POWer<m>:HARMonics:FREQuency:EN** <FundFreqEN61000>

Sets the fundamental frequency of the input signal for the EN61000 standard, if `POWer<m>:HARMonics:STANdard` is set to `ENA / ENB / ENC / END`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<FundFreqEN61000> F50 | F60 | AUTO

`F50`: 50 Hz
`F60`: 60 Hz
`AUTO`: automatically set

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Fund. frequency" on page 383

**POWer<m>:HARMonics:FREQuency:MIL** <FundFreqMIL>

Sets the fundamental frequency of the input signal for the MIL standard, if `POWer<m>:HARMonics:STANdard` is set to `MIL`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<FundFreqMIL> F60 | F400

`F60`: 60 Hz
`F400`: 400 Hz

*RST: F60

**Usage:** Asynchronous command

**Manual operation:** See "Fund. frequency" on page 383

**POWer<m>:HARMonics:FREQuency:RTCA** <FundamFreqRTCA>

Sets the fundamental frequency of the input signal for the RTCA standard, if `POWer<m>:HARMonics:STANdard` is set to `RTCA`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<FundFreqRTCA> F400 | NVF | WVF

`F400`: 400 Hz
`NVF`
`WVF`

*RST: F400

**Usage:** Asynchronous command

**Manual operation:** See "Fund. frequency" on page 383

---

**POWer<m>:HARMonics:DISPlay:HARMonics[:DISPlay]** <DispedHarmonics>

Selects which harmonics are displayed in the bar graph: all, odd even or depending on the standard definition.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<DispedHarmonics> ALL | EVEN | ODD | STANdard
*RST: STANdard

**Usage:** Asynchronous command

**Manual operation:** See "Displ. harmonics" on page 384

---

**POWer<m>:HARMonics:DISPlay:BARGraph:VERTical:SCALe:TYPE** <Scaling>

Selects a logarithmic or linear scale for the display for the harmonics bar graph.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Scaling> LIN | LOG
*RST: LIN

**Usage:** Asynchronous command

**Manual operation:** See "Scale type" on page 384

---

**POWer<m>:HARMonics:DISPlay:HARMonics:STARt[:VALue]** <BarGraphStart>

Sets the first harmonic to be shown in the bar graph.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<BarGraphStart> Range: 1 to 37
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Start harmonic, Max harmonic" on page 384

---

**POWer<m>:HARMonics:DISPlay:HARMonics:STOP[:VALue]** <BarGraphStop>

Sets the last harmonic to be shown in the bar graph.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<BarGraphStop> Range: 3 to 40
Increment: 1
*RST: 40

**Usage:** Asynchronous command

**Manual operation:** See "Start harmonic, Max harmonic" on page 384

---

**POWer<m>:HARMonics:RPOWer[:MODE]** <ActPowMd>

Available if `POWer<m>:HARMonics:STANdard` is set to `END`.

Selects if the power factor is defined automatically, or a user-defined value is used (`POWer<m>:HARMonics:RPOWer:USER`).

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ActPowMd> AUTO | USER
*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Real power, Power" on page 383

---

**POWer<m>:HARMonics:RPOWer:USER** <UsrActPow>

Selects the revision of the EN61000 standard, if `POWer<m>:HARMonics:STANdard` is set to `END` and `POWer<m>:HARMonics:RPOWer:USER` is set to `USER`.

Sets a user-defined power value.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<UsrActPow> Range: 0 to 10000
Increment: 1
*RST: 100
Default unit: W

**Usage:** Asynchronous command

**Manual operation:** See "Real power, Power" on page 383

---

**POWer<m>:HARMonics:PFACtor:USER** <UsrPowFact>

Available if `POWer<m>:HARMonics:STANdard` is set to `ENC` and `POWer<m>:HARMonics:PFACtor[:MODE]` is set to `USER`.

Sets a user-defined power factor.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<UsrPowFact>
Range: 0 to 100
Increment: 1
*RST: 100
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Power factor" on page 383

---

**POWer<m>:HARMonics:PFACtor[:MODE]** <PowFactMd>

Available if `POWer<m>:HARMonics:STANdard` is set to `ENC`.

Selects if the power factor is defined automatically, or a user-defined value is used (`POWer<m>:HARMonics:PFACtor:USER`).

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<PowFactMd>
AUTO | USER
*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Power factor" on page 383

### Power harmonics results

POWer<m>:HARMonics:AVAilable? 1268
POWer<m>:HARMonics:MEASurement:FREQuency[:ACTual]? 1269
POWer<m>:HARMonics:MEASurement:POWer:ACTive[:ACTual]? 1269
POWer<m>:HARMonics:MEASurement:THDFundament[:ACTual]? 1269
POWer<m>:HARMonics:MEASurement:THDRms[:ACTual]? 1269
POWer<m>:HARMonics:RESult<n>:VIOLation? 1270
POWer<m>:HARMonics:RESult<n>[:FREQuency]? 1270
POWer<m>:HARMonics:RESult<n>:MAGNitude:PERCent:MAXimum? 1270
POWer<m>:HARMonics:RESult<n>:MAGNitude:PERCent:MINimum? 1270
POWer<m>:HARMonics:RESult<n>:MAGNitude:PERCent[:ACTual]? 1271
POWer<m>:HARMonics:RESult<n>:MAGNitude:RMS:MAXimum? 1271
POWer<m>:HARMonics:RESult<n>:MAGNitude:RMS:MINimum? 1271
POWer<m>:HARMonics:RESult<n>:MAGNitude:RMS[:ACTual]? 1272

---

**POWer<m>:HARMonics:AVAilable?**

Returns the number of measured harmonics.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**POWer<m>:HARMonics:MEASurement:FREQuency[:ACTual]?**

Returns the frequency of measured harmonics.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:HARMonics:MEASurement:POWer:ACTive[:ACTual]?**

Returns the active power. Only available, if `POWer<m>:HARMonics:STANdard` on page 1264 is set to `ENC / END`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Usage:** Query only
Asynchronous command

---

**POWer<m>:HARMonics:MEASurement:THDFundament[:ACTual]?**

Returns the total harmonic distortion relative to fundamental (THDF).

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Usage:** Query only
Asynchronous command

---

**POWer<m>:HARMonics:MEASurement:THDRms[:ACTual]?**

Returns the total harmonic distortion relative to the RMS amplitude (THD RMS).

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Usage:** Query only
Asynchronous command

---

**POWer<m>:HARMonics:RESult<n>:VIOLation?**

Queries if the value of n-th harmonic is within the defined limit for the current measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...n, index of the harmonics

**Return values:**
<Harmonic>

**Usage:** Query only
Asynchronous command

---

**POWer<m>:HARMonics:RESult<n>[:FREQuency]?**

Returns the frequency of the n-th harmonic.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...n, index of the harmonics

**Return values:**
<Harmonic>

**Usage:** Query only
Asynchronous command

---

**POWer<m>:HARMonics:RESult<n>:MAGNitude:PERCent:MAXimum?**

Returns the maximum magnitude of the n-th harmonic in percent.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...n, index of the harmonics

**Return values:**
<Harmonic>

**Usage:** Query only
Asynchronous command

---

**POWer<m>:HARMonics:RESult<n>:MAGNitude:PERCent:MINimum?**

Returns the minimum magnitude of the n-th harmonic in percent.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...n, index of the harmonics

**Return values:**
<Harmonic>

**Usage:** Query only
Asynchronous command

---

**POWer<m>:HARMonics:RESult<n>:MAGNitude:PERCent[:ACTual]?**

Returns the magnitude of the n-th harmonic in percent.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...n, index of the harmonics

**Return values:**
<Harmonic>

**Usage:** Query only
Asynchronous command

---

**POWer<m>:HARMonics:RESult<n>:MAGNitude:RMS:MAXimum?**

Returns the maximum magnitude of the n-th harmonic in RMS.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...n, index of the harmonics

**Return values:**
<Harmonic>

**Usage:** Query only
Asynchronous command

---

**POWer<m>:HARMonics:RESult<n>:MAGNitude:RMS:MINimum?**

Returns the minimum magnitude of the n-th harmonic in RMS.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...n, index of the harmonics

**Return values:**
<Harmonic>

**Usage:** Query only
Asynchronous command

**POWer<m>:HARMonics:RESult<n>:MAGNitude:RMS[:ACTual]?**

Returns the magnitude of the n-th harmonic in RMS.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...n, index of the harmonics

**Return values:**
<Harmonic>

**Usage:** Query only
Asynchronous command

### 18.17.3.4 Switching loss


- Switching loss setup....1272
- Switching loss display....1275
- Switching loss results....1277


#### Switching loss setup


POWer<m>:SWITching:SOURce:CURRent....1272
POWer<m>:SWITching:SOURce[:VOLTage]....1273
POWer<m>:SWITching:DISPlay:TYPE....1273
POWer<m>:SWITching:REGion:TOTal[:STARt]....1273
POWer<m>:SWITching:REGion:TON[:STARt]....1273
POWer<m>:SWITching:REGion:TON:STOP....1273
POWer<m>:SWITching:REGion:CONDuction[:STARt]....1273
POWer<m>:SWITching:REGion:CONDuction:STOP....1274
POWer<m>:SWITching:REGion:TOFF[:STARt]....1274
POWer<m>:SWITching:REGion:TOFF:STOP....1274
POWer<m>:SWITching:REGion:NCONduction[:STARt]....1274
POWer<m>:SWITching:REGion:NCONduction:STOP....1275
POWer<m>:SWITching:REGion:TOTal:STOP....1275


**POWer<m>:SWITching:SOURce:CURRent** <SourceCurrent>

Selects the current source waveform. Analog channels and math waveforms can be used.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<SourceCurrent> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Current" on page 389

**POWer<m>:SWITching:SOURce[:VOLTage]** <SourceVoltage>

Selects the voltage source waveform. Analog channels and math waveforms can be used.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<SourceVoltage> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Voltage" on page 389

**POWer<m>:SWITching:DISPlay:TYPE** <Unit>

Selects the measurement type: power or energy.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Unit> POWer | ENERgy
*RST: POWer

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 389

**POWer<m>:SWITching:REGion:TOTal[:STARt]** <T1Position>
**POWer<m>:SWITching:REGion:TON[:STARt]** <T1Position>

Sets the start time for the turn on area in relation to the trigger point. This value is also the start time of the total switching cycle.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<T1Position> Range: -500 to 500
Increment: 1E-09
*RST: -1E-06
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "T<n> position" on page 389

**POWer<m>:SWITching:REGion:TON:STOP** <T2Position>
**POWer<m>:SWITching:REGion:CONDuction[:STARt]** <T2Position>

Sets the start time for the conduction area in relation to the trigger point. This value is also end time of the turn on area.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<T2Position>
Range: -500 to 500
Increment: 1E-09
*RST: 1E-06
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "T<n> position" on page 389

---

**POWer<m>:SWITching:REGion:CONDuction:STOP** <T3Position>
**POWer<m>:SWITching:REGion:TOFF[:STARt]** <T3Position>

Sets the start time for the turn off area in relation to the trigger point. This value is also the end time of the conduction area.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<T3Position>
Range: -500 to 500
Increment: 1E-09
*RST: 5E-06
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "T<n> position" on page 389

---

**POWer<m>:SWITching:REGion:TOFF:STOP** <T4Position>
**POWer<m>:SWITching:REGion:NCONduction[:STARt]** <T4Position>

Sets the start time for the non-conduction area in relation to the trigger point. This value is also the end time of the turn off area.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<T4Position>
Range: -500 to 500
Increment: 1E-09
*RST: 6E-06
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "T<n> position" on page 389

**POWer<m>:SWITching:REGion:NCONduction:STOP** <T5Position>
**POWer<m>:SWITching:REGion:TOTal:STOP** <T5Position>

Sets the end time for the non-conduction area in relation to the trigger point. This value is also the end of the total switching cycle.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<T5Position>
Range: -500 to 500
Increment: 1E-09
*RST: 1E-05
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "T<n> position" on page 389

### Switching loss display

POWer<m>:SWITching:DISPlay:CONDuction[:ENABle] 1275
POWer<m>:SWITching:DISPlay:NCONduction[:ENABle] 1275
POWer<m>:SWITching:DISPlay:TOFF[:ENABle] 1275
POWer<m>:SWITching:DISPlay:TON[:ENABle] 1275
POWer<m>:SWITching:DISPlay:CONDuction:COLor 1276
POWer<m>:SWITching:DISPlay:NCONduction:COLor 1276
POWer<m>:SWITching:DISPlay:TOFF:COLor 1276
POWer<m>:SWITching:DISPlay:TON:COLor 1276
POWer<m>:SWITching:DISPlay:TOTal[:ENABle] 1276
POWer<m>:SWITching:DISPlay:LABel[:ENABle] 1276

**POWer<m>:SWITching:DISPlay:CONDuction[:ENABle]** <Show>
**POWer<m>:SWITching:DISPlay:NCONduction[:ENABle]** <Show>
**POWer<m>:SWITching:DISPlay:TOFF[:ENABle]** <Show>
**POWer<m>:SWITching:DISPlay:TON[:ENABle]** <Show>

The commands enable the measurements of the conduction area, non-conduction area, turn off area and turn on area, respectively. Results of enabled mesurements are shown in the result table.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Show> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Turn on" on page 390

**POWer<m>:SWITching:DISPlay:CONDuction:COLor** <Color>
**POWer<m>:SWITching:DISPlay:NCONduction:COLor** <Color>
**POWer<m>:SWITching:DISPlay:TOFF:COLor** <Color>
**POWer<m>:SWITching:DISPlay:TON:COLor** <Color>

The commands set the display color of the conduction area, non-conduction area, turn off area and turn on area, respectively.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Color> Decimal value of the ARGB color. Use the color dialog on the instrument to get the hex value of the color, and convert the hex value to a decimal value.
0 is fully transparent black.
4278190080 (dec) = FF000000 (hex) is opaque black.
4294967295 (dec) = FFFFFFFF (hex) is opaque white.

Range: 0 to 4294967295
Increment: 1

**Usage:** Asynchronous command

**Manual operation:** See "Turn on" on page 390

Reset values:

| | |
|---|---|
| Conduction | 2030010343 |
| Non-conduction | 2023358259 |
| Turn off | 2030002176 |
| Turn on | 2026077439 |

**POWer<m>:SWITching:DISPlay:TOTal[:ENABle]** <ShowTotal>

Enables the measurements of the total switching cycle.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShowTotal> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Total" on page 391

**POWer<m>:SWITching:DISPlay:LABel[:ENABle]** <ShowLabel>

Displays the names of the switching cycle areas.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ShowLabel> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Area label" on page 391

### Switching loss results

See "Overview of statistic results" on page 1247.

POWer<m>:SWITching:RESult:CONDuction:ENERgy[:ACTual]? 1278
POWer<m>:SWITching:RESult:CONDuction:POWer[:ACTual]? 1278
POWer<m>:SWITching:RESult:NCONduction:ENERgy[:ACTual]? 1278
POWer<m>:SWITching:RESult:NCONduction:POWer[:ACTual]? 1278
POWer<m>:SWITching:RESult:TOFF:ENERgy[:ACTual]? 1278
POWer<m>:SWITching:RESult:TOFF:POWer[:ACTual]? 1279
POWer<m>:SWITching:RESult:TON:ENERgy[:ACTual]? 1279
POWer<m>:SWITching:RESult:TON:POWer[:ACTual]? 1279
POWer<m>:SWITching:RESult:TOTal:ENERgy[:ACTual]? 1279
POWer<m>:SWITching:RESult:TOTal:POWer[:ACTual]? 1279
POWer<m>:SWITching:RESult:CONDuction:ENERgy:AVERage? 1279
POWer<m>:SWITching:RESult:CONDuction:POWer:AVERage? 1279
POWer<m>:SWITching:RESult:NCONduction:ENERgy:AVERage? 1279
POWer<m>:SWITching:RESult:NCONduction:POWer:AVERage? 1279
POWer<m>:SWITching:RESult:TOFF:ENERgy:AVERage? 1279
POWer<m>:SWITching:RESult:TOFF:POWer:AVERage? 1279
POWer<m>:SWITching:RESult:TON:ENERgy:AVERage? 1279
POWer<m>:SWITching:RESult:TON:POWer:AVERage? 1279
POWer<m>:SWITching:RESult:TOTal:ENERgy:AVERage? 1279
POWer<m>:SWITching:RESult:TOTal:POWer:AVERage? 1279
POWer<m>:SWITching:RESult:CONDuction:ENERgy:EVTCount? 1279
POWer<m>:SWITching:RESult:CONDuction:POWer:EVTCount? 1279
POWer<m>:SWITching:RESult:NCONduction:ENERgy:EVTCount? 1279
POWer<m>:SWITching:RESult:NCONduction:POWer:EVTCount? 1279
POWer<m>:SWITching:RESult:TOFF:ENERgy:EVTCount? 1279
POWer<m>:SWITching:RESult:TOFF:POWer:EVTCount? 1279
POWer<m>:SWITching:RESult:TON:ENERgy:EVTCount? 1279
POWer<m>:SWITching:RESult:TON:POWer:EVTCount? 1279
POWer<m>:SWITching:RESult:TOTal:ENERgy:EVTCount? 1280
POWer<m>:SWITching:RESult:TOTal:POWer:EVTCount? 1280
POWer<m>:SWITching:RESult:CONDuction:ENERgy:NPEak? 1280
POWer<m>:SWITching:RESult:CONDuction:POWer:NPEak? 1280
POWer<m>:SWITching:RESult:NCONduction:ENERgy:NPEak? 1280
POWer<m>:SWITching:RESult:NCONduction:POWer:NPEak? 1280
POWer<m>:SWITching:RESult:TOFF:ENERgy:NPEak? 1280
POWer<m>:SWITching:RESult:TOFF:POWer:NPEak? 1280
POWer<m>:SWITching:RESult:TON:ENERgy:NPEak? 1280

POWer<m>:SWITching:RESult:TON:POWer:NPEak? 1280
POWer<m>:SWITching:RESult:TOTal:ENERgy:NPEak? 1280
POWer<m>:SWITching:RESult:TOTal:POWer:NPEak? 1280
POWer<m>:SWITching:RESult:CONDuction:ENERgy:PPEak? 1280
POWer<m>:SWITching:RESult:CONDuction:POWer:PPEak? 1280
POWer<m>:SWITching:RESult:NCONduction:ENERgy:PPEak? 1280
POWer<m>:SWITching:RESult:NCONduction:POWer:PPEak? 1280
POWer<m>:SWITching:RESult:TOFF:ENERgy:PPEak? 1280
POWer<m>:SWITching:RESult:TOFF:POWer:PPEak? 1280
POWer<m>:SWITching:RESult:TON:ENERgy:PPEak? 1280
POWer<m>:SWITching:RESult:TON:POWer:PPEak? 1280
POWer<m>:SWITching:RESult:TOTal:ENERgy:PPEak? 1281
POWer<m>:SWITching:RESult:TOTal:POWer:PPEak? 1281
POWer<m>:SWITching:RESult:CONDuction:ENERgy:RMS? 1281
POWer<m>:SWITching:RESult:CONDuction:POWer:RMS? 1281
POWer<m>:SWITching:RESult:NCONduction:ENERgy:RMS? 1281
POWer<m>:SWITching:RESult:NCONduction:POWer:RMS? 1281
POWer<m>:SWITching:RESult:TOFF:ENERgy:RMS? 1281
POWer<m>:SWITching:RESult:TOFF:POWer:RMS? 1281
POWer<m>:SWITching:RESult:TON:ENERgy:RMS? 1281
POWer<m>:SWITching:RESult:TON:POWer:RMS? 1281
POWer<m>:SWITching:RESult:TOTal:ENERgy:RMS? 1281
POWer<m>:SWITching:RESult:TOTal:POWer:RMS? 1281
POWer<m>:SWITching:RESult:CONDuction:ENERgy:STDDev? 1281
POWer<m>:SWITching:RESult:CONDuction:POWer:STDDev? 1281
POWer<m>:SWITching:RESult:NCONduction:ENERgy:STDDev? 1281
POWer<m>:SWITching:RESult:NCONduction:POWer:STDDev? 1281
POWer<m>:SWITching:RESult:TOFF:ENERgy:STDDev? 1281
POWer<m>:SWITching:RESult:TOFF:POWer:STDDev? 1281
POWer<m>:SWITching:RESult:TON:ENERgy:STDDev? 1281
POWer<m>:SWITching:RESult:TON:POWer:STDDev? 1281
POWer<m>:SWITching:RESult:TOTal:ENERgy:STDDev? 1282
POWer<m>:SWITching:RESult:TOTal:POWer:STDDev? 1282
POWer<m>:SWITching:RESult:CONDuction:ENERgy:WFMCount? 1282
POWer<m>:SWITching:RESult:CONDuction:POWer:WFMCount? 1282
POWer<m>:SWITching:RESult:NCONduction:ENERgy:WFMCount? 1282
POWer<m>:SWITching:RESult:NCONduction:POWer:WFMCount? 1282
POWer<m>:SWITching:RESult:TOFF:ENERgy:WFMCount? 1282
POWer<m>:SWITching:RESult:TOFF:POWer:WFMCount? 1282
POWer<m>:SWITching:RESult:TON:ENERgy:WFMCount? 1282
POWer<m>:SWITching:RESult:TON:POWer:WFMCount? 1282
POWer<m>:SWITching:RESult:TOTal:ENERgy:WFMCount? 1282
POWer<m>:SWITching:RESult:TOTal:POWer:WFMCount? 1282

**POWer<m>:SWITching:RESult:CONDuction:ENERgy[:ACTual]?**
**POWer<m>:SWITching:RESult:CONDuction:POWer[:ACTual]?**
**POWer<m>:SWITching:RESult:NCONduction:ENERgy[:ACTual]?**
**POWer<m>:SWITching:RESult:NCONduction:POWer[:ACTual]?**
**POWer<m>:SWITching:RESult:TOFF:ENERgy[:ACTual]?**

**POWer<m>:SWITching:RESult:TOFF:POWer[:ACTual]?**
**POWer<m>:SWITching:RESult:TON:ENERgy[:ACTual]?**
**POWer<m>:SWITching:RESult:TON:POWer[:ACTual]?**
**POWer<m>:SWITching:RESult:TOTal:ENERgy[:ACTual]?**
**POWer<m>:SWITching:RESult:TOTal:POWer[:ACTual]?**

Return the current result value of the selected measurement type.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:SWITching:RESult:CONDuction:ENERgy:AVERage?**
**POWer<m>:SWITching:RESult:CONDuction:POWer:AVERage?**
**POWer<m>:SWITching:RESult:NCONduction:ENERgy:AVERage?**
**POWer<m>:SWITching:RESult:NCONduction:POWer:AVERage?**
**POWer<m>:SWITching:RESult:TOFF:ENERgy:AVERage?**
**POWer<m>:SWITching:RESult:TOFF:POWer:AVERage?**
**POWer<m>:SWITching:RESult:TON:ENERgy:AVERage?**
**POWer<m>:SWITching:RESult:TON:POWer:AVERage?**
**POWer<m>:SWITching:RESult:TOTal:ENERgy:AVERage?**
**POWer<m>:SWITching:RESult:TOTal:POWer:AVERage?**

Return the average value of the selected measurement type if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<Average> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:SWITching:RESult:CONDuction:ENERgy:EVTCount?**
**POWer<m>:SWITching:RESult:CONDuction:POWer:EVTCount?**
**POWer<m>:SWITching:RESult:NCONduction:ENERgy:EVTCount?**
**POWer<m>:SWITching:RESult:NCONduction:POWer:EVTCount?**
**POWer<m>:SWITching:RESult:TOFF:ENERgy:EVTCount?**
**POWer<m>:SWITching:RESult:TOFF:POWer:EVTCount?**
**POWer<m>:SWITching:RESult:TON:ENERgy:EVTCount?**
**POWer<m>:SWITching:RESult:TON:POWer:EVTCount?**

**POWer<m>:SWITching:RESult:TOTal:ENERgy:EVTCount?**
**POWer<m>:SWITching:RESult:TOTal:POWer:EVTCount?**

Return the number of calculated measurement results of the selected measurement type if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<Count> Range: 0 to 4000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:SWITching:RESult:CONDuction:ENERgy:NPEak?**
**POWer<m>:SWITching:RESult:CONDuction:POWer:NPEak?**
**POWer<m>:SWITching:RESult:NCONduction:ENERgy:NPEak?**
**POWer<m>:SWITching:RESult:NCONduction:POWer:NPEak?**
**POWer<m>:SWITching:RESult:TOFF:ENERgy:NPEak?**
**POWer<m>:SWITching:RESult:TOFF:POWer:NPEak?**
**POWer<m>:SWITching:RESult:TON:ENERgy:NPEak?**
**POWer<m>:SWITching:RESult:TON:POWer:NPEak?**
**POWer<m>:SWITching:RESult:TOTal:ENERgy:NPEak?**
**POWer<m>:SWITching:RESult:TOTal:POWer:NPEak?**

Return the negative peak value (minimum) of the selected measurement type if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<MinPeak> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:SWITching:RESult:CONDuction:ENERgy:PPEak?**
**POWer<m>:SWITching:RESult:CONDuction:POWer:PPEak?**
**POWer<m>:SWITching:RESult:NCONduction:ENERgy:PPEak?**
**POWer<m>:SWITching:RESult:NCONduction:POWer:PPEak?**
**POWer<m>:SWITching:RESult:TOFF:ENERgy:PPEak?**
**POWer<m>:SWITching:RESult:TOFF:POWer:PPEak?**
**POWer<m>:SWITching:RESult:TON:ENERgy:PPEak?**
**POWer<m>:SWITching:RESult:TON:POWer:PPEak?**

**POWer<m>:SWITching:RESult:TOTal:ENERgy:PPEak?**
**POWer<m>:SWITching:RESult:TOTal:POWer:PPEak?**

Return the positive peak value (maximum) of the selected measurement type if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<MaxPeak> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:SWITching:RESult:CONDuction:ENERgy:RMS?**
**POWer<m>:SWITching:RESult:CONDuction:POWer:RMS?**
**POWer<m>:SWITching:RESult:NCONduction:ENERgy:RMS?**
**POWer<m>:SWITching:RESult:NCONduction:POWer:RMS?**
**POWer<m>:SWITching:RESult:TOFF:ENERgy:RMS?**
**POWer<m>:SWITching:RESult:TOFF:POWer:RMS?**
**POWer<m>:SWITching:RESult:TON:ENERgy:RMS?**
**POWer<m>:SWITching:RESult:TON:POWer:RMS?**
**POWer<m>:SWITching:RESult:TOTal:ENERgy:RMS?**
**POWer<m>:SWITching:RESult:TOTal:POWer:RMS?**

Return the RMS value of the selected measurement type if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<RMS> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:SWITching:RESult:CONDuction:ENERgy:STDDev?**
**POWer<m>:SWITching:RESult:CONDuction:POWer:STDDev?**
**POWer<m>:SWITching:RESult:NCONduction:ENERgy:STDDev?**
**POWer<m>:SWITching:RESult:NCONduction:POWer:STDDev?**
**POWer<m>:SWITching:RESult:TOFF:ENERgy:STDDev?**
**POWer<m>:SWITching:RESult:TOFF:POWer:STDDev?**
**POWer<m>:SWITching:RESult:TON:ENERgy:STDDev?**
**POWer<m>:SWITching:RESult:TON:POWer:STDDev?**

**POWer<m>:SWITching:RESult:TOTal:ENERgy:STDDev?**
**POWer<m>:SWITching:RESult:TOTal:POWer:STDDev?**

Return the standard deviation of the selected measurement type if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:SWITching:RESult:CONDuction:ENERgy:WFMCount?**
**POWer<m>:SWITching:RESult:CONDuction:POWer:WFMCount?**
**POWer<m>:SWITching:RESult:NCONduction:ENERgy:WFMCount?**
**POWer<m>:SWITching:RESult:NCONduction:POWer:WFMCount?**
**POWer<m>:SWITching:RESult:TOFF:ENERgy:WFMCount?**
**POWer<m>:SWITching:RESult:TOFF:POWer:WFMCount?**
**POWer<m>:SWITching:RESult:TON:ENERgy:WFMCount?**
**POWer<m>:SWITching:RESult:TON:POWer:WFMCount?**
**POWer<m>:SWITching:RESult:TOTal:ENERgy:WFMCount?**
**POWer<m>:SWITching:RESult:TOTal:POWer:WFMCount?**

Return the number of calculated waveforms of the selected measurement type if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<WaveformsCount> Range: 0 to 4000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

### 18.17.3.5 Turn on/off time


- Turn on/off setup ..... 1282
- Turn on/off display ..... 1285
- Turn on/off results ..... 1286


#### Turn on/off setup


POWer<m>:ONOFf:INPut:AC:ABSolute[:VALue] ..... 1283
POWer<m>:ONOFf:INPut:DC:ABSolute[:VALue] ..... 1283
POWer<m>:ONOFf:INPut:TYPE ..... 1283

POWer<m>:ONOFf:INPut[:SOURce]....1284
POWer<m>:ONOFf:ONUMber....1284
POWer<m>:ONOFf:OUTPut<n>:DC:ABSolute[:VALue]....1284
POWer<m>:ONOFf:OUTPut<n>:DISPlay:RESult[:ENABle]....1284
POWer<m>:ONOFf:OUTPut<n>[:SOURce]....1285
POWer<m>:ONOFf[:TYPE]....1285

**POWer<m>:ONOFf:INPut:AC:ABSolute[:VALue]** <InputACThreshold>

Sets the threshold for the AC input signal.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<InputACThreshold> Range: 0 to 1E+26
Increment: 0.001
*RST: 1
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "AC threshold" on page 396

**POWer<m>:ONOFf:INPut:DC:ABSolute[:VALue]** <InputDCThreshold>

Sets the threshold for the DC input signal.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<InputDCThreshold> Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 1
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "DC threshold" on page 395

**POWer<m>:ONOFf:INPut:TYPE** <InputType>

Selects whether the input signal is AC or CD.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<InputType> DC | AC
*RST: DC

**Usage:** Asynchronous command

**Manual operation:** See "Input type" on page 395

**POWer<m>:ONOFf:INPut[:SOURce]** <InputSource>

Selects the source waveform of the input signal.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<InputSource> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Input source" on page 395

**POWer<m>:ONOFf:ONUMber** <OutputCount>

Sets the number of outputs to be used in the measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<OutputCount>

**Usage:** Asynchronous command

**POWer<m>:ONOFf:OUTPut<n>:DC:ABSolute[:VALue]** <OutDCThreshold>

Sets the threshold for the selected output signal.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1..3, index of the output signal (4-channel instrument).
1..7, index of the output signal (8-channel instrument).

**Parameters:**
<OutDCThreshold> Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 1
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "DC threshold" on page 396

**POWer<m>:ONOFf:OUTPut<n>:DISPlay:RESult[:ENABle]** <Enable>

Activates the indicated output line.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1..3, index of the output signal (4-channel instrument).
1..7, index of the output signal (8-channel instrument).

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable output" on page 396

---

**POWer<m>:ONOFf:OUTPut<n>[:SOURce]** <OutputSource>

Selects the channel of the output signal. All analog channels except for the input channel can be used. Each channel can be used only once.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1..3, index of the output signal (4-channel instrument).
1..7, index of the output signal (8-channel instrument).

**Parameters:**
<OutputSource> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Example:** `POW2:ONOF:OUTP3 C4`
Sets the 3rd output to channel 4.

**Usage:** Asynchronous command

**Manual operation:** See "Output source" on page 396

---

**POWer<m>:ONOFf[:TYPE]** <Mode>

Selects the turn on or turn off measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Mode> TON | TOFF
*RST: TON

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 395

### Turn on/off display

---

**POWer<m>:ONOFf:OUTPut<n>:DISPlay:RLINe[:ENABle]** <Display>

Enables the result lines for the selected output.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1..3, index of the output signal (4-channel instrument).
1..7, index of the output signal (8-channel instrument).

**Parameters:**

<Display> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Result line <n>" on page 397

### Turn on/off results

**POWer<m>:ONOFf:RESult<n>:TIME[:ACTual]?**

Returns the measured turn-on time or turn-off time of the specified input-output pair.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> 1..*
Index of the output signal.

**Return values:**

<Time> Range: -1E+26 to 1E+26
Increment: 0.1
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

## 18.17.3.6 Power efficiency


- Efficiency setup....1286
- Efficiency gate....1289
- Efficiency display....1289
- Efficiency results....1290


### Efficiency setup


POWer<m>:EFFiciency:INPut<n>:CURRent[:SOURce]....1286
POWer<m>:EFFiciency:INPut<n>:VOLTage[:SOURce]....1287
POWer<m>:EFFiciency:INPut<n>[:TYPE]....1287
POWer<m>:EFFiciency:ONUMber?....1287
POWer<m>:EFFiciency:OUTPut<n>:CURRent[:SOURce]....1288
POWer<m>:EFFiciency:OUTPut<n>:VOLTage[:SOURce]....1288
POWer<m>:EFFiciency:OUTPut<n>[:TYPE]....1288


**POWer<m>:EFFiciency:INPut<n>:CURRent[:SOURce]** <SourceCurrent>

Selects the current source waveform of the input line.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Parameters:**
<SourceCurrent> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Current (Input)" on page 400

---

**POWer<m>:EFFiciency:INPut<n>:VOLTage[:SOURce]** <SourceVoltage>

Selects the voltage source waveform of the input line.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Parameters:**
<SourceVoltage> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Voltage (Input)" on page 400

---

**POWer<m>:EFFiciency:INPut<n>[:TYPE]** <Type>

Selects the type of the current flow: AC or DC.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Parameters:**
<Type> DC | AC
*RST: DC

**Usage:** Asynchronous command

**Manual operation:** See "Input type" on page 400

---

**POWer<m>:EFFiciency:ONUMber?** [MAX]

Returns the number of available outputs.

You can query the maximum value with `<command>? MAX`.

The maximum number of output lines is *Number of channels / 2 - 1*.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<OutputCount> Number of outputs

**Usage:** Query only
Asynchronous command

**POWer<m>:EFFiciency:OUTPut<n>:CURRent[:SOURce]** <SourceCurrent>

Selects the current source waveform of the selected output line.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1..3, output index
Suffix >1 requires an instrument with more than 4 channels.

**Parameters:**
<SourceCurrent> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Current (Output)" on page 401

**POWer<m>:EFFiciency:OUTPut<n>:VOLTage[:SOURce]** <SourceVoltage>

Selects the voltage source waveform of the selected output line.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1..3, output index
Suffix >1 requires an instrument with more than 4 channels.

**Parameters:**
<SourceVoltage> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Voltage (Output)" on page 401

**POWer<m>:EFFiciency:OUTPut<n>[:TYPE]** <Type>

Selects the type of the current flow: AC or DC.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1..3, output index
Suffix >1 requires an instrument with more than 4 channels.

**Parameters:**
<Type> DC | AC

*RST: DC

**Usage:** Asynchronous command

**Manual operation:** See "Output type" on page 401

### Efficiency gate

**POWer<m>:EFFiciency:GATE** <Gate>

Sets the gate for the specified efficiency analysis. Configure the gate before you can assign it. Make sure that the suffix matches the power effciency measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Gate> 0 to 8, index of the assigned gate. The value 0 indicates that no gate is assigned.

**Usage:** Asynchronous command

### Efficiency display

POWer<m>:EFFiciency:DISPlay:RESult:OUTPut<n>:EFFiciency[:ENABle] ..... 1289
POWer<m>:EFFiciency:DISPlay:RESult:OUTPut<n>:POWer[:ENABle] ..... 1289
POWer<m>:EFFiciency:DISPlay:RESult:TOTal<n>:EFFiciency[:ENABle] ..... 1289
POWer<m>:EFFiciency:DISPlay:RESult:TOTal<n>:OPOWer[:ENABle] ..... 1289

**POWer<m>:EFFiciency:DISPlay:RESult:OUTPut<n>:EFFiciency[:ENABle]** <Show>
**POWer<m>:EFFiciency:DISPlay:RESult:OUTPut<n>:POWer[:ENABle]** <Show>

The commands enable the power and efficiency measurements of the selected output line.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1..3, output index
Suffix >1 requires an instrument with more than 4 channels.

**Parameters:**
<Show> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**POWer<m>:EFFiciency:DISPlay:RESult:TOTal<n>:EFFiciency[:ENABle]** <Show>
**POWer<m>:EFFiciency:DISPlay:RESult:TOTal<n>:OPOWer[:ENABle]** <Show>

The commands enable the total power and efficiency measurements of the selected output line. These measurements require an instrument with more than 4 channels.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Parameters:**

<Show> OFF | ON

*RST: ON

**Usage:** Asynchronous command

### Efficiency results

See "Overview of statistic results" on page 1247.

POWer<m>:EFFiciency:RESult:INPut:POWer[:ACTual]? 1290
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency[:ACTual]? 1290
POWer<m>:EFFiciency:RESult:TOTal:OPOWer[:ACTual]? 1290
POWer<m>:EFFiciency:RESult:INPut:POWer:AVERage? 1291
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:AVERage? 1291
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:AVERage? 1291
POWer<m>:EFFiciency:RESult:INPut:POWer:EVTCount? 1291
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:EVTCount? 1291
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:EVTCount? 1291
POWer<m>:EFFiciency:RESult:INPut:POWer:NPEak? 1291
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:NPEak? 1291
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:NPEak? 1291
POWer<m>:EFFiciency:RESult:INPut:POWer:PPEak? 1292
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:PPEak? 1292
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:PPEak? 1292
POWer<m>:EFFiciency:RESult:INPut:POWer:RMS? 1292
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:RMS? 1292
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:RMS? 1292
POWer<m>:EFFiciency:RESult:INPut:POWer:STDDev? 1292
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:STDDev? 1292
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:STDDev? 1292
POWer<m>:EFFiciency:RESult:INPut:POWer:WFMCount? 1293
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:WFMCount? 1293
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:WFMCount? 1293

---

**POWer<m>:EFFiciency:RESult:INPut:POWer[:ACTual]?**
**POWer<m>:EFFiciency:RESult:TOTal:EFFiciency[:ACTual]?**
**POWer<m>:EFFiciency:RESult:TOTal:OPOWer[:ACTual]?**

Return the current result value for input power, total efficiency and total output power.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

**Return values:**

<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**POWer<m>:EFFiciency:RESult:INPut:POWer:AVERage?**
**POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:AVERage?**
**POWer<m>:EFFiciency:RESult:TOTal:OPOWer:AVERage?**

Return the average value for input power, total efficiency and total output power if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<Average> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**POWer<m>:EFFiciency:RESult:INPut:POWer:EVTCount?**
**POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:EVTCount?**
**POWer<m>:EFFiciency:RESult:TOTal:OPOWer:EVTCount?**

Return the number of calculated measurement results for input power, total efficiency and total output power if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<Count> Range: 0 to 4000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**POWer<m>:EFFiciency:RESult:INPut:POWer:NPEak?**
**POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:NPEak?**
**POWer<m>:EFFiciency:RESult:TOTal:OPOWer:NPEak?**

Return the negative peak value (minimum) for input power, total efficiency and total output power if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<MinPeak> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**POWer<m>:EFFiciency:RESult:INPut:POWer:PPEak?**
**POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:PPEak?**
**POWer<m>:EFFiciency:RESult:TOTal:OPOWer:PPEak?**

Return the npositive peak value (maximum) for input power, total efficiency and total output power if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<MaxPeak> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**POWer<m>:EFFiciency:RESult:INPut:POWer:RMS?**
**POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:RMS?**
**POWer<m>:EFFiciency:RESult:TOTal:OPOWer:RMS?**

Return the RMS value for input power, total efficiency and total output power if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<RMS> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**POWer<m>:EFFiciency:RESult:INPut:POWer:STDDev?**
**POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:STDDev?**
**POWer<m>:EFFiciency:RESult:TOTal:OPOWer:STDDev?**

Return the standard deviation for input power, total efficiency and total output power if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**POWer<m>:EFFiciency:RESult:INPut:POWer:WFMCount?**
**POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:WFMCount?**
**POWer<m>:EFFiciency:RESult:TOTal:OPOWer:WFMCount?**

Return the number of calculated waveforms for input power, total efficiency and total output power if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<WaveformsCount> Range: 0 to 4000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

### 18.17.3.7 Safe operating area (SOA)


- SOA setup....1293
- SOA mask....1294
- SOA actions....1300
- SOA results....1302


#### SOA setup

**POWer<m>:SOA:MTESt<n>:IMEXport:NAME** <Name>

Sets the path, the filename and the file format of the mask file.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Parameters:**
<Name> String with path and file name with extension `.xml`.

**Usage:** Asynchronous command

**Manual operation:** See "Save mask" on page 406

**POWer<m>:SOA:MTESt<n>:IMEXport:OPEN**

Opens and loads the mask selected by `POWer<m>:SOA:MTESt<n>:IMEXport:NAME`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Open" on page 407

---

**POWer<m>:SOA:MTESt<n>:IMEXport:SAVE**

Saves the mask test to the file selected by `POWer<m>:SOA:MTESt<n>:IMEXport:NAME`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save mask" on page 406

---

**POWer<m>:SOA:SOURce:CURRent** <SourceCurrent>

Selects the current source waveform. Analog channels and math waveforms can be used.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<SourceCurrent> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Current" on page 406

---

**POWer<m>:SOA:SOURce:VOLTage** <SourceVoltage>

Selects the voltage source waveform. Analog channels and math waveforms can be used.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<SourceVoltage> C1 | C2 | ... | C8 | M1 | M2 | ... | M8

**Usage:** Asynchronous command

**Manual operation:** See "Voltage" on page 406

### SOA mask

POWer<m>:SOA:LIMit:APPLy ..... 1295
POWer<m>:SOA:LIMit:IMAX ..... 1295
POWer<m>:SOA:LIMit:PMAX ..... 1295
POWer<m>:SOA:LIMit:VMAX ..... 1296
POWer<m>:SOA:MTESt<n>:SEGMent<o>:ADD ..... 1296

POWer<m>:SOA:MTESt<n>:SEGMent<o>:COUNt? ..... 1296
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:ADD ..... 1297
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:COUNt? ..... 1297
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:REMove ..... 1298
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:VALid? ..... 1298
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:X ..... 1298
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:Y ..... 1299
POWer<m>:SOA:MTESt<n>:SEGMent<o>:REMove ..... 1299

**POWer<m>:SOA:LIMit:APPLy**

Generates a mask using the given maximum values. The result is mask segment in the first quadrant of the XY-diagram.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Generate mask" on page 409

**POWer<m>:SOA:LIMit:IMAX** <CurrentMax>

Sets the maximum current for the mask.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<CurrentMax>
Range: 0 to 1E+26
Increment: 0.001
*RST: 1
Default unit: A

**Usage:** Asynchronous command

**Manual operation:** See "I max" on page 408

**POWer<m>:SOA:LIMit:PMAX** <PowerMax>

Sets the maximum power for the mask.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<PowerMax>
Range: 0 to 1E+26
Increment: 0.001
*RST: 50
Default unit: W

**Usage:** Asynchronous command

**Manual operation:** See "P max" on page 409

---

**POWer<m>:SOA:LIMit:VMAX** <VoltageMax>

Sets the maximum voltage for the mask.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<VoltageMax>
Range: 0 to 1E+26
Increment: 0.001
*RST: 100
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "U max" on page 408

---

**POWer<m>:SOA:MTESt<n>:SEGMent<o>:ADD**

Adds a new segment to the SOA mask. Consider the order of the segments (1, 2, 3, 4) and use the next free segment index. If the specified segment index already exists, the given number is ignored and the new segment gets the next free index.

The new segment has no points. Use `POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:ADD` to add the points.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

<o> 1...4, index of the segment.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add segment" on page 408

---

**POWer<m>:SOA:MTESt<n>:SEGMent<o>:COUNt?**

Returns the number of segments that belong to the SOA mask.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

<o> Irrelevant, omit the suffix.

**Return values:**

<Count> Number of the mask segments

Range: 0 to 1
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Add segment" on page 408

---

**POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:ADD**

Adds a corner point to the selected mask segment at the next free suffix. The new point has the coordinates 0;0.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

<o> 1...4, index of the segment.

<p> Index is ignored.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add segment" on page 408

---

**POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:COUNt?** [MAX]

Returns the number of points that was added to the indicated mask segment.

You can query the maximum value with `<command>? MAX`.

**Suffix:**

<m> 1..*

<n> *

<o> *

<p> *

**Return values:**

<Count> Number of defined points

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Add segment" on page 408

**POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:REMove**

Deletes the specified corner point of the mask segment. When a point is deleted, the remaining points are reordered similar to segment deletion (`POWer<m>:SOA:MTESt<n>:SEGMent<o>:REMove`).

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

<o> 1...4, index of the segment.

<p> 1...9, point index

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add segment" on page 408

**POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:VALid?**

Checks the validity of the selected point.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

<o> 1...4, index of the segment.

<p> 1...9, point index

**Return values:**

<Valid> OFF | ON

ON = valid
OFF = invalid

*RST: ON

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Add segment" on page 408

**POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:X** <X>

Sets the horizontal position of the selected point.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

<o> 1...4, index of the segment.

<p> 1...4, index of the segment.

**Parameters:**

<X>

Range: -1E+26 to 1E+26
Increment: 1E-06
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Point, X, Y" on page 336

---

**POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:Y** <Y>

Sets the vertical position of the selected point.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

<o> 1...4, index of the segment.

<p> 1...4, index of the segment.

**Parameters:**

<Y>

Range: -1E+26 to 1E+26
Increment: 1E-06
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Point, X, Y" on page 336

---

**POWer<m>:SOA:MTESt<n>:SEGMent<o>:REMove**

Deletes the specified mask segment. When a segment is deleted, the remaining segments are reordered. For example, there are 3 segments (1, 2, 3). When you delete segment 2, then segment 3 gets the index 2.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

<o> 1...4, index of the segment.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add segment" on page 408

### SOA actions

POWer<m>:SOA:MTESt<n>:ONViolation:BEEP ..... 1300
POWer<m>:SOA:MTESt<n>:ONViolation:SCReenshot ..... 1300
POWer<m>:SOA:MTESt<n>:ONViolation:STOP ..... 1300
POWer<m>:SOA:MTESt<n>:ONViolation:TRIGgerout ..... 1301
POWer<m>:SOA:MTESt<n>:ONViolation:WFMSave ..... 1301

**POWer<m>:SOA:MTESt<n>:ONViolation:BEEP** <Beep>

Generates a beep sound if the command is set to `SUCCess` or `VIOLation`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Parameters:**
<Beep> NOACtion | SUCCess | VIOLation
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Beep" on page 339

**POWer<m>:SOA:MTESt<n>:ONViolation:SCReenshot** <SaveScreenshot>

Saves the waveform data to file if the command is set to `SUCCess` or `VIOLation`.

To configure the screenshot settings, use the commands described in Section 18.13.8, "Screenshots", on page 1137.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Parameters:**
<SaveScreenshot> NOACtion | SUCCess | VIOLation
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Screenshot" on page 340

**POWer<m>:SOA:MTESt<n>:ONViolation:STOP** <StopAcq>

Stops the running acquisition if the command is set to `SUCCess` or `VIOLation`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Parameters:**

<StopAcq> NOACtion | SUCCess | VIOLation

*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Stop acq" on page 339

---

**POWer<m>:SOA:MTESt<n>:ONViolation:TRIGgerout** <TrigOutPls>

Sends an outgoing pulse to the [Trigger Out] connector if the command is set to `SUCCess` or `VIOLation`.

To configure the pulse, user the following commands:

- `TRIGger:ACTions:OUT:SOURce` on page 1063
- `TRIGger:ACTions:OUT:POLarity` on page 1064
- `TRIGger:ACTions:OUT:DELay` on page 1063
- `TRIGger:ACTions:OUT:PLENgth` on page 1064

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Parameters:**

<TrigOutPls> NOACtion | SUCCess | VIOLation

*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Trigger out pulse" on page 340

---

**POWer<m>:SOA:MTESt<n>:ONViolation:WFMSave** <SaveWfm>

Saves the waveform data to file if the command is set to `SUCCess` or `VIOLation`.

To define the path and file names, use the EXPort:WAVeform:AUTonaming:* commands:

- `EXPort:WAVeform:AUTonaming:NAME` on page 1127
- `EXPort:WAVeform:AUTonaming:PATH` on page 1127
- `EXPort:WAVeform:AUTonaming:TYPE` on page 1128

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Parameters:**

<SaveWfm> NOACtion | SUCCess | VIOLation

*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Save wfm" on page 339

### SOA results

POWer<m>:SOA:RESult<n>:COUNt:FWAVeforms? 1302
POWer<m>:SOA:RESult<n>:COUNt:PWAVeforms? 1302
POWer<m>:SOA:RESult<n>:COUNt:WAVeforms? 1302
POWer<m>:SOA:RESult<n>:FRATe? 1303
POWer<m>:SOA:RESult<n>[:RESult]? 1303

#### POWer<m>:SOA:RESult<n>:COUNt:FWAVeforms?

Returns the number of acquisitions that have failed the mask test, they hit the mask limits.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**
<AcqsFailed> Range: 0 to 10000000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

#### POWer<m>:SOA:RESult<n>:COUNt:PWAVeforms?

Returns the number of acquisitions that have passed the mask test without violation.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**
<AcqsPassed> Range: 0 to 10000000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

#### POWer<m>:SOA:RESult<n>:COUNt:WAVeforms?

Returns the number of tested acquisitions (waveforms).

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**
<AcqsCompleted> Range: 0 to 10000000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**POWer<m>:SOA:RESult<n>:FRATe?**

Returns the ratio of failed acquisitions to the number of tested acquisitions.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**
<FailRate> *RST: 0
Default unit: %

**Usage:** Query only
Asynchronous command

---

**POWer<m>:SOA:RESult<n>[:RESult]?**

Returns the summary test status.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**
<TestResult> PASS | FAIL
*RST: PASS

**Usage:** Query only
Asynchronous command

### 18.17.3.8 Power waveform

POWer<m>:EFFiciency:DISPlay:WAVeform:INPut<n>:POWer[:ENABle] 1304
POWer<m>:EFFiciency:DISPlay:WAVeform:OUTPut<n>:POWer[:ENABle] 1304
POWer<m>:EFFiciency:DISPlay:WAVeform:TOTal<n>:OPOWer[:ENABle] 1304
POWer<m>:HARMonics:DISPlay:POWer:WAVeform:ENABle 1304
POWer<m>:QUALity:DISPlay:POWer:WAVeform:ENABle 1304
POWer<m>:SWITching:DISPlay:POWer:WAVeform[:ENABle] 1304
POWer<m>:EFFiciency:INPut<n>:POWer:VERTical:OFFSet 1304
POWer<m>:EFFiciency:OUTPut<n>:POWer:VERTical:OFFSet 1304
POWer<m>:EFFiciency:TOTal<n>:OPOWer:VERTical:OFFSet 1304
POWer<m>:HARMonics:POWer:VERTical:OFFSet 1304
POWer<m>:QUALity:POWer:VERTical:OFFSet 1304

POWer<m>:SWITching:POWer:VERTical:OFFSet....1304
POWer<m>:EFFiciency:INPut<n>:POWer:VERTical:SCALe:MODE....1305
POWer<m>:EFFiciency:OUTPut<n>:POWer:VERTical:SCALe:MODE....1305
POWer<m>:EFFiciency:TOTal<n>:OPOWer:VERTical:SCALe:MODE....1305
POWer<m>:HARMonics:POWer:VERTical:SCALe:MODE....1305
POWer<m>:QUALity:POWer:VERTical:SCALe:MODE....1305
POWer<m>:SWITching:POWer:VERTical:SCALe:MODE....1305
POWer<m>:EFFiciency:INPut<n>:POWer:VERTical:SCALe[:VALue]....1305
POWer<m>:EFFiciency:OUTPut<n>:POWer:VERTical:SCALe[:VALue]....1305
POWer<m>:EFFiciency:TOTal<n>:OPOWer:VERTical:SCALe[:VALue]....1305
POWer<m>:HARMonics:POWer:VERTical:SCALe[:VALue]....1305
POWer<m>:QUALity:POWer:VERTical:SCALe[:VALue]....1305
POWer<m>:SWITching:POWer:VERTical:SCALe[:VALue]....1305

---

**POWer<m>:EFFiciency:DISPlay:WAVeform:INPut<n>:POWer[:ENABle]** <State>
**POWer<m>:EFFiciency:DISPlay:WAVeform:OUTPut<n>:POWer[:ENABle]** <State>
**POWer<m>:EFFiciency:DISPlay:WAVeform:TOTal<n>:OPOWer[:ENABle]** <State>
**POWer<m>:HARMonics:DISPlay:POWer:WAVeform:ENABle** <State>
**POWer<m>:QUALity:DISPlay:POWer:WAVeform:ENABle** <State>
**POWer<m>:SWITching:DISPlay:POWer:WAVeform[:ENABle]** <State>

Displays or hides the specified power waveform.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> INPut and TOTal index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**

<State> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Display" on page 415

---

**POWer<m>:EFFiciency:INPut<n>:POWer:VERTical:OFFSet** <VerticalOffset>
**POWer<m>:EFFiciency:OUTPut<n>:POWer:VERTical:OFFSet** <VerticalOffset>
**POWer<m>:EFFiciency:TOTal<n>:OPOWer:VERTical:OFFSet** <VerticalOffset>
**POWer<m>:HARMonics:POWer:VERTical:OFFSet** <VerticalOffset>
**POWer<m>:QUALity:POWer:VERTical:OFFSet** <VerticalOffset>
**POWer<m>:SWITching:POWer:VERTical:OFFSet** <VerticalOffset>

Sets the vertical offset of the power waveform if the relevant `POWer<m>:*:POWer:VERTical:SCALe:MODE` command is set to `MANual`.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> INPut and TOTal index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**

<VerticalOffset> Range: -100000000000000 to 100000000000000
Increment: 0.01
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Offset" on page 416

---

**POWer<m>:EFFiciency:INPut<n>:POWer:VERTical:SCALe:MODE** <VertSclMd>
**POWer<m>:EFFiciency:OUTPut<n>:POWer:VERTical:SCALe:MODE** <VertSclMd>
**POWer<m>:EFFiciency:TOTal<n>:OPOWer:VERTical:SCALe:MODE** <VertSclMd>
**POWer<m>:HARMonics:POWer:VERTical:SCALe:MODE** <VertSclMd>
**POWer<m>:QUALity:POWer:VERTical:SCALe:MODE** <VertSclMd>
**POWer<m>:SWITching:POWer:VERTical:SCALe:MODE** <VertSclMd>

Selects the scaling mode for display of the power waveform.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> INPut and TOTal index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**

<VertSclMd> MANual | AUTO

**AUTO**
The instrument selects the most appropriate scale.

**MANual**
You can change the vertical scale and offset.

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Scale mode" on page 416

---

**POWer<m>:EFFiciency:INPut<n>:POWer:VERTical:SCALe[:VALue]**
<VerticalScale>
**POWer<m>:EFFiciency:OUTPut<n>:POWer:VERTical:SCALe[:VALue]**
<VerticalScale>
**POWer<m>:EFFiciency:TOTal<n>:OPOWer:VERTical:SCALe[:VALue]**
<VerticalScale>
**POWer<m>:HARMonics:POWer:VERTical:SCALe[:VALue]** <VerticalScale>
**POWer<m>:QUALity:POWer:VERTical:SCALe[:VALue]** <VerticalScale>
**POWer<m>:SWITching:POWer:VERTical:SCALe[:VALue]** <VerticalScale>

Sets the vertical scale of the power waveform if the relevant `POWer<m>:*:POWer:VERTical:SCALe:MODE` command is set to `MANual`.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> INPut and TOTal index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**

<VerticalScale>
Range: 1E-12 to 100000000000000
Increment: 1E-05
*RST: 0.5
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Scale" on page 416

### 18.17.3.9 Reference level

POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:HYSTeresis....1307
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:HYSTeresis....1307
POWer<m>:HARMonics:REFLevel<rl>:ABSolute:HYSTeresis....1307
POWer<m>:QUALity:REFLevel<rl>:ABSolute:HYSTeresis....1307
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:LLEVel....1307
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:LLEVel....1307
POWer<m>:HARMonics:REFLevel<rl>:ABSolute:LLEVel....1307
POWer<m>:QUALity:REFLevel<rl>:ABSolute:LLEVel....1307
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:MLEVel....1308
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:MLEVel....1308
POWer<m>:HARMonics:REFLevel<rl>:ABSolute:MLEVel....1308
POWer<m>:QUALity:REFLevel<rl>:ABSolute:MLEVel....1308
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:ULEVel....1308
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:ULEVel....1308
POWer<m>:HARMonics:REFLevel<rl>:ABSolute:ULEVel....1308
POWer<m>:QUALity:REFLevel<rl>:ABSolute:ULEVel....1308
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:LMODe....1309
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:LMODe....1309
POWer<m>:HARMonics:REFLevel<rl>:LMODe....1309
POWer<m>:QUALity:REFLevel<rl>:LMODe....1309
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:HYSTeresis....1309
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:HYSTeresis....1309
POWer<m>:HARMonics:REFLevel<rl>:RELative:HYSTeresis....1309
POWer<m>:QUALity:REFLevel<rl>:RELative:HYSTeresis....1309
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:LOWer....1309
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:LOWer....1309
POWer<m>:HARMonics:REFLevel<rl>:RELative:LOWer....1310
POWer<m>:QUALity:REFLevel<rl>:RELative:LOWer....1310
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MIDDle....1310
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MIDDle....1310
POWer<m>:HARMonics:REFLevel<rl>:RELative:MIDDle....1310
POWer<m>:QUALity:REFLevel<rl>:RELative:MIDDle....1310
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MODE....1310
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MODE....1310
POWer<m>:HARMonics:REFLevel<rl>:RELative:MODE....1311
POWer<m>:QUALity:REFLevel<rl>:RELative:MODE....1311

POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:UPPer ..... 1311
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:UPPer ..... 1311
POWer<m>:HARMonics:REFLevel<rl>:RELative:UPPer ..... 1311
POWer<m>:QUALity:REFLevel<rl>:RELative:UPPer ..... 1311

---

**POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:HYSTeresis** <HystAbs>
**POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:HYSTeresis** <HystAbs>
**POWer<m>:HARMonics:REFLevel<rl>:ABSolute:HYSTeresis** <HystAbs>
**POWer<m>:QUALity:REFLevel<rl>:ABSolute:HYSTeresis** <HystAbs>

Defines a hysteresis for the middle reference level. A rise or fall from the middle reference value that does not exceed the hysteresis is rejected as noise.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

<n> INPut index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**

<HystAbs>
Range: 0 to 1E+26
Increment: 0.001
*RST: 0.005
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis" on page 414

---

**POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:LLEVel** <LowerLevel>
**POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:LLEVel** <LowerLevel>
**POWer<m>:HARMonics:REFLevel<rl>:ABSolute:LLEVel** <LowerLevel>
**POWer<m>:QUALity:REFLevel<rl>:ABSolute:LLEVel** <LowerLevel>

Sets the lower reference level in absolute values. This is required, e.g., to determine a fall.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

<n> INPut index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**

<LowerLevel>
Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level, Lower level" on page 411

---

**POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:MLEVel** <MiddleLevel>
**POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:MLEVel** <MiddleLevel>
**POWer<m>:HARMonics:REFLevel<rl>:ABSolute:MLEVel** <MiddleLevel>
**POWer<m>:QUALity:REFLevel<rl>:ABSolute:MLEVel** <MiddleLevel>

Sets the middle reference level in absolute values.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

<n> INPut index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**
<MiddleLevel>
Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level, Lower level" on page 411

---

**POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:ULEVel** <UpperLevel>
**POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:ULEVel** <UpperLevel>
**POWer<m>:HARMonics:REFLevel<rl>:ABSolute:ULEVel** <UpperLevel>
**POWer<m>:QUALity:REFLevel<rl>:ABSolute:ULEVel** <UpperLevel>

Sets the upper reference level in absolute values. This is required to determine a rise.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

<n> INPut index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**
<UpperLevel>
Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level, Lower level" on page 411

**POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:LMODe** <LevelMode>
**POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:LMODe** <LevelMode>
**POWer<m>:HARMonics:REFLevel<rl>:LMODe** <LevelMode>
**POWer<m>:QUALity:REFLevel<rl>:LMODe** <LevelMode>

Defines if the reference level is set in absolute or relative values.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

<n> INPut index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**

<LevelMode> ABS | REL

*RST: REL

**Usage:** Asynchronous command

**Manual operation:** See "Level mode" on page 411

**POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:HYSTeresis** <HystRel>
**POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:HYSTeresis** <HystRel>
**POWer<m>:HARMonics:REFLevel<rl>:RELative:HYSTeresis** <HystRel>
**POWer<m>:QUALity:REFLevel<rl>:RELative:HYSTeresis** <HystRel>

Defines a hysteresis for the middle reference level. A rise or fall from the middle reference value that does not exceed the hysteresis is rejected as noise.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

<n> INPut index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**

<HystRel> Range: 0 to 50
Increment: 1
*RST: 10
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis" on page 414

**POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:LOWer** <LowRefLevRel>
**POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:LOWer** <LowRefLevRel>

**POWer<m>:HARMonics:REFLevel<rl>:RELative:LOWer** <LowRefLevRel>
**POWer<m>:QUALity:REFLevel<rl>:RELative:LOWer** <LowRefLevRel>

Sets the lower relative reference level if `POWer<m>:QUALity:REFLevel<rl>:RELative:MODE` is set to `USER`.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

<n> INPut index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**

<LowRefLevRel>
Range: -100 to 200
Increment: 1
*RST: 10
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level,Lower level" on page 413

---

**POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MIDDle** <MidRefLevRel>
**POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MIDDle** <MidRefLevRel>
**POWer<m>:HARMonics:REFLevel<rl>:RELative:MIDDle** <MidRefLevRel>
**POWer<m>:QUALity:REFLevel<rl>:RELative:MIDDle** <MidRefLevRel>

Sets the middle relative reference level if `POWer<m>:QUALity:REFLevel<rl>:RELative:MODE` is set to `USER`.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

<n> INPut index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**

<MidRefLevRel>
Range: -100 to 200
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level,Lower level" on page 413

---

**POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MODE** <RelativeLevels>
**POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MODE** <RelativeLevels>

**POWer<m>:HARMonics:REFLevel<rl>:RELative:MODE** <RelativeLevels>
**POWer<m>:QUALity:REFLevel<rl>:RELative:MODE** <RelativeLevels>

The lower, middle and upper reference levels, defined as percentages of the high signal level.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

<n> INPut index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**

<RelativeLevels> FIVE | TEN | TWENty | USER

**FIVE**
5/50/95

**TEN**
10/50/90

**TWENty**
20/50/80

**USER**
Set the reference levels to individual values with `POWer<m>:QUALity:REFLevel<rl>:RELative:LOWer`, `POWer<m>:QUALity:REFLevel<rl>:RELative:MIDDle`, and `POWer<m>:QUALity:REFLevel<rl>:RELative:UPPer`.

*RST: TEN

**Usage:** Asynchronous command

**Manual operation:** See "Relative levels" on page 412

---

**POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:UPPer** <UppRefLevRel>
**POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:UPPer** <UppRefLevRel>
**POWer<m>:HARMonics:REFLevel<rl>:RELative:UPPer** <UppRefLevRel>
**POWer<m>:QUALity:REFLevel<rl>:RELative:UPPer** <UppRefLevRel>

Sets the upper relative reference level if `POWer<m>:QUALity:REFLevel<rl>:RELative:MODE` on page 1311 is set to `USER`.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

<n> INPut index: omit the suffix. OUTPut index: 1 to 3 (suffix >1 requires an instrument with more than 4 channels).

**Parameters:**

<UppRefLevRel>
Range: -100 to 200
Increment: 1
*RST: 90
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level,Lower level" on page 413

### 18.17.3.10 Power analysis data

This section describes the commands for exporting power analysis measurement data.

POWer<m>:EFFiciency:INPut<n>:POWer:DATA:HEADer?....1312
POWer<m>:EFFiciency:OUTPut<n>:POWer:DATA:HEADer?....1312
POWer<m>:EFFiciency:TOTal<n>:OPOWer:DATA:HEADer?....1312
POWer<m>:HARMonics:POWer:DATA:HEADer?....1312
POWer<m>:QUALity:POWer:DATA:HEADer?....1312
POWer<m>:SWITching:POWer:DATA:HEADer?....1312
POWer<m>:EFFiciency:INPut<n>:POWer:DATA[:VALues]?....1313
POWer<m>:EFFiciency:OUTPut<n>:POWer:DATA[:VALues]?....1313
POWer<m>:EFFiciency:TOTal<n>:OPOWer:DATA[:VALues]?....1313
POWer<m>:HARMonics:POWer:DATA[:VALues]?....1313
POWer<m>:QUALity:POWer:DATA[:VALues]?....1313
POWer<m>:SWITching:POWer:DATA[:VALues]?....1313

---

**POWer<m>:EFFiciency:INPut<n>:POWer:DATA:HEADer?**
**POWer<m>:EFFiciency:OUTPut<n>:POWer:DATA:HEADer?**
**POWer<m>:EFFiciency:TOTal<n>:OPOWer:DATA:HEADer?**
**POWer<m>:HARMonics:POWer:DATA:HEADer?**
**POWer<m>:QUALity:POWer:DATA:HEADer?**
**POWer<m>:SWITching:POWer:DATA:HEADer?**

Returns the header of the power analysis waveform data. The header contains the attributes of the waveform.

For power harmonics measurements, data is only available if `POWer<m>:HARMonics:STANdard` is set to `ENC` or `END`.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

**Return values:**

<XStart> 1. header value: start time in s

<XStop> 2. header value: end time in s

<RecordLength> 3. header value: record length of the waveform in samples

<ValuesPerSample> 4. header value: number of values per sample. For power quality measurements, the value is 1.

**Usage:** Query only
Asynchronous command

---

**POWer<m>:EFFiciency:INPut<n>:POWer:DATA[:VALues]?** [<Offset>[,<Length>]]
**POWer<m>:EFFiciency:OUTPut<n>:POWer:DATA[:VALues]?** [<Offset>[,<Length>]]
**POWer<m>:EFFiciency:TOTal<n>:OPOWer:DATA[:VALues]?** [<Offset>[,<Length>]]
**POWer<m>:HARMonics:POWer:DATA[:VALues]?** [<Offset>[,<Length>]]
**POWer<m>:QUALity:POWer:DATA[:VALues]?** [<Offset>[,<Length>]]
**POWer<m>:SWITching:POWer:DATA[:VALues]?** [<Offset>[,<Length>]]

Returns the data of the power analysis waveform points for transmission from the instrument to the controlling computer.

Without parameters, the complete waveform is retrieved. Using the offset and length parameters, data can be retrieved in smaller portions, which makes the command faster. If you send only one parameter, it is interpreted as offset, and the data is retrieved from offset to the end of the waveform.

To set the export format, use `FORMat[:DATA]`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Query parameters:**
<Offset> Number of offset waveform points to be skipped.

<Length> Number of waveform points to be retrieved.

**Return values:**
<Data> List of comma-separated values according to the format and content settings.

**Usage:** Query only
Asynchronous command

## 18.17.4 3-phase power analysis (option R&S MXO3-K333)

Some of the commands in the following section are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, section "Command sequence and synchronization"

- 3-phase general settings....1314
- 3-phase power quality....1315
- 3-phase power harmonics....1329
- 3-phase power efficiency....1336

- 3-phase power sources....1341
- 3-phase power waveform....1345
- 3-phase reference level....1349
- 3-phase power analysis data....1354

### 18.17.4.1 3-phase general settings

P3PHase<m>[:ENABle]....1314
P3PHase<m>:TYPE....1314
EXPort:RESult:SELect:P3PHase<m>....1314

**P3PHase<m>[:ENABle]** <State>

Enables the 3-phase power measurement instance.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<State> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "+ Add" on page 418

**P3PHase<m>:TYPE** <Type>

Sets the type for the respective 3-phase power analysis measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Type> P3Quality | P3Harmonics | P3EFficiency

P3Quality = 3-phase power quality
P3Harmonics = 3-phase power harmonics
P3EFficiency = 3-phase power efficiency

**Example:** `P3PH1:TYPE P3Q`
Sets the first power analysis measurement to quality.

**Usage:** Asynchronous command

**Manual operation:** See "+ Add" on page 418

**EXPort:RESult:SELect:P3PHase<m>** <ExportResults>

If enabled, includes the results of selected 3-phase power analysis measurement in the results export file.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ExportResults> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

### 18.17.4.2 3-phase power quality


- 3-phase power quality setup ..... 1315
- 3-phase phasor diagram ..... 1316
- 3-phase power quality measurement display ..... 1317
- 3-phase power quality results ..... 1321


#### 3-phase power quality setup


P3PHase<m>:QUALity:FREQuency[:VALue] ..... 1315
P3PHase<m>:QUALity:FREQuency:USER ..... 1315


---

**P3PHase<m>:QUALity:FREQuency[:VALue]** <FundamFreq>

Sets the input frequency of the source signal in Hz.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<FundamFreq> F50 | F60 | F400 | USER | AUTO
*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Fund. frequency" on page 428

---

**P3PHase<m>:QUALity:FREQuency:USER** <UsrFundamFreq>

Sets the horizontal timebase to control the time scale of the waveform display.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<UsrFundamFreq> Range: 10 to 1000000
Increment: 1
*RST: 50
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "User Frequency" on page 428

### 3-phase phasor diagram

P3PHase<m>:QUALity:PHASor[:ENABle] ..... 1316
P3PHase<m>:QUALity:PHASor:VOLTage:OFFSet ..... 1316
P3PHase<m>:QUALity:PHASor:CURRent:OFFSet ..... 1316
P3PHase<m>:QUALity:PHASor:VOLTage:MAXimum ..... 1316
P3PHase<m>:QUALity:PHASor:CURRent:MAXimum ..... 1316
P3PHase<m>:QUALity:PHASor:FSCReen ..... 1317

---

**P3PHase<m>:QUALity:PHASor[:ENABle]** <State>

Enables the phasor measurement for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 429

---

**P3PHase<m>:QUALity:PHASor:VOLTage:OFFSet** <VoltageOffset>
**P3PHase<m>:QUALity:PHASor:CURRent:OFFSet** <CurrentOffset>

Sets the inner point of the phasor diagram.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<CurrentOffset> Range: -100000 to 100000
Increment: 0.1
*RST: 0
Default unit: A

**Usage:** Asynchronous command

**Manual operation:** See "Offset" on page 429

---

**P3PHase<m>:QUALity:PHASor:VOLTage:MAXimum** <VoltageMaximum>
**P3PHase<m>:QUALity:PHASor:CURRent:MAXimum** <CurrentMaximum>

Sets the outer circumference of the phasor diagram.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<CurrentMaximum> Range: -100000 to 100000
Increment: 0.1
*RST: 3
Default unit: A

**Usage:** Asynchronous command

**Manual operation:** See "Maximum" on page 430

---

**P3PHase<m>:QUALity:PHASor:FSCReen**

Selects the most appropriate scale for the voltage and current phasors.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Fit to Screen" on page 430

### 3-phase power quality measurement display

P3PHase<m>:QUALity:DISPlay:CURRent:CREStfactor<n>[:ENABle] 1317
P3PHase<m>:QUALity:DISPlay:CURRent:PEAK<n>[:ENABle] 1317
P3PHase<m>:QUALity:DISPlay:CURRent:RMS<n>[:ENABle] 1318
P3PHase<m>:QUALity:DISPlay:POWer<n>:ACTive[:ENABle] 1318
P3PHase<m>:QUALity:DISPlay:POWer<n>:APParent[:ENABle] 1318
P3PHase<m>:QUALity:DISPlay:POWer<n>:PFACtor[:ENABle] 1319
P3PHase<m>:QUALity:DISPlay:POWer<n>:PHASe[:ENABle] 1319
P3PHase<m>:QUALity:DISPlay:POWer<n>:REACtive[:ENABle] 1319
P3PHase<m>:QUALity:DISPlay:VOLTage:CREStfactor<n>[:ENABle] 1320
P3PHase<m>:QUALity:DISPlay:VOLTage:FREQuency<n>[:ENABle] 1320
P3PHase<m>:QUALity:DISPlay:VOLTage:PEAK<n>[:ENABle] 1320
P3PHase<m>:QUALity:DISPlay:VOLTage:RMS<n>[:ENABle] 1320

---

**P3PHase<m>:QUALity:DISPlay:CURRent:CREStfactor<n>[:ENABle]**
<DisplayState>

Enables the current crest factor measurement for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<DisplayState> OFF | ON
*RST: ON

**Usage:** Asynchronous command

---

**P3PHase<m>:QUALity:DISPlay:CURRent:PEAK<n>[:ENABle]** <DisplayState>

Enables the current peak value measurement for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<DisplayState> OFF | ON

*RST: ON

**Usage:** Asynchronous command

---

**P3PHase<m>:QUALity:DISPlay:CURRent:RMS<n>[:ENABle]** <DisplayState>

Enables the current RMS measurement for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<DisplayState> OFF | ON

*RST: ON

**Usage:** Asynchronous command

---

**P3PHase<m>:QUALity:DISPlay:POWer<n>:ACTive[:ENABle]** <DisplayState>

Enables the active measurement for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<DisplayState> OFF | ON

*RST: ON

**Usage:** Asynchronous command

---

**P3PHase<m>:QUALity:DISPlay:POWer<n>:APParent[:ENABle]** <DisplayState>

Enables the apparent power measurement for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<DisplayState> OFF | ON
*RST: ON

**Usage:** Asynchronous command

---

**P3PHase<m>:QUALity:DISPlay:POWer<n>:PFACtor[:ENABle]** <DisplayState>

Enables the power factor measurement for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement
<n> 1...3, index of the individual phase

**Parameters:**
<DisplayState> OFF | ON
*RST: ON

**Usage:** Asynchronous command

---

**P3PHase<m>:QUALity:DISPlay:POWer<n>:PHASe[:ENABle]** <DisplayState>

Enables the power phase measurement for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement
<n> 1...3, index of the individual phase

**Parameters:**
<DisplayState> OFF | ON
*RST: ON

**Usage:** Asynchronous command

---

**P3PHase<m>:QUALity:DISPlay:POWer<n>:REACtive[:ENABle]** <DisplayState>

Enables the reactive power measurement for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement
<n> 1...3, index of the individual phase

**Parameters:**
<DisplayState> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**P3PHase<m>:QUALity:DISPlay:VOLTage:CRESTfactor<n>[:ENABle]**
<DisplayState>

Enables the voltage crest factor measurement for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<DisplayState> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**P3PHase<m>:QUALity:DISPlay:VOLTage:FREQuency<n>[:ENABle]**
<DisplayState>

Enables the voltage frequency measurement for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<DisplayState> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**P3PHase<m>:QUALity:DISPlay:VOLTage:PEAK<n>[:ENABle]** <DisplayState>

Enables the voltage peak value measurement for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<DisplayState> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**P3PHase<m>:QUALity:DISPlay:VOLTage:RMS<n>[:ENABle]** <DisplayState>

Enables the voltage RMS measurement for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<DisplayState> OFF | ON
*RST: ON

**Usage:** Asynchronous command

### 3-phase power quality results

P3PHase<m>:QUALity:RESult:CSOurce:FREQuency<n>[:ACTual]?....1321
P3PHase<m>:QUALity:RESult:CURRent:CREStfactor<n>[:ACTual]?....1322
P3PHase<m>:QUALity:RESult:TOTal:CURRent:CREStfactor<n>[:ACTual]?....1322
P3PHase<m>:QUALity:RESult:CURRent:PEAK<n>[:ACTual]?....1322
P3PHase<m>:QUALity:RESult:TOTal:CURRent:PEAK<n>[:ACTual]?....1323
P3PHase<m>:QUALity:RESult:CURRent:RMS<n>[:ACTual]?....1323
P3PHase<m>:QUALity:RESult:TOTal:CURRent:RMS<n>[:ACTual]?....1323
P3PHase<m>:QUALity:RESult:POWer:ACTive<n>[:ACTual]?....1324
P3PHase<m>:QUALity:RESult:TOTal:POWer:ACTive<n>[:ACTual]?....1324
P3PHase<m>:QUALity:RESult:POWer:APParent<n>[:ACTual]?....1324
P3PHase<m>:QUALity:RESult:TOTal:POWer:APParent<n>[:ACTual]?....1324
P3PHase<m>:QUALity:RESult:POWer:PFACtor<n>[:ACTual]?....1325
P3PHase<m>:QUALity:RESult:TOTal:POWer:PFACtor<n>[:ACTual]?....1325
P3PHase<m>:QUALity:RESult:POWer:PHASe<n>[:ACTual]?....1325
P3PHase<m>:QUALity:RESult:TOTal:POWer:PHASe<n>[:ACTual]?....1326
P3PHase<m>:QUALity:RESult:POWer:REACtive<n>[:ACTual]?....1326
P3PHase<m>:QUALity:RESult:TOTal:POWer:REACtive<n>[:ACTual]?....1326
P3PHase<m>:QUALity:RESult:VOLTage:CREStfactor<n>[:ACTual]?....1327
P3PHase<m>:QUALity:RESult:TOTal:VOLTage:CREStfactor<n>[:ACTual]?....1327
P3PHase<m>:QUALity:RESult:VOLTage:PEAK<n>[:ACTual]?....1327
P3PHase<m>:QUALity:RESult:TOTal:VOLTage:PEAK<n>[:ACTual]?....1328
P3PHase<m>:QUALity:RESult:VOLTage:RMS<n>[:ACTual]?....1328
P3PHase<m>:QUALity:RESult:TOTal:VOLTage:RMS<n>[:ACTual]?....1328
P3PHase<m>:QUALity:PHASor:RESult:VOLTage<n>:ANGLe[:ACTual]?....1329

---

#### P3PHase<m>:QUALity:RESult:CSOurce:FREQuency<n>[:ACTual]?

Returns the cycle source frequency of each channel for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1 to 8, index of the analog channel

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:CURRent:CREStfactor<n>[:ACTual]?**

Returns the current crest factors for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:TOTal:CURRent:CREStfactor<n>[:ACTual]?**

Returns the current total crest factor for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:CURRent:PEAK<n>[:ACTual]?**

Returns the current peak values for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:TOTal:CURRent:PEAK<n>[:ACTual]?**

Returns the total current peak values for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:CURRent:RMS<n>[:ACTual]?**

Returns the current RMS values for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:TOTal:CURRent:RMS<n>[:ACTual]?**

Returns the current total RMS for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**P3PHase<m>:QUALity:RESult:POWer:ACTive<n>[:ACTual]?**

Returns the active power for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**P3PHase<m>:QUALity:RESult:TOTal:POWer:ACTive<n>[:ACTual]?**

Returns the total active power for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**P3PHase<m>:QUALity:RESult:POWer:APParent<n>[:ACTual]?**

Returns the apparent power for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**P3PHase<m>:QUALity:RESult:TOTal:POWer:APParent<n>[:ACTual]?**

Returns the total apparent power for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:POWer:PFACtor<n>[:ACTual]?**

Returns the power factor for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:TOTal:POWer:PFACtor<n>[:ACTual]?**

Returns the total power factor for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:POWer:PHASe<n>[:ACTual]?**

Returns the phase values for individual phases for the 3-phase power quality analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**

<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:TOTal:POWer:PHASe<n>[:ACTual]?**

Returns the total phase value for the 3-phase power quality analysis.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**

<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:POWer:REACtive<n>[:ACTual]?**

Returns the reactive power for individual phases for the 3-phase power quality analysis.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**

<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:TOTal:POWer:REACtive<n>[:ACTual]?**

Returns the total reactive power for the 3-phase power quality analysis.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**

<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:VOLTage:CREStfactor<n>[:ACTual]?**

Returns the voltage crest factors for individual phases for the 3-phase power quality analysis.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**

<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:TOTal:VOLTage:CREStfactor<n>[:ACTual]?**

Returns the total voltage crest factor for the 3-phase power quality analysis.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**

<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:VOLTage:PEAK<n>[:ACTual]?**

Returns the voltage peak values for individual phases for the 3-phase power quality analysis.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**

<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:TOTal:VOLTage:PEAK<n>[:ACTual]?**

Returns the total voltage peak value for the 3-phase power quality analysis.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**

<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:VOLTage:RMS<n>[:ACTual]?**

Returns the voltage RMS values for individual phases for the 3-phase power quality analysis.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**

<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:RESult:TOTal:VOLTage:RMS<n>[:ACTual]?**

Returns the total voltage RMS value for the 3-phase power quality analysis.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> Irrelevant, omit the suffix.

**Return values:**

<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:PHASor:RESult:VOLTage<n>:ANGLe[:ACTual]?**

Returns the voltage angle of the selected phasor phase relative to phase 1.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**

<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

### 18.17.4.3 3-phase power harmonics


- 3-phase power harmonics setup ..... 1329
- 3-phase bar graph ..... 1332
- 3-phase power harmonics results ..... 1334


#### 3-phase power harmonics setup


P3PHase<m>:HARMonics:STANdard ..... 1329
P3PHase<m>:HARMonics:REVision ..... 1330
P3PHase<m>:HARMonics:FREQuency:EN ..... 1330
P3PHase<m>:HARMonics:FREQuency:MIL ..... 1330
P3PHase<m>:HARMonics:PFACtor:MODE ..... 1331
P3PHase<m>:HARMonics:PFACtor:USER ..... 1331
P3PHase<m>:HARMonics:RPOWer:MODE ..... 1331
P3PHase<m>:HARMonics:RPOWer:USER ..... 1332


---

**P3PHase<m>:HARMonics:STANdard** <Limits>

Sets a standard for the current harmonic measurement.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

**Parameters:**

<Limits> ENA | ENB | ENC | END | MIL

ENA: EN 61000-3-2 Class A
ENB: EN 61000-3-2 Class B

ENC: EN 61000-3-2 Class C
END: EN 61000-3-2 Class D
MIL: MIL-STD-1399

*RST: ENA

**Usage:** Asynchronous command

**Manual operation:** See "Standard, Revision" on page 435

---

**P3PHase<m>:HARMonics:REVision** <Revision>

Selects the revision of the EN61000 standard, if P3PHase<m>:HARMonics:STANdard is set to ENA / ENB / ENC / END.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Revision> REV2011 | REV2019

*RST: REV2019

**Usage:** Asynchronous command

**Manual operation:** See "Standard, Revision" on page 435

---

**P3PHase<m>:HARMonics:FREQuency:EN** <FundamFreqEN61000>

Sets the fundamental frequency of the input signal for the EN61000 standard, if P3PHase<m>:HARMonics:STANdard is set to ENA / ENB / ENC / END.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<FundamFreqEN61000> F50 | F60 | AUTO

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Fund. frequency" on page 436

---

**P3PHase<m>:HARMonics:FREQuency:MIL** <FundamFreqMIL>

Sets the fundamental frequency of the input signal for the MIL-STD-1399 standard, if P3PHase<m>:HARMonics:STANdard is set to MIL.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<FundamFreqMIL> F60 | F400

*RST: F60

**Usage:** Asynchronous command

**Manual operation:** See "Fund. frequency" on page 436

---

**P3PHase<m>:HARMonics:PFACtor:MODE** <PowFactMd>

Available if `P3PHase<m>:HARMonics:STANdard` is set to `ENC`.

Selects if the power factor is defined automatically, or a user-defined value is used (`P3PHase<m>:HARMonics:PFACtor:USER`).

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<PowFactMd> AUTO | USER
*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Power factor" on page 436

---

**P3PHase<m>:HARMonics:PFACtor:USER** <UsrPowFact>

Available if `P3PHase<m>:HARMonics:STANdard` is set to `ENC` and `P3PHase<m>:HARMonics:PFACtor:MODE` is set to `USER`.

Sets a user-defined power factor.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<UsrPowFact> Range: 0 to 100
Increment: 1
*RST: 100
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Power factor" on page 436

---

**P3PHase<m>:HARMonics:RPOWer:MODE** <ActPowMd>

Available if `P3PHase<m>:HARMonics:STANdard` is set to `END`.

Selects if the power factor is defined automatically, or a user-defined value is used (`P3PHase<m>:HARMonics:RPOWer:USER`).

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<ActPowMd> AUTO | USER
*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Real power, Power" on page 436

---

**P3PHase<m>:HARMonics:RPOWer:USER** <UsrActPow>

Available if `P3PHase<m>:HARMonics:STANdard` is set to `END` and `P3PHase<m>:HARMonics:RPOWer:USER` is set to `USER`.

Sets a user-defined real power value.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<UsrActPow>
Range: 0 to 10000
Increment: 1
*RST: 100
Default unit: W

**Usage:** Asynchronous command

**Manual operation:** See "Real power, Power" on page 436

### 3-phase bar graph

P3PHase<m>:HARMonics:DISPlay:BARGraph[:ENABle] 1332
P3PHase<m>:HARMonics:DISPlay:HARMonics[:DISPlay] 1332
P3PHase<m>:HARMonics:DISPlay:BARGraph:VERTical:SCALe:TYPE 1333
P3PHase<m>:HARMonics:DISPlay:HARMonics:STARt[:VALue] 1333
P3PHase<m>:HARMonics:DISPlay:HARMonics:STOP[:VALue] 1333

---

**P3PHase<m>:HARMonics:DISPlay:BARGraph[:ENABle]** <State>

Displays or hides the bar graph.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<State>
OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 437

---

**P3PHase<m>:HARMonics:DISPlay:HARMonics[:DISPlay]** <DispedHarmonics>

Selects which harmonics are displayed in the bar graph: all, odd even or depending on the standard definition.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<DispedHarmonics> ALL | EVEN | ODD | STANdard
*RST: STANdard

**Usage:** Asynchronous command

**Manual operation:** See "Displ. harmonics" on page 437

---

**P3PHase<m>:HARMonics:DISPlay:BARGraph:VERTical:SCALe:TYPE** <Scaling>

Selects a logarithmic or linear scale for the display for the harmonics bar graph.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Scaling> LIN | LOG
*RST: LIN

**Usage:** Asynchronous command

**Manual operation:** See "Scale type" on page 437

---

**P3PHase<m>:HARMonics:DISPlay:HARMonics:STARt[:VALue]** <BarGraphStart>

Sets the first harmonic to be shown in the bar graph.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<BarGraphStart> Range: 1 to 37
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Start harmonic" on page 437

---

**P3PHase<m>:HARMonics:DISPlay:HARMonics:STOP[:VALue]** <BarGraphStop>

Sets the last harmonic to be shown in the bar graph.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<BarGraphStop> Range: 3 to 40
Increment: 1
*RST: 40

**Usage:** Asynchronous command

**Manual operation:** See "Max harmonic" on page 437

### 3-phase power harmonics results

P3PHase<m>:HARMonics:AVAilable<n> 1334
P3PHase<m>:HARMonics:MEASurement:CSOurce:FREQuency<n>[:ACTual]? 1334
P3PHase<m>:HARMonics:MEASurement:POWer<n>:ACTive[:ACTual]? 1334
P3PHase<m>:HARMonics:MEASurement:THDFundament<n>[:ACTual]? 1335
P3PHase<m>:HARMonics:MEASurement:THDRms<n>[:ACTual]? 1335
P3PHase<m>:HARMonics:RESult<n>:VIOLation<o>? 1335
P3PHase<m>:HARMonics:RESult<n>[:FREQuency]? 1335
P3PHase<m>:HARMonics:RESult<n>:MAGNitude<o>:PERCent[:ACTual]? 1336
P3PHase<m>:HARMonics:RESult<n>:MAGNitude<o>:RMS[:ACTual]? 1336

#### P3PHase<m>:HARMonics:AVAilable<n>

Returns the number of measured harmonics for the individual phases.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**
<Count>

**Usage:** Asynchronous command

#### P3PHase<m>:HARMonics:MEASurement:CSOurce:FREQuency<n>[:ACTual]?

Returns the cycle source frequency of each channel for the 3-phase power harmonics analysis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1 to 8, index of the analog channel

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

#### P3PHase<m>:HARMonics:MEASurement:POWer<n>:ACTive[:ACTual]?

Returns the active power for the individual phases.. Only available, if `P3PHase<m>:HARMonics:STANdard` is set to `ENC / END`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:HARMonics:MEASurement:THDFundament<n>[:ACTual]?**

Returns the total harmonic distortion relative to fundamental (THDF) for the individual phases.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:HARMonics:MEASurement:THDRms<n>[:ACTual]?**

Returns the total harmonic distortion relative to the RMS amplitude (THD RMS) for the individual phases.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:HARMonics:RESult<n>:VIOLation<o>?**

Queries if the value of o-th harmonic is within the defined limit for the current measurement for the individual phases.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

<o> 1...n, index of the harmonics

**Return values:**
<Harmonic>

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:HARMonics:RESult<n>[:FREQuency]?**

Returns the frequency of the n-th harmonic.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...n, index of the harmonics

**Return values:**
<Harmonic>

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:HARMonics:RESult<n>:MAGNitude<o>:PERCent[:ACTual]?**

Returns the magnitude of the o-th harmonic in percent for the individual phases.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

<o> 1...n, index of the harmonics

**Return values:**
<Harmonic>

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:HARMonics:RESult<n>:MAGNitude<o>:RMS[:ACTual]?**

Returns the magnitude of the o-th harmonic as RMS for the individual phases.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

<o> 1...n, index of the harmonics

**Return values:**
<Harmonic>

**Usage:** Query only
Asynchronous command

### 18.17.4.4 3-phase power efficiency


- 3-phase power efficiency setup....1336
- 3-phase power efficiency results....1337


#### 3-phase power efficiency setup


P3PHase<m>:EFFiciency:STATistic:ENABle....1337
P3PHase<m>:EFFiciency:STATistic:RESet....1337
P3PHase<m>:EFFiciency:STATistic:WFMCount?....1337

**P3PHase<m>:EFFiciency:STATistic:ENABle** <EnabStatistics>

Enable the statistical calculation for the selected 3-phase efficiency measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<EnabStatistics> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Statistics" on page 441

**P3PHase<m>:EFFiciency:STATistic:RESet**

Restarts the statistical calculation for the selected 3-phase efficiency measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Clear results" on page 442

**P3PHase<m>:EFFiciency:STATistic:WFMCount?**

Returns the number of waveforms included in the 3-phase efficiency measurement.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<WaveformCount>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Statistics" on page 441

### 3-phase power efficiency results

See "Overview of statistic results" on page 1247.

P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer[:ACTual]? 1338
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer[:ACTual]? 1338
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult[:ACTual]? 1338
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:AVERage? 1338
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:AVERage? 1338
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:AVERage? 1338
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:EVTCount? 1339
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:EVTCount? 1339
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:EVTCount? 1339

P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:NPEak?....1339
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:NPEak?....1339
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:NPEak?....1339
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:PPEak?....1339
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:PPEak?....1339
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:PPEak?....1339
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:RMS?....1340
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:RMS?....1340
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:RMS?....1340
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:STDDev?....1340
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:STDDev?....1340
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:STDDev?....1340
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:WFMCount?....1340
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:WFMCount?....1340
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:WFMCount?....1340

---

**P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer[:ACTual]?**
**P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer[:ACTual]?**
**P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult[:ACTual]?**

Returns the current result value for the total input and output power and total efficiency.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<Actual> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:AVERage?**
**P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:AVERage?**
**P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:AVERage?**

Returns the average value for the total input and output power and total efficiency if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<Average> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:EVTCount?**
**P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:EVTCount?**
**P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:EVTCount?**

Returns the number of calculated measurement results for the total input and output power and total efficiency if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<Count> Range: 0 to 4000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:NPEak?**
**P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:NPEak?**
**P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:NPEak?**

Returns the negative peak value (minimum) for the total input and output power and total efficiency if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<MinPeak> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:PPEak?**
**P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:PPEak?**
**P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:PPEak?**

Returns the npositive peak value (maximum) for the total input and output power and total efficiency if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<MaxPeak> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:RMS?**
**P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:RMS?**
**P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:RMS?**

Returns the RMS value for the total input and output power and total efficiency if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<RMS> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:STDDev?**
**P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:STDDev?**
**P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:STDDev?**

Returns the standard deviation for the total input and output power and total efficiency if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<StdDev> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0

**Usage:** Query only
Asynchronous command

**P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:WFMCount?**
**P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:WFMCount?**
**P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:WFMCount?**

Returns the number of calculated waveforms for the total input and output power and total efficiency if statistics are enabled.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Return values:**
<WaveformsCount> Range: 0 to 4000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

### 18.17.4.5 3-phase power sources

P3PHase<m>:EFFiciency:INPut<n>:TYPE ..... 1342
P3PHase<m>:EFFiciency:OUTPut<n>:TYPE ..... 1342
P3PHase<m>:EFFiciency:INPut<n>:SOURce:WIRing ..... 1342
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:WIRing ..... 1342
P3PHase<m>:QUALity:SOURce:WIRing ..... 1342
P3PHase<m>:HARMonics:SOURce:WIRing ..... 1342
P3PHase<m>:QUALity:SOURce:INOut ..... 1342
P3PHase<m>:HARMonics:SOURce:INOut ..... 1342
P3PHase<m>:EFFiciency:INPut<n>:SOURce:LINes:W2V2a ..... 1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:LINes:W2V2a ..... 1343
P3PHase<m>:QUALity:SOURce:LINes:W2V2a ..... 1343
P3PHase<m>:HARMonics:SOURce:LINes:W2V2a ..... 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W1V1a ..... 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W2V2a ..... 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W3V3a ..... 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W3VN3a ..... 1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W1V1a ..... 1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W2V2a ..... 1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W3V3a ..... 1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W3VN3a ..... 1343
P3PHase<m>:QUALity:SOURce:CURRent:W2V2a ..... 1343
P3PHase<m>:QUALity:SOURce:CURRent:W3V3a ..... 1343
P3PHase<m>:QUALity:SOURce:CURRent:W3VN3a ..... 1343
P3PHase<m>:HARMonics:SOURce:CURRent:W2V2a ..... 1343
P3PHase<m>:HARMonics:SOURce:CURRent:W3V3a ..... 1343
P3PHase<m>:HARMonics:SOURce:CURRent:W3VN3a ..... 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W1V1a ..... 1344
P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W2V2a ..... 1344
P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W3V3a ..... 1344
P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W3VN3a ..... 1344
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W1V1a ..... 1344
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W2V2a ..... 1344
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W3V3a ..... 1344
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W3VN3a ..... 1344
P3PHase<m>:QUALity:SOURce:VOLTage:W2V2a ..... 1344
P3PHase<m>:QUALity:SOURce:VOLTage:W3V3a ..... 1344
P3PHase<m>:QUALity:SOURce:VOLTage:W3VN3a ..... 1344
P3PHase<m>:HARMonics:SOURce:VOLTage:W2V2a ..... 1344
P3PHase<m>:HARMonics:SOURce:VOLTage:W3V3a ..... 1344
P3PHase<m>:HARMonics:SOURce:VOLTage:W3VN3a ..... 1344
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CYCLe:SOURce ..... 1344
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CYCLe:SOURce ..... 1344
P3PHase<m>:QUALity:SOURce:CYCLe[:SOURce] ..... 1344
P3PHase<m>:HARMonics:SOURce:CYCLe[:SOURce] ..... 1344
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CYCLe:FILTer ..... 1345
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CYCLe:FILTer ..... 1345
P3PHase<m>:QUALity:SOURce:CYCLe:FILTer ..... 1345
P3PHase<m>:HARMonics:SOURce:CYCLe:FILTer ..... 1345

P3PHase<m>:EFFiciency:GATE....1345
P3PHase<m>:QUALity:GATE....1345

**P3PHase<m>:EFFiciency:INPut<n>:TYPE** <InputType>
**P3PHase<m>:EFFiciency:OUTPut<n>:TYPE** <InputType>

Selects the type of the current flow: AC or DC.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<InputType> DC | AC
*RST: AC

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 420

**P3PHase<m>:EFFiciency:INPut<n>:SOURce:WIRing** <Wiring>
**P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:WIRing** <Wiring>
**P3PHase<m>:QUALity:SOURce:WIRing** <Wiring>
**P3PHase<m>:HARMonics:SOURce:WIRing** <Wiring>

Configures how your probes are connected to the three-phase device. It allows you to visualize the connections for proper setup. If you can only measure on four channels, select the "W2V2A" wiring.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Wiring> W1V1A | W2V2A | W3V3A | W3VN3A
*RST: W3VN3A

**Usage:** Asynchronous command

**Manual operation:** See "Wiring" on page 420

**P3PHase<m>:QUALity:SOURce:INOut** <InputOutput>
**P3PHase<m>:HARMonics:SOURce:INOut** <InputOutput>

Configures whether the oscilloscope is configured to receive signals (input) for analysis, or to generate signals output) as a source for testing.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<InputOutput> INPut | OUTPut
*RST: INPut

**Usage:** Asynchronous command

**Manual operation:** See "Input/output" on page 421

---

**P3PHase<m>:EFFiciency:INPut<n>:SOURce:LINes:W2V2a** <InputSelcLines>
**P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:LINes:W2V2a** <OutputSelcLines>
**P3PHase<m>:QUALity:SOURce:LINes:W2V2a** <Lines>
**P3PHase<m>:HARMonics:SOURce:LINes:W2V2a** <LineType>

Select the configuration for your probe setup.

Selects the gate that is used for limiting the range of the 3-phase power quality measurement.

This setting is only necessary, if you have configured `P3PHase<m>:HARMonics:SOURce:WIRing` =`W2V2A`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<LineType> V12LV32L | V13LV23L | V21LV31L | V12LV23L

**Usage:** Asynchronous command

**Manual operation:** See "Select lines" on page 421

---

**P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W1V1a** <Sources>
**P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W2V2a** <Sources>
**P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W3V3a** <Sources>
**P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W3VN3a** <Sources>
**P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W1V1a** <Sources>
**P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W2V2a** <Sources>
**P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W3V3a** <Sources>
**P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W3VN3a** <Sources>
**P3PHase<m>:QUALity:SOURce:CURRent:W2V2a** <Sources>
**P3PHase<m>:QUALity:SOURce:CURRent:W3V3a** <Sources>
**P3PHase<m>:QUALity:SOURce:CURRent:W3VN3a** <Sources>
**P3PHase<m>:HARMonics:SOURce:CURRent:W2V2a** <Sources>
**P3PHase<m>:HARMonics:SOURce:CURRent:W3V3a** <Sources>
**P3PHase<m>:HARMonics:SOURce:CURRent:W3VN3a** <Sources>

Selects the current source waveform. Analog channels and math waveforms can be used.

Provide the sources as comma-separated values.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<Sources> C1 | C2 | ... | C8 | M1 | M2 | ... | M8 | R1 | R2 | ... | R8

**Usage:** Asynchronous command

**Manual operation:** See "Current" on page 422

---

**P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W1V1a** <Sources>
**P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W2V2a** <Sources>
**P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W3V3a** <Sources>
**P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W3VN3a** <Sources>
**P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W1V1a** <Sources>
**P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W2V2a** <Sources>
**P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W3V3a** <Sources>
**P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W3VN3a** <Sources>
**P3PHase<m>:QUALity:SOURce:VOLTage:W2V2a** <Sources>
**P3PHase<m>:QUALity:SOURce:VOLTage:W3V3a** <Sources>
**P3PHase<m>:QUALity:SOURce:VOLTage:W3VN3a** <Sources>
**P3PHase<m>:HARMonics:SOURce:VOLTage:W2V2a** <Sources>
**P3PHase<m>:HARMonics:SOURce:VOLTage:W3V3a** <Sources>
**P3PHase<m>:HARMonics:SOURce:VOLTage:W3VN3a** <Sources>

Selects the voltage source waveform. Analog channels and math waveforms can be used.

Provide the sources as comma-separated values.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<Sources> C1 | C2 | ... | C8 | M1 | M2 | ... | M8 | R1 | R2 | ... | R8

**Usage:** Asynchronous command

**Manual operation:** See "Voltage" on page 422

---

**P3PHase<m>:EFFiciency:INPut<n>:SOURce:CYCLe:SOURce** <CycleSource>
**P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CYCLe:SOURce** <CycleSource>
**P3PHase<m>:QUALity:SOURce:CYCLe[:SOURce]** <CycleSource>
**P3PHase<m>:HARMonics:SOURce:CYCLe[:SOURce]** <CycleSource>

Selects the channel (voltage source) to serve as the timing reference for data calculations across all sources when the periods are not aligned on the x-axis.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<CycleSource> C1 | C2 | ... | C8 | M1 | M2 | ... | M8 | R1 | R2 | ... | R8

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 423

**P3PHase<m>:EFFiciency:INPut<n>:SOURce:CYCLe:FILTer**
<DigFilterLowpassCutOffFreq>
**P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CYCLe:FILTer**
<DigFilterLowpassCutOffFreq>
**P3PHase<m>:QUALity:SOURce:CYCLe:FILTer** <DigFilterLowpassCutOffFreq>
**P3PHase<m>:HARMonics:SOURce:CYCLe:FILTer** <DigFilterLowpassCutOffFreq>

Configures a bandwidth limitation.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<DigFilterLowpassCutOffFreq> Range: 100 to 4000000000
Increment: 1000
*RST: 10000000
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Cut-off" on page 423

**P3PHase<m>:EFFiciency:GATE** <Gate>
**P3PHase<m>:QUALity:GATE** <Gate>

Selects the gate that is used for limiting the range of the 3-phase power quality and efficiency measurements.

Enable the gate before you assign a measurement to it (`GATE<g>:ENABle` =`ON`).

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<Gate> Number of the gate to be used

**Usage:** Asynchronous command

### 18.17.4.6 3-phase power waveform

P3PHase<m>:QUALity:DISPlay:POWer<n>:WAVeform[:ENABle] 1346
P3PHase<m>:HARMonics:DISPlay:POWer<n>:WAVeform[:ENABle] 1346
P3PHase<m>:EFFiciency:DISPlay:WAVeform:INPut:TOTal:POWer[:ENABle] 1346
P3PHase<m>:EFFiciency:DISPlay:WAVeform:OUTPut:TOTal:POWer[:ENABle] 1346
P3PHase<m>:QUALity:DISPlay:TOTal:POWer:WAVeform[:ENABle] 1346
P3PHase<m>:HARMonics:DISPlay:TOTal:POWer:WAVeform[:ENABle] 1346
P3PHase<m>:QUALity:POWer<n>:VERTical:OFFSet 1346
P3PHase<m>:HARMonics:POWer<n>:VERTical:OFFSet 1346
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:VERTical:OFFSet 1347
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:VERTical:OFFSet 1347
P3PHase<m>:QUALity:TOTal:POWer:VERTical:OFFSet 1347
P3PHase<m>:HARMonics:TOTal:POWer:VERTical:OFFSet 1347
P3PHase<m>:QUALity:POWer<n>:VERTical:SCALe:MODE 1347

P3PHase<m>:HARMonics:POWer<n>:VERTical:SCALe:MODE....1347
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:VERTical:SCALe:MODE....1348
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:VERTical:SCALe:MODE....1348
P3PHase<m>:QUALity:TOTal:POWer:VERTical:SCALe:MODE....1348
P3PHase<m>:HARMonics:TOTal:POWer:VERTical:SCALe:MODE....1348
P3PHase<m>:QUALity:POWer<n>:VERTical:SCALe[:VALue]....1348
P3PHase<m>:HARMonics:POWer<n>:VERTical:SCALe[:VALue]....1348
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:VERTical:SCALe[:VALue]....1348
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:VERTical:SCALe[:VALue]....1348
P3PHase<m>:QUALity:TOTal:POWer:VERTical:SCALe[:VALue]....1348
P3PHase<m>:HARMonics:TOTal:POWer:VERTical:SCALe[:VALue]....1348

---

**P3PHase<m>:QUALity:DISPlay:POWer<n>:WAVeform[:ENABle]** <State>
**P3PHase<m>:HARMonics:DISPlay:POWer<n>:WAVeform[:ENABle]** <State>

Displays or hides the specified power waveform.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<State> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Display individual phases" on page 424

---

**P3PHase<m>:EFFiciency:DISPlay:WAVeform:INPut:TOTal:POWer[:ENABle]** <State>
**P3PHase<m>:EFFiciency:DISPlay:WAVeform:OUTPut:TOTal:POWer[:ENABle]** <State>
**P3PHase<m>:QUALity:DISPlay:TOTal:POWer:WAVeform[:ENABle]** <State>
**P3PHase<m>:HARMonics:DISPlay:TOTal:POWer:WAVeform[:ENABle]** <State>

Displays or hides the total power waveform.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<State> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Display" on page 425

---

**P3PHase<m>:QUALity:POWer<n>:VERTical:OFFSet** <VerticalOffset>
**P3PHase<m>:HARMonics:POWer<n>:VERTical:OFFSet** <VerticalOffset>

Sets the vertical offset of the specified power waveform if the relevant `P3PHase<m>:*:POWer:VERTical:SCALe:MODE` command is set to `MANual`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<VerticalOffset> Range: -100000000000000 to 100000000000000
Increment: 0.01
*RST: 0
Default unit: div

**Usage:** Asynchronous command

---

**P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:VERTical:OFFSet** <VerticalOffset>
**P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:VERTical:OFFSet** <VerticalOffset>
**P3PHase<m>:QUALity:TOTal:POWer:VERTical:OFFSet** <VerticalOffset>
**P3PHase<m>:HARMonics:TOTal:POWer:VERTical:OFFSet** <VerticalOffset>

Sets the vertical offset of the total power waveform if the relevant `P3PHase<m>:*:POWer:VERTical:SCALe:MODE` command is set to `MANual`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<VerticalOffset> Range: -100000000000000 to 100000000000000
Increment: 0.01
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Offset" on page 425

---

**P3PHase<m>:QUALity:POWer<n>:VERTical:SCALe:MODE** <VertSclMd>
**P3PHase<m>:HARMonics:POWer<n>:VERTical:SCALe:MODE** <VertSclMd>

Selects the scaling mode for display of the specific power waveform.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<VertSclMd> MANual | AUTO
*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 424

**P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:VERTical:SCALe:MODE** <VertSclMd>
**P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:VERTical:SCALe:MODE** <VertSclMd>
**P3PHase<m>:QUALity:TOTal:POWer:VERTical:SCALe:MODE** <VertSclMd>
**P3PHase<m>:HARMonics:TOTal:POWer:VERTical:SCALe:MODE** <VertSclMd>

Selects the scaling mode for display of the total power waveform.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**
<VertSclMd> MANual | AUTO
*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 425

**P3PHase<m>:QUALity:POWer<n>:VERTical:SCALe[:VALue]** <VerticalScale>
**P3PHase<m>:HARMonics:POWer<n>:VERTical:SCALe[:VALue]** <VerticalScale>

Sets the vertical scale of the specific power waveform if the relevant `P3PHase<m>:*:POWer:VERTical:SCALe:MODE` command is set to `MANual`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Parameters:**
<VerticalScale> Range: 1E-12 to 100000000000000
Increment: 1E-05
*RST: 0.5
Default unit: VA

**Usage:** Asynchronous command

**Manual operation:** See "Scale" on page 425

**P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:VERTical:SCALe[:VALue]** <VerticalScale>
**P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:VERTical:SCALe[:VALue]** <VerticalScale>
**P3PHase<m>:QUALity:TOTal:POWer:VERTical:SCALe[:VALue]** <VerticalScale>
**P3PHase<m>:HARMonics:TOTal:POWer:VERTical:SCALe[:VALue]** <VerticalScale>

Sets the vertical scale of the total power waveform if the relevant `P3PHase<m>:*:POWer:VERTical:SCALe:MODE` command is set to `MANual`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Parameters:**

<VerticalScale>
Range: 1E-12 to 100000000000000
Increment: 1E-05
*RST: 0.5
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Scale" on page 425

### 18.17.4.7 3-phase reference level

P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:HYSTeresis....1350
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:HYSTeresis....1350
P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:HYSTeresis....1350
P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:HYSTeresis....1350
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:LLEVel....1350
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:LLEVel....1350
P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:LLEVel....1350
P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:LLEVel....1350
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:MLEVel....1351
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:MLEVel....1351
P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:MLEVel....1351
P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:MLEVel....1351
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:ULEVel....1351
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:ULEVel....1351
P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:ULEVel....1351
P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:ULEVel....1351
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:LMODe....1351
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:LMODe....1351
P3PHase<m>:QUALity:REFLevel<rl>:LMODe....1351
P3PHase<m>:HARMonics:REFLevel<rl>:LMODe....1351
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:HYSTeresis....1352
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:HYSTeresis....1352
P3PHase<m>:QUALity:REFLevel<rl>:RELative:HYSTeresis....1352
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:HYSTeresis....1352
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:LOWer....1352
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:LOWer....1352
P3PHase<m>:QUALity:REFLevel<rl>:RELative:LOWer....1352
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:LOWer....1352
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MIDDle....1353
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MIDDle....1353
P3PHase<m>:QUALity:REFLevel<rl>:RELative:MIDDle....1353
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:MIDDle....1353
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MODE....1353
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MODE....1353
P3PHase<m>:QUALity:REFLevel<rl>:RELative:MODE....1353
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:MODE....1353
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:UPPer....1354

P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:UPPer....1354
P3PHase<m>:QUALity:REFLevel<rl>:RELative:UPPer....1354
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:UPPer....1354

---

**P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:HYSTeresis** <HystAbs>
**P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:HYSTeresis** <HystAbs>
**P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:HYSTeresis** <HystAbs>
**P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:HYSTeresis** <HystAbs>

Defines a hysteresis for the middle reference level. A rise or fall from the middle reference value that does not exceed the hysteresis is rejected as noise.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

**Parameters:**
<HystAbs>
Range: 0 to 1E+26
Increment: 0.001
*RST: 0.005
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis" on page 414

---

**P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:LLEVel** <LowerLevel>
**P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:LLEVel** <LowerLevel>
**P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:LLEVel** <LowerLevel>
**P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:LLEVel** <LowerLevel>

Sets the lower reference level in absolute values. This is required, e.g., to determine a fall.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

**Parameters:**
<LowerLevel>
Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level, Lower level" on page 411

**P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:MLEVel** <MiddleLevel>
**P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:MLEVel** <MiddleLevel>
**P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:MLEVel** <MiddleLevel>
**P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:MLEVel** <MiddleLevel>

Sets the middle reference level in absolute values.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

**Parameters:**
<MiddleLevel> Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level, Lower level" on page 411

---

**P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:ULEVel** <UpperLevel>
**P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:ULEVel** <UpperLevel>
**P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:ULEVel** <UpperLevel>
**P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:ULEVel** <UpperLevel>

Sets the upper reference level in absolute values. This is required to determine a rise.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

**Parameters:**
<UpperLevel> Range: -1E+26 to 1E+26
Increment: 0.001
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level, Lower level" on page 411

---

**P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:LMODe** <LevelMode>
**P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:LMODe** <LevelMode>
**P3PHase<m>:QUALity:REFLevel<rl>:LMODe** <LevelMode>
**P3PHase<m>:HARMonics:REFLevel<rl>:LMODe** <LevelMode>

Defines if the reference level is set in absolute or relative values.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

**Parameters:**
<LevelMode> ABS | REL
*RST: REL

**Usage:** Asynchronous command

**Manual operation:** See "Level mode" on page 411

---

**P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:HYSTeresis** <HystRel>
**P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:HYSTeresis** <HystRel>
**P3PHase<m>:QUALity:REFLevel<rl>:RELative:HYSTeresis** <HystRel>
**P3PHase<m>:HARMonics:REFLevel<rl>:RELative:HYSTeresis** <HystRel>

Defines a hysteresis for the middle reference level. A rise or fall from the middle reference value that does not exceed the hysteresis is rejected as noise.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

**Parameters:**
<HystRel> Range: 0 to 50
Increment: 1
*RST: 10
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis" on page 414

---

**P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:LOWer** <LowRefLevRel>
**P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:LOWer** <LowRefLevRel>
**P3PHase<m>:QUALity:REFLevel<rl>:RELative:LOWer** <LowRefLevRel>
**P3PHase<m>:HARMonics:REFLevel<rl>:RELative:LOWer** <LowRefLevRel>

Sets the lower relative reference level if `P3PHase<m>:HARMonics:REFLevel<rl>:RELative:MODE` is set to `USER`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

**Parameters:**

<LowRefLevRel>
Range: -100 to 200
Increment: 1
*RST: 10
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level,Lower level" on page 413

---

**P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MIDDle** <MidRefLevRel>
**P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MIDDle** <MidRefLevRel>
**P3PHase<m>:QUALity:REFLevel<rl>:RELative:MIDDle** <MidRefLevRel>
**P3PHase<m>:HARMonics:REFLevel<rl>:RELative:MIDDle** <MidRefLevRel>

Sets the middle relative reference level if `P3PHase<m>:HARMonics:REFLevel<rl>:RELative:MODE` is set to `USER`.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

**Parameters:**

<MidRefLevRel>
Range: -100 to 200
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level,Lower level" on page 413

---

**P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MODE** <RelativeLevels>
**P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MODE** <RelativeLevels>
**P3PHase<m>:QUALity:REFLevel<rl>:RELative:MODE** <RelativeLevels>
**P3PHase<m>:HARMonics:REFLevel<rl>:RELative:MODE** <RelativeLevels>

The lower, middle and upper reference levels, defined as percentages of the high signal level.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

**Parameters:**

<RelativeLevels> FIVE | TEN | TWENty | USER

**FIVE**
5/50/95

**TEN**
10/50/90

**TWENty**
20/50/80

**USER**
Set the reference levels to individual values with `P3PHase<m>:HARMonics:REFLevel<rl>:RELative:LOWer`, `P3PHase<m>:HARMonics:REFLevel<rl>:RELative:MIDDle`, and `P3PHase<m>:HARMonics:REFLevel<rl>:RELative:UPPer`.

*RST: TEN

**Usage:** Asynchronous command

**Manual operation:** See "Relative levels" on page 412

---

**P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:UPPer** <UppRefLevRel>
**P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:UPPer** <UppRefLevRel>
**P3PHase<m>:QUALity:REFLevel<rl>:RELative:UPPer** <UppRefLevRel>
**P3PHase<m>:HARMonics:REFLevel<rl>:RELative:UPPer** <UppRefLevRel>

Sets the upper relative reference level if `P3PHase<m>:HARMonics:REFLevel<rl>:RELative:MODE` is set to `USER`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<rl> Irrelevant, omit the suffix.

**Parameters:**
<UppRefLevRel>
Range: -100 to 200
Increment: 1
*RST: 90
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Middle level,Lower level" on page 413

### 18.17.4.8 3-phase power analysis data

This section describes the commands for exporting 3-phase power analysis measurement data.

P3PHase<m>:QUALity:POWer<n>:DATA:HEADer? ..... 1355
P3PHase<m>:HARMonics:POWer<n>:DATA:HEADer? ..... 1355
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:DATA:HEADer? ..... 1355
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:DATA:HEADer? ..... 1355
P3PHase<m>:QUALity:TOTal:POWer:DATA:HEADer? ..... 1355

P3PHase<m>:HARMonics:TOTal:POWer:DATA:HEADer?....1355
P3PHase<m>:QUALity:POWer<n>:DATA[:VALues]?.... 1356
P3PHase<m>:HARMonics:POWer<n>:DATA[:VALues]?.... 1356
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:DATA[:VALues]?....1356
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:DATA[:VALues]?.... 1356
P3PHase<m>:QUALity:TOTal:POWer:DATA[:VALues]?....1356
P3PHase<m>:HARMonics:TOTal:POWer:DATA[:VALues]?.... 1356

**P3PHase<m>:QUALity:POWer<n>:DATA:HEADer?**
**P3PHase<m>:HARMonics:POWer<n>:DATA:HEADer?**

Returns the header of the 3-phase power analysis waveform data for the individual phases. The header contains the attributes of the waveform.

For power harmonics measurements, data is only available if `P3PHase<m>:HARMonics:STANdard` is set to `ENC` or `END`.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Return values:**

<XStart> 1. header value: start time in s

<XStop> 2. header value: end time in s

<RecordLength> 3. header value: record length of the waveform in samples

<ValsPerSmp> 4. header value: number of values per sample. For 3-phase power quality measurements, the value is 1.

**Usage:** Query only
Asynchronous command

**P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:DATA:HEADer?**
**P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:DATA:HEADer?**
**P3PHase<m>:QUALity:TOTal:POWer:DATA:HEADer?**
**P3PHase<m>:HARMonics:TOTal:POWer:DATA:HEADer?**

Returns the header of the 3-phase power analysis waveform data for the total phase. The header contains the attributes of the waveform.

For power harmonics measurements, data is only available if `P3PHase<m>:HARMonics:STANdard` is set to `ENC` or `END`.

**Suffix:**

<m> 1 to 3, index of the power analysis measurement

**Return values:**

<XStart> 1. header value: start time in s

<XStop> 2. header value: end time in s

<RecordLength> 3. header value: record length of the waveform in samples

<ValsPerSmp> 4. header value: number of values per sample. For 3-phase power quality measurements, the value is 1.

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:QUALity:POWer<n>:DATA[:VALues]?** [<Offset>],[<Length>]
**P3PHase<m>:HARMonics:POWer<n>:DATA[:VALues]?** [<Offset>],[<Length>]

Returns the individual phase data of the 3-phase power analysis waveform points for transmission from the instrument to the controlling computer.

Without parameters, the complete waveform is retrieved. Using the offset and length parameters, data can be retrieved in smaller portions, which makes the command faster. If you send only one parameter, it is interpreted as offset, and the data is retrieved from offset to the end of the waveform.

To set the export format, use `FORMat[:DATA]`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

<n> 1...3, index of the individual phase

**Query parameters:**
<Offset> Number of offset waveform points to be skipped.

<Length> Number of waveform points to be retrieved.

**Usage:** Query only
Asynchronous command

---

**P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:DATA[:VALues]?** [<Offset>], [<Length>]
**P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:DATA[:VALues]?** [<Offset>], [<Length>]
**P3PHase<m>:QUALity:TOTal:POWer:DATA[:VALues]?** [<Offset>],[<Length>]
**P3PHase<m>:HARMonics:TOTal:POWer:DATA[:VALues]?** [<Offset>],[<Length>]

Returns the total phase data of the 3-phase power analysis waveform points for transmission from the instrument to the controlling computer.

Without parameters, the complete waveform is retrieved. Using the offset and length parameters, data can be retrieved in smaller portions, which makes the command faster. If you send only one parameter, it is interpreted as offset, and the data is retrieved from offset to the end of the waveform.

To set the export format, use `FORMat[:DATA]`.

**Suffix:**
<m> 1 to 3, index of the power analysis measurement

**Query parameters:**
<Offset> Number of offset waveform points to be skipped.

<Length> Number of waveform points to be retrieved.

**Usage:** Query only
Asynchronous command

## 18.17.5 Digital voltmeter

This section describes the commands for setting up and querying the results of the digital voltmeter.

METer:BANDwidth ..... 1357
METer:DVMeter<m>:ENABle? ..... 1357
METer:DVMeter<m>:ACRMs:ENABle ..... 1358
METer:DVMeter<m>:ACRMs:RESult? ..... 1358
METer:DVMeter<m>:DC:ENABle ..... 1358
METer:DVMeter<m>:DC:RESult? ..... 1358
METer:DVMeter<m>:DCRMs:ENABle ..... 1359
METer:DVMeter<m>:DCRMs:RESult? ..... 1359
METer:MTIMe ..... 1359

---

**METer:BANDwidth** <Bandwidth>

Sets the filter bandwidth.

**Parameters:**
<Bandwidth> B20M | B10M | B5M | B2M | B1M | B500 | B200
`B20M`: 20 MHz
`B10M`: 10 MHz
`B5M`: 5 MHz
`B2M`: 2 MHz
`B1M`: 1 MHz
`B500`: 500 kHz
`B200`: 200 kHz
*RST: B20M

**Usage:** Asynchronous command

**Manual operation:** See "Filter bandwidth" on page 445

---

**METer:DVMeter<m>:ENABle?**

Queries the state of the voltmeter for the respective channel.

**Suffix:**
<m> 1 to 8, index of the analog channel

**Return values:**
<State> OFF | ON
*RST: OFF

**Usage:** Query only
Asynchronous command

**METer:DVMeter<m>:ACRMs:ENABle** <ACRMSResults_St>

Enables the AC RMS voltmeter measurement for the respective channel.

**Suffix:**
<m> 1 to 8, index of the analog channel

**Parameters:**
<ACRMSResults_St> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Measurements" on page 445

**METer:DVMeter<m>:ACRMs:RESult?**

Returns the result of the AC RMS voltmeter measurement.

**Suffix:**
<m> 1 to 8, index of the analog channel

**Return values:**
<Results_ACRMS> Range: -1E+26 to 1E+26
Increment: 0.1
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Measurements" on page 445

**METer:DVMeter<m>:DC:ENABle** <DCResults_St>

Enables the DC voltmeter measurement for the respective channel.

**Suffix:**
<m> 1 to 8, index of the analog channel

**Parameters:**
<DCResults_St> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Measurements" on page 445

**METer:DVMeter<m>:DC:RESult?**

Returns the result of the DC voltmeter measurement.

**Suffix:**
<m> 1 to 8, index of the analog channel

**Return values:**
<Results_DC> Range: -1E+26 to 1E+26
Increment: 0.1
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Measurements" on page 445

---

**METer:DVMeter<m>:DCRMs:ENABle** <DCRMSResults_St>

Enables the DC RMS voltmeter measurement for the respective channel.

**Suffix:**
<m> 1 to 8, index of the analog channel

**Parameters:**
<DCRMSResults_St> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Measurements" on page 445

---

**METer:DVMeter<m>:DCRMs:RESult?**

Returns the result of the DC RMS voltmeter measurement.

**Suffix:**
<m> 1 to 8, index of the analog channel

**Return values:**
<Results_DCRMS> Range: -1E+26 to 1E+26
Increment: 0.1
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Measurements" on page 445

---

**METer:MTIMe** <MeasTime>

Sets the measurement time. The time begins in the moment that a measurement is enabled.

**Parameters:**
<MeasTime> Range: 0.05 to 1
Increment: 0.05
*RST: 0.5
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Measurement time" on page 445

### 18.17.6 XY-plot

XY<m>:SWAP ..... 1360
XY<m>[:STATe] ..... 1360
XY<m>:YSOurce ..... 1360
XY<m>:XSOurce ..... 1361

**XY<m>:SWAP**

Replaces the source of the x-axis with the source of the y-axis and vice versa.

**Suffix:**
<m> 1 to 4, index of the XY-plot

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Swap XY" on page 447

**XY<m>[:STATe]** <State>

Activates an XY-waveform.

**Suffix:**
<m> 1 to 4, index of the XY-plot

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Display" on page 446

**XY<m>:YSOurce** <YSource>

Defines the signal source that supplies the y-values of the XY-diagram.

**Suffix:**
<m> 1 to 4, index of the XY-plot

**Parameters:**
<YSource> NONE | C1 | C2 | ... | C8 | M1 | M2 | ... | M8 | R1 | R2 | ... | R8 | TRK1 | TRK2 | TRK3 | ... | TRK16

**Usage:** Asynchronous command

**Manual operation:** See "X-source, Y-source" on page 447

**XY<m>:XSOurce** <XSource>

Defines the signal source that supplies the x-values of the XY-diagram.

**Suffix:**
<m> 1 to 4, index of the XY-plot

**Parameters:**
<XSource> NONE | C1 | C2 | ... | C8 | M1 | M2 | ... | M8 | R1 | R2 | ... | R8 | TRK1 | TRK2 | TRK3 | ... | TRK16

**Usage:** Asynchronous command

**Manual operation:** See "X-source, Y-source" on page 447

## 18.18 Protocols


- Configuration settings for all serial protocols ..... 1361
- SPI (option R&S MXO3-K510) ..... 1367
- QUAD-SPI (option R&S MXO3-K510) ..... 1401
- OCTO-SPI (option R&S MXO3-K510) ..... 1433
- I²C (option R&S MXO3-K510) ..... 1464
- I3C (option R&S MXO3-K550) ..... 1500
- UART / RS-232 (option R&S MXO3- K510) ..... 1533
- NRZ Clocked (option R&S MXO3-K510) ..... 1566
- NRZ Unclocked (option R&S MXO3-K510) ..... 1595
- Manchester (option R&S MXO3-K510) ..... 1625
- CAN (option R&S MXO3-K520) ..... 1655
- LIN (option R&S MXO3-K520) ..... 1703
- SENT (option R&S MXO3-K520) ..... 1736
- ARINC 429 (option R&S MXO3-K530) ..... 1772
- MIL-1553 (option R&S MXO3-K530) ..... 1800
- SpaceWire (option R&S MXO3-K530) ..... 1829
- SPMI (option R&S MXO3-K550) ..... 1856
- RFFE (option R&S MXO3-K550) ..... 1887
- 10BASE-T1S (option R&S MXO3-K560) ..... 1920
- USB 1.1 (option R&S MXO3-K570) ..... 1951
- USB 2.0 (option R&S MXO3-K570) ..... 1982


### 18.18.1 Configuration settings for all serial protocols

#### 18.18.1.1 General settings


SBUS<sb>[:STATe] ..... 1362
SBUS<sb>:TYPE ..... 1362
SBUS<sb>:FORMat ..... 1363
SBUS<sb>:RESult ..... 1364

SBUS<sb>:RMSBus....1364
SBUS<sb>:THReshold....1364
SBUS<sb>:ZCOupling....1365

**SBUS<sb>[:STATe]** <ProtocolState>

Enables the decoding of the specified bus.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ProtocolState> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 490

**SBUS<sb>:TYPE** <Protocol Type>

Selects the bus type for analysis. The type of buses available depends on the installed options.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Protocol Type> I2C | I3C | SPI | QSPI | QSPI | UART | CAN | LIN | SENT | TNOS | MILS1553 | MILStd | ARIN429 | ARINc429 | SWIR | SWIRe | RFFE | SPMI | MANC | NRZC | NRZU | MDIO | DALI | USB1 | USB2 | USBHspeed | USBLow | USBFull

`SPI`: SPI protocol, see Section 18.18.2, "SPI (option R&S MXO3-K510)", on page 1367.
`OSPI`: Octo SPI protocol, see Section 18.18.4, "OCTO-SPI (option R&S MXO3-K510)", on page 1433.
`QSPI`: Quad SPI protocol, see Section 18.18.3, "QUAD-SPI (option R&S MXO3-K510)", on page 1401.
`I2C`: I²C protocol, see Section 18.18.5, "I²C (option R&S MXO3-K510)", on page 1464.
`I3C`: I3C protocol, see Section 18.18.6, "I3C (option R&S MXO3-K550)", on page 1500.
`UART`: UART protocol, see Section 18.18.7, "UART / RS-232 (option R&S MXO3- K510)", on page 1533.
`NRZC | MDIO`: NRZ clocked protocol, see Section 18.18.8, "NRZ Clocked (option R&S MXO3-K510)", on page 1566.
`NRZU`: NRZ unclocked protocol, see Section 18.18.9, "NRZ Unclocked (option R&S MXO3-K510)", on page 1595.
`MANC | DALI`: Manchester protocol, see Section 18.18.10, "Manchester (option R&S MXO3-K510)", on page 1625
`CAN`: CAN protocol, see Section 18.18.11, "CAN (option R&S MXO3-K520)", on page 1655.

`LIN`: LIN protocol, see Section 18.18.12, "LIN (option R&S MXO3-K520)", on page 1703.
`SENT`: SENT protocol, see Section 18.18.13, "SENT (option R&S MXO3-K520)", on page 1736.
`ARIN429 | ARINc429`: ARINC 429 protocol, see Section 18.18.14, "ARINC 429 (option R&S MXO3-K530)", on page 1772.
`MILS1553 | MILStd`: MIL-1553 protocol, see Section 18.18.15, "MIL-1553 (option R&S MXO3-K530)", on page 1800.
`SWIR | SWIRe`: SpaceWire protocol, see Section 18.18.16, "SpaceWire (option R&S MXO3-K530)", on page 1829.
`SPMI`: SPMI protocol, see Section 18.18.17, "SPMI (option R&S MXO3-K550)", on page 1856.
`RFFE`: RFFE protocol, see Section 18.18.18, "RFFE (option R&S MXO3-K550)", on page 1887.
`TNOS`: Ethernet 10BASE-T1S protocol, see Section 18.18.19, "10BASE-T1S (option R&S MXO3-K560)", on page 1920.
`USB1 | USBLow | USBFull`: USB 1.1 protocol, see Section 18.18.20, "USB 1.1 (option R&S MXO3-K570)", on page 1951
`USB2 | USBHspeed`: USB 2.0 protocol, see Section 18.18.21, "USB 2.0 (option R&S MXO3-K570)", on page 1982

**Usage:** Asynchronous command

**Manual operation:** See "Protocol type" on page 490

---

**SBUS<sb>:FORMat** <DataFormat>

Sets the number format for decoded data values of the indicated serial bus. It defines the format in the decode table, and in the combs of the decoded signal on the screen.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DataFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | SYMB | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, e.g. 8 bits signed ranges from -128 to +127 decimal

**DEC=USIG**
Unsigned, e.g. 8 bits unsigned ranges from 0 to 255 decimal

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Data format" on page 493

---

**SBUS<sb>:RESult** <ShwResTbl>

Enables a table with decoded data of the serial signal. The function requires the option for the analyzed protocol.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ShwResTbl> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show decode table" on page 493

---

**SBUS<sb>:RMSBus**

Deletes the specified serial bus.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add serial bus, Delete serial bus" on page 490

---

**SBUS<sb>:THReshold** <ShwThresLines>

If enabled, the threshold lines are displayed in the diagram.

The label of the threshold line is set according to the signal that it is referring to, see also Show threshold lines.

**Suffix:**
<sb> 1..4

**Parameters:**
<ShwThresLines> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show threshold lines" on page 491

---

**SBUS<sb>:ZCOupling** <ZoomCoupling>

If enabled, the protocol decode zoom and result table are synchronized.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ZoomCoupling> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Zoom coupling" on page 494

### 18.18.1.2 Export results settings

SBUS<sb>:EXPResult:DETail ..... 1365
SBUS<sb>:EXPResult:SAVE ..... 1365
SBUS<sb>:EXPResult:TIME ..... 1366
SBUS<sb>:EXPResult:EXTension ..... 1366
SBUS<sb>:EXPResult:PATH ..... 1366

---

**SBUS<sb>:EXPResult:DETail** <IncludeDetails>

If enabled, includes the detailed results for all frames in the export result file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<IncludeDetails> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Include details" on page 509

---

**SBUS<sb>:EXPResult:SAVE** <FileName>

Saves the selected results to the indicated file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Example:**
```
SBUS1:EXPResult:SAVE
'/home/storage/userData/export_10base-t1s.csv'
```

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 510

---

**SBUS<sb>:EXPResult:TIME** <IncludeTiming>

If enabled, includes the frame timing in the export result file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<IncludeTiming> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Include timing" on page 509

---

**SBUS<sb>:EXPResult:EXTension** <FileType>

Selects the file format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<FileType> HTML | CSV | XML | PY

**HTML**
Hypertext markup language

**CSV**
Comma-separated values

**XML**
Extensible markup language

**PY**
Python

*RST: HTML

**Usage:** Asynchronous command

**Manual operation:** See "File type" on page 509

---

**SBUS<sb>:EXPResult:PATH** <Path>

Sets the path where the protocol export files are stored. On the instrument, the default path is `/home/storage/userData/Protocol`. You can create subfolders in this folder.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Path> String parameter

**Usage:** Asynchronous command

**Manual operation:** See "Save as" on page 510

## 18.18.2 SPI (option R&S MXO3-K510)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.


- Configuration ..... 1367
- Filter ..... 1374
- Format ..... 1380
- Hardware trigger ..... 1389
- Software trigger ..... 1391
- Decode results ..... 1396


### 18.18.2.1 Configuration


SBUS<sb>:SPI:BORDer ..... 1368
SBUS<sb>:SPI:FRCondition ..... 1368
SBUS<sb>:SPI:CSELect:HYSTeresis ..... 1368
SBUS<sb>:SPI:CSELect:POLarity ..... 1369
SBUS<sb>:SPI:CSELect:SOURce ..... 1369
SBUS<sb>:SPI:CSELect:THReshold ..... 1369
SBUS<sb>:SPI:MISO:HYSTeresis ..... 1369
SBUS<sb>:SPI:MISO:POLarity ..... 1370
SBUS<sb>:SPI:MISO:SOURce ..... 1370
SBUS<sb>:SPI:MISO:THReshold ..... 1370
SBUS<sb>:SPI:MOSI:HYSTeresis ..... 1371
SBUS<sb>:SPI:MOSI:POLarity ..... 1371
SBUS<sb>:SPI:MOSI:SOURce ..... 1371
SBUS<sb>:SPI:MOSI:THReshold ..... 1371
SBUS<sb>:SPI:SCLK:HYSTeresis ..... 1372
SBUS<sb>:SPI:SCLK:SOURce ..... 1372
SBUS<sb>:SPI:SCLK:THReshold ..... 1372
SBUS<sb>:SPI:TIMeout ..... 1373
SBUS<sb>:SPI:WSIZe ..... 1373
SBUS<sb>:SPI:MISO:POSition ..... 1373
SBUS<sb>:SPI:MISO:SCALe ..... 1373
SBUS<sb>:SPI:MOSI:POSition ..... 1374
SBUS<sb>:SPI:MOSI:SCALe ..... 1374

**SBUS<sb>:SPI:BORDer** <BitOrder>

Selects the bit order, which determines if the data of the messages starts with MSB (most significant bit) or LSB (least significant bit).

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<BitOrder> LSBF | MSBF
*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 517

**SBUS<sb>:SPI:FRCondition** <FrameCondition>

Defines the start of a frame. A frame contains a number of successive words, at least one word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<FrameCondition> CS | CLKTimeout

**CS**
Start and end of the frame is defined by the active state of the sub select signal, see `SBUS<sb>:SPI:CSELect:POLarity`.

**CLKTimeout**
Defines a timeout on the clock line SCLK as limiter between two frames. The timeout condition is used for SPI connections without a CS line.

*RST: CS

**Usage:** Asynchronous command

**SBUS<sb>:SPI:CSELect:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the CS channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 518

**SBUS<sb>:SPI:CSELect:POLarity** <CSPolarity>

Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CSPolarity> ACTLow | ACTHigh
*RST: ACTLow

**Usage:** Asynchronous command

**Manual operation:** See "Polarity: MOSI, MISO, CS" on page 517

**SBUS<sb>:SPI:CSELect:SOURce** <CSSource>

Sets the input channel of the CS line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CSSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "SCLK,MOSI,MISO,CS" on page 516

**SBUS<sb>:SPI:CSELect:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 518

**SBUS<sb>:SPI:MISO:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the MISO channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 518

---

**SBUS<sb>:SPI:MISO:POLarity** <MISOPolarity>

Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MISOPolarity> ACTLow | ACTHigh
*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Polarity: MOSI, MISO, CS" on page 517

---

**SBUS<sb>:SPI:MISO:SOURce** <MISOSource>

Sets the input channel of the MISO line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MISOSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "SCLK,MOSI,MISO,CS" on page 516

---

**SBUS<sb>:SPI:MISO:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 518

**SBUS<sb>:SPI:MOSI:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the MOSI channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 518

**SBUS<sb>:SPI:MOSI:POLarity** <MOSIPolarity>

Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MOSIPolarity> ACTLow | ACTHigh
*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Polarity: MOSI, MISO, CS" on page 517

**SBUS<sb>:SPI:MOSI:SOURce** <MOSISource>

Sets the input channel of the MOSI line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MOSISource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "SCLK,MOSI,MISO,CS" on page 516

**SBUS<sb>:SPI:MOSI:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 518

---

**SBUS<sb>:SPI:SCLK:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the SCLK channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 518

---

**SBUS<sb>:SPI:SCLK:SOURce** <SCLKSource>

Sets the input channel of the SCLK line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SCLKSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "SCLK,MOSI,MISO,CS" on page 516

---

**SBUS<sb>:SPI:SCLK:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 518

**SBUS<sb>:SPI:TIMeout** <ClockTimeout>

Sets the minimum clock idle time if a timeout on the clock line SCLK is used as limiter between two frames.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ClockTimeout>
Range: 5E-08 to 10
Increment: 1E-06
*RST: 0.001
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Timeout" on page 517

**SBUS<sb>:SPI:WSIZe** <WordLength>

Sets the word length (or symbol size), which is the number of bits in a message. The maximum word length is 32 bit.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<WordLength>
Range: 4 to 32
Increment: 1
*RST: 8

**Usage:** Asynchronous command

**Manual operation:** See "Word length" on page 517

**SBUS<sb>:SPI:MISO:POSition** <SPI MISO position>

Sets the vertical position of the MISO signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SPI MISO position>

**Usage:** Asynchronous command

**SBUS<sb>:SPI:MISO:SCALe** <SPI MISO scale>

Set the vertical scale of the MISO signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SPI MISO scale>

**Usage:** Asynchronous command

---

**SBUS<sb>:SPI:MOSI:POSition** <SPI MISO position>

Sets the vertical position of the MOSI signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SPI MISO position>

**Usage:** Asynchronous command

---

**SBUS<sb>:SPI:MOSI:SCALe** <SPI MOSI scale>

Set the vertical scale of the MOSI signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SPI MOSI scale>

**Usage:** Asynchronous command

### 18.18.2.2 Filter

There are two commands for each parameter, that you can use for defining the SPI settings.

For example, to set the *Frame type =MISO > Field =MISO >01100* value you can use one of the following commands:

- `SBUS:SPI:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:SPI:FILTer:DMIN "MISO", "MISO", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:SPI:FILTer:CHKall ... 1375
SBUS<sb>:SPI:FILTer:CLR ... 1375
SBUS<sb>:SPI:FILTer:INVert ... 1375
SBUS<sb>:SPI:FILTer:RST ... 1376
SBUS<sb>:SPI:FILTer:BIT ... 1376
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1376
SBUS<sb>:SPI:FILTer:DMAX ... 1376
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1376
SBUS<sb>:SPI:FILTer:DMIN ... 1377
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1377

SBUS<sb>:SPI:FILTer:DOPerator....1377
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1377
SBUS<sb>:SPI:FILTer:ERENable....1377
SBUS<sb>:SPI:FILTer:ERRor<n>:ENABle....1377
SBUS<sb>:SPI:FILTer:FIENable....1378
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1378
SBUS<sb>:SPI:FILTer:FRENable....1378
SBUS<sb>:SPI:FILTer:FRAMe<fr>:ENABle....1378
SBUS<sb>:SPI:FILTer:IMAX....1378
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1378
SBUS<sb>:SPI:FILTer:IMIN....1379
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1379
SBUS<sb>:SPI:FILTer:IOPerator....1379
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1379

### SBUS<sb>:SPI:FILTer:CHKall

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 519

### SBUS<sb>:SPI:FILTer:CLR

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 519

### SBUS<sb>:SPI:FILTer:INVert

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 519

**SBUS<sb>:SPI:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 519

**SBUS<sb>:SPI:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:SPI:FILTer:BIT?** <Bit>
**SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 520

**SBUS<sb>:SPI:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:SPI:FILTer:DMAX?** <Data>
**SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 520

**SBUS<sb>:SPI:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:SPI:FILTer:DMIN?** <Data>
**SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 520

**SBUS<sb>:SPI:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:SPI:FILTer:DOPerator?** <Operator>
**SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 520

**SBUS<sb>:SPI:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:SPI:FILTer:ERENable?** <Enabler>
**SBUS<sb>:SPI:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the error

**Parameters:**
<Enable> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 521

---

**SBUS<sb>:SPI:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:SPI:FILTer:FIENable?** <Enabler>
**SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 520

---

**SBUS<sb>:SPI:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:SPI:FILTer:FRENable?** <Enabler>
**SBUS<sb>:SPI:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 520

---

**SBUS<sb>:SPI:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:SPI:FILTer:IMAX?** <Data>
**SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 520

---

**SBUS<sb>:SPI:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:SPI:FILTer:IMIN?** <Data>
**SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 520

---

**SBUS<sb>:SPI:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:SPI:FILTer:IOPerator?** <Operator>
**SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 520

### 18.18.2.3 Format

This section describes the commands to specify and query frame formats for the "SPI" protocol.

SBUS<sb>:SPI:FORMat:SELect....1380
SBUS<sb>:SPI:FORMat:ADDFrame....1381
SBUS<sb>:SPI:FORMat:LOAD....1381
SBUS<sb>:SPI:FORMat:SAVE....1381
SBUS<sb>:SPI:FORMat:CLR....1381
SBUS<sb>:SPI:FORMat:FCOunt?....1382
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ADDField....1382
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLDCount?....1382
SBUS<sb>:SPI:FORMat:FRAMe<fr>:NAME....1382
SBUS<sb>:SPI:FORMat:FRAMe<fr>:COLor....1383
SBUS<sb>:SPI:FORMat:FRAMe<fr>:TYPE....1383
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID1Field....1383
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID2Field....1383
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID3Field....1383
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID1Value....1384
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID2Value....1384
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID3Value....1384
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:NAME....1384
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:COLor....1384
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1385
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1385
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1385
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1386
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1386
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1387
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1387
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1388
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1388
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:PARA....1388
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:PARB....1388

---

**SBUS<sb>:SPI:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 522

**SBUS<sb>:SPI:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 522

**SBUS<sb>:SPI:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 523

**SBUS<sb>:SPI:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 523

**SBUS<sb>:SPI:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 522

**SBUS<sb>:SPI:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 522

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 524

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 523

---

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 523

---

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 523

---

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 523

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 523

---

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 524

---

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 524

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<FirstBit> Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 524

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 525

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 525

---

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 525

---

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 525

---

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 526

---

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 526

---

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 526

---

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 526

---

**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)

- Parameter B: Offset

Available if `SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<ParameterB> Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 526

### 18.18.2.4 Hardware trigger

Trigger commands for serial buses affect the current trigger source. Therefore, set the trigger source to the required serial bus with `TRIGger:EVENt<ev>:SOURce` before sending the bus-specific trigger commands.

TRIGger:SBHW:SPI:TYPE....1389
TRIGger:SBHW:SPI:PALignment....1390
TRIGger:SBHW:SPI:FCONdition....1390
TRIGger:SBHW:SPI:DPOSition....1390
TRIGger:SBHW:SPI:DMINpattern....1390

---

**TRIGger:SBHW:SPI:TYPE** <Type>

Selects the trigger type for SPI analysis.

**Parameters:**
<Type> FRSTart | FRENd | MOSI | MISO

**FRSTart**
Triggers on the beginning of the frame.

**FRENd**
Triggers on the end of the frame.

**MOSI**
Triggers on a specified data pattern in that is expected on the MOSI line.

**MISO**
Triggers on a specified data pattern in that is expected on the MISO line.

*RST: FRSTart

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 527

**TRIGger:SBHW:SPI:PALignment** <DataAlignment>

Defines how the specified data pattern is searched.

**Parameters:**

<DataAlignment> WORD | BIT

**WORD**
The pattern is matched only at word boundaries.

**BIT**
Bit-by-bit: the pattern can start at any position in the message.

*RST: WORD

**Usage:** Asynchronous command

**Manual operation:** See "Search mode" on page 528

**TRIGger:SBHW:SPI:FCONdition** <DataOperator>

Selects the operator for the MISO and MOSI pattern.

**Parameters:**

<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 528

**TRIGger:SBHW:SPI:DPOSition** <DataPosition>

Sets the number of bits or words to be ignored before the first bit or word of interest. The effect is defined by `TRIGger:SBHW:SPI:PALignment`.

**Parameters:**

<DataPosition> Range: 1 to 4096
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Position" on page 528

**TRIGger:SBHW:SPI:DMINpattern** <DataPattern>

Specifies a data bit pattern, or sets the start value of a pattern range.

**Parameters:**

<DataPattern>

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 528

### 18.18.2.5 Software trigger

There are two commands for each parameter, that you can use for defining the SPI settings.

For example, to set the *Frame type =MISO > Field =MISO >01100* value you can use one of the following commands:

- `TRIGger:SBSW:SPI:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:SBSW:SPI:DMIN "MISO", "MISO", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:SPI:CHKall ... 1391
TRIGger:SBSW:SPI:CLR ... 1392
TRIGger:SBSW:SPI:INVert ... 1392
TRIGger:SBSW:SPI:RST ... 1392
TRIGger:SBSW:SPI:FRENable ... 1392
TRIGger:SBSW:SPI:FRAMe<fr>:ENABle ... 1392
TRIGger:SBSW:SPI:BIT ... 1392
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:BIT ... 1392
TRIGger:SBSW:SPI:DMAX ... 1393
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DMAX ... 1393
TRIGger:SBSW:SPI:DMIN ... 1393
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DMIN ... 1393
TRIGger:SBSW:SPI:DOPerator ... 1394
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DOPerator ... 1394
TRIGger:SBSW:SPI:FIENable ... 1394
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:ENABle ... 1394
TRIGger:SBSW:SPI:IMAX ... 1394
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IMAX ... 1394
TRIGger:SBSW:SPI:IMIN ... 1395
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IMIN ... 1395
TRIGger:SBSW:SPI:IOPerator ... 1395
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IOPerator ... 1395
TRIGger:SBSW:SPI:ERENable ... 1395
TRIGger:SBSW:SPI:ERRor<m>:ENABle ... 1395
SBUS<sb>:SPI:SWTindex? ... 1396
SBUS<sb>:SPI:SWTTime? ... 1396

---

**TRIGger:SBSW:SPI:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 529

**TRIGger:SBSW:SPI:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 529

**TRIGger:SBSW:SPI:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 529

**TRIGger:SBSW:SPI:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 529

**TRIGger:SBSW:SPI:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:SPI:FRENable?** <Enabler>
**TRIGger:SBSW:SPI:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 530

**TRIGger:SBSW:SPI:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:SPI:BIT?** <Bit>
**TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 530

---

**TRIGger:SBSW:SPI:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SPI:DMAX?** <Data>
**TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 530

---

**TRIGger:SBSW:SPI:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SPI:DMIN?** <Data>
**TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 530

**TRIGger:SBSW:SPI:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:SPI:DOPerator?** <Operator>
**TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 530

**TRIGger:SBSW:SPI:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:SPI:FIENable?** <Enabler>
**TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 530

**TRIGger:SBSW:SPI:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SPI:IMAX?** <Data>
**TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 530

---

**TRIGger:SBSW:SPI:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SPI:IMIN?** <Data>
**TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 530

---

**TRIGger:SBSW:SPI:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:SPI:IOPerator?** <Operator>
**TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 530

---

**TRIGger:SBSW:SPI:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:SPI:ERENable?** <Enabler>
**TRIGger:SBSW:SPI:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 531

---

**SBUS<sb>:SPI:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPI:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.2.6 Decode results

SBUS<sb>:SPI:FCOunt? 1397
SBUS<sb>:SPI:FRAMe<fr>:BITRate? 1397
SBUS<sb>:SPI:FRAMe<fr>:DATA? 1397
SBUS<sb>:SPI:FRAMe<fr>:STARt? 1398
SBUS<sb>:SPI:FRAMe<fr>:STATus? 1398
SBUS<sb>:SPI:FRAMe<fr>:STOP? 1398

SBUS<sb>:SPI:FRAMe<fr>:WCOunt? 1399
SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:FMISo? 1399
SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:FMOSi? 1399
SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:MISO? 1400
SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:MOSI? 1400
SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:STARt? 1400
SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:STOP? 1401

### SBUS<sb>:SPI:FCOunt?

Returns the number of decoded frames.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

### SBUS<sb>:SPI:FRAMe<fr>:BITRate?

Returns the bit rate of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<PrimaryBitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

### SBUS<sb>:SPI:FRAMe<fr>:DATA?

Returns the data words of the specified frame in comma-separated values.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameData> Comma-separated sequence of integer values (N, L1, R1,…, LN, RN). N is the number of word pairs in the frame, and {L1,R1}…{LN,RN} are the value pairs. The values Lx and Rx are associated with the MOSI and the MISO channel, respectively. If a channel is disabled, an empty value is returned.

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPI:FRAMe<fr>:STARt?**

Returns the start time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPI:FRAMe<fr>:STATus?**

Returns the overall state of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameState> OK | VOID | INComplete | LENGth

`OK`: the frame is valid.
`VOID`: the frame is empty.
`INComplete`: INComplete word. The word is cut off at the left or right side of the acquisition.
`LENGth`: The frame that was found has an unexpected length: there are either too few or too many bits in this frame.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPI:FRAMe<fr>:STOP?**

Returns the end time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPI:FRAMe<fr>:WCOunt?**

Returns the number of words in the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameWordCount> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:FMISo?**

Returns the formatted value of the specified word on the MISO line.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<w> Index of the word

**Return values:**

<FormattedMISOVal>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:FMOSi?**

Returns the formatted value of the specified word on the MOSI line.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<w> Index of the word

**Return values:**
<FormattedMOSIVal>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:MISO?**

Returns the data value of the specified word on the MISO line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<w> Index of the word

**Return values:**
<MISOValue> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:MOSI?**

Returns the data value of the specified word on the MOSI line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<w> Index of the word

**Return values:**
<MOSIValue> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:STARt?**

Returns the start time of the specified data word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<w> Index of the word

**Return values:**

<FrameWordStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:STOP?**

Returns the end time of the specified data word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<w> Index of the word

**Return values:**

<FrameWordStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

## 18.18.3 QUAD-SPI (option R&S MXO3-K510)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.

- Configuration....1401
- Filter....1411
- Format....1416
- Software trigger....1421
- Decode results....1427

### 18.18.3.1 Configuration

SBUS<sb>:QSPI:CSEL:HYSTeresis....1402
SBUS<sb>:QSPI:CSEL:POLarity....1402
SBUS<sb>:QSPI:CSEL:SOURce....1403
SBUS<sb>:QSPI:CSEL:THReshold....1403
SBUS<sb>:QSPI:INSTruction....1403
SBUS<sb>:QSPI:IOONe:HYSTeresis....1403
SBUS<sb>:QSPI:IOONe:POLarity....1404
SBUS<sb>:QSPI:IOONe:POSition....1404

SBUS<sb>:QSPI:IOONe:SCALe ..... 1404
SBUS<sb>:QSPI:IOONe:SOURce ..... 1404
SBUS<sb>:QSPI:IOONe:THReshold ..... 1405
SBUS<sb>:QSPI:IOTHree:HYSTeresis ..... 1405
SBUS<sb>:QSPI:IOTHree:POLarity ..... 1405
SBUS<sb>:QSPI:IOTHree:POSition ..... 1406
SBUS<sb>:QSPI:IOTHree:SCALe ..... 1406
SBUS<sb>:QSPI:IOTHree:SOURce ..... 1406
SBUS<sb>:QSPI:IOTHree:THReshold ..... 1406
SBUS<sb>:QSPI:IOTWo:HYSTeresis ..... 1407
SBUS<sb>:QSPI:IOTWo:POLarity ..... 1407
SBUS<sb>:QSPI:IOTWo:POSition ..... 1407
SBUS<sb>:QSPI:IOTWo:SCALe ..... 1407
SBUS<sb>:QSPI:IOTWo:SOURce ..... 1408
SBUS<sb>:QSPI:IOTWo:THReshold ..... 1408
SBUS<sb>:QSPI:IOZero:HYSTeresis ..... 1408
SBUS<sb>:QSPI:IOZero:POLarity ..... 1408
SBUS<sb>:QSPI:IOZero:POSition ..... 1409
SBUS<sb>:QSPI:IOZero:SCALe ..... 1409
SBUS<sb>:QSPI:IOZero:SOURce ..... 1409
SBUS<sb>:QSPI:IOZero:THReshold ..... 1409
SBUS<sb>:QSPI:SCLK:HYSTeresis ..... 1410
SBUS<sb>:QSPI:SCLK:POLarity ..... 1410
SBUS<sb>:QSPI:SCLK:SOURce ..... 1410
SBUS<sb>:QSPI:SCLK:THReshold ..... 1411

---

**SBUS<sb>:QSPI:CSEL:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the CS channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "CS/SCLK/IO<n> hysteresis" on page 537

---

**SBUS<sb>:QSPI:CSEL:POLarity** <CSPolarity>

Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CSPolarity> ACTLow | ACTHigh
*RST: ACTLow

**Usage:** Asynchronous command

**Manual operation:** See "Polarity: CS, IO<n>" on page 535

---

**SBUS<sb>:QSPI:CSEL:SOURce** <CSSource>

Sets the input channel of the CS line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CSSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "CS, SCLK, IO n" on page 535

---

**SBUS<sb>:QSPI:CSEL:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "CS/SCLK/IO<n> threshold" on page 536

---

**SBUS<sb>:QSPI:INSTruction** <Instruction>

Selects the instruction mode that defines how many lanes are used to transmit data.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Instruction> SINGle | DUAL | QUAD
*RST: SINGle

**Usage:** Asynchronous command

**Manual operation:** See "Instruction" on page 535

---

**SBUS<sb>:QSPI:IOONe:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the IO1 line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "CS/SCLK/IO<n> hysteresis" on page 537

---

**SBUS<sb>:QSPI:IOONe:POLarity** <IO1Polarity>

Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<IO1Polarity> ACTLow | ACTHigh
*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Polarity: CS, IO<n>" on page 535

---

**SBUS<sb>:QSPI:IOONe:POSition** <QUADSPI IO1 position>

Sets the vertical position of the IO1 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<QUADSPI IO1 position>

**Usage:** Asynchronous command

---

**SBUS<sb>:QSPI:IOONe:SCALe** <QUADSPI IO1 scale>

Set the vertical scale of the IO1 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<QUADSPI IO1 scale>

**Usage:** Asynchronous command

---

**SBUS<sb>:QSPI:IOONe:SOURce** <IO1Source>

Sets the input channel of the IO 1 line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<IO1Source> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "CS, SCLK, IO n" on page 535

---

**SBUS<sb>:QSPI:IOONe:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "CS/SCLK/IO<n> threshold" on page 536

---

**SBUS<sb>:QSPI:IOTHree:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the IO0 line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "CS/SCLK/IO<n> hysteresis" on page 537

---

**SBUS<sb>:QSPI:IOTHree:POLarity** <IO3Polarity>

Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<IO3Polarity> ACTLow | ACTHigh
*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Polarity: CS, IO<n>" on page 535

**SBUS<sb>:QSPI:IOTHree:POSition** <QUADSPI IO3 position>

Sets the vertical position of the IO3 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<QUADSPI IO3 position>

**Usage:** Asynchronous command

**SBUS<sb>:QSPI:IOTHree:SCALe** <QUADSPI IO3 scale>

Set the vertical scale of the IO3 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<QUADSPI IO3 scale>

**Usage:** Asynchronous command

**SBUS<sb>:QSPI:IOTHree:SOURce** <IO3Source>

Sets the input channel of the IO 3 line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<IO3Source> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "CS, SCLK, IO n" on page 535

**SBUS<sb>:QSPI:IOTHree:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "CS/SCLK/IO<n> threshold" on page 536

**SBUS<sb>:QSPI:IOTWo:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the IO2 line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "CS/SCLK/IO<n> hysteresis" on page 537

**SBUS<sb>:QSPI:IOTWo:POLarity** <IO2Polarity>

Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<IO2Polarity> ACTLow | ACTHigh
*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Polarity: CS, IO<n>" on page 535

**SBUS<sb>:QSPI:IOTWo:POSition** <QUADSPI IO2 position>

Sets the vertical position of the IO2 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<QUADSPI IO2 position>

**Usage:** Asynchronous command

**SBUS<sb>:QSPI:IOTWo:SCALe** <QUADSPI IO2 scale>

Set the vertical scale of the IO2 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<QUADSPI IO2 scale>

**Usage:** Asynchronous command

**SBUS<sb>:QSPI:IOTWo:SOURce** <IO2Source>

Sets the input channel of the IO 2 line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<IO2Source> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "CS, SCLK, IO n" on page 535

**SBUS<sb>:QSPI:IOTWo:THReshold** <Threshold>

Sets a value for the hysteresis for the IO2 line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "CS/SCLK/IO<n> threshold" on page 536

**SBUS<sb>:QSPI:IOZero:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the IO0 line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "CS/SCLK/IO<n> hysteresis" on page 537

**SBUS<sb>:QSPI:IOZero:POLarity** <IO0Polarity>

Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<IO0Polarity> ACTLow | ACTHigh
*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Polarity: CS, IO<n>" on page 535

---

**SBUS<sb>:QSPI:IOZero:POSition** <QUADSPI IO0 position>

Sets the vertical position of the IO0 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<QUADSPI IO0 position>

**Usage:** Asynchronous command

---

**SBUS<sb>:QSPI:IOZero:SCALe** <QUADSPI IO0 scale>

Set the vertical scale of the IO0 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<QUADSPI IO0 scale>

**Usage:** Asynchronous command

---

**SBUS<sb>:QSPI:IOZero:SOURce** <IO0Source>

Sets the input channel of the IO 0 line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<IO0Source> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "CS, SCLK, IO n" on page 535

---

**SBUS<sb>:QSPI:IOZero:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "CS/SCLK/IO<n> threshold" on page 536

---

**SBUS<sb>:QSPI:SCLK:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the SCLK channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "CS/SCLK/IO<n> hysteresis" on page 537

---

**SBUS<sb>:QSPI:SCLK:POLarity** <SCLKPolarity>

Sets the polarity for the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SCLKPolarity> FALLing | RISing
*RST: RISing

**Usage:** Asynchronous command

**Manual operation:** See "SCLK SDR Polarity" on page 535

---

**SBUS<sb>:QSPI:SCLK:SOURce** <SCLKSource>

Sets the input channel of the SCLK clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SCLKSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "CS, SCLK, IO n" on page 535

**SBUS<sb>:QSPI:SCLK:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "CS/SCLK/IO<n> threshold" on page 536

### 18.18.3.2 Filter

There are two commands for each parameter, that you can use for defining the QUAD SPI settings.

For example, to set the *Frame type =Data > Field =Addr >01100* value you can use one of the following commands:

- `SBUS:QSPI:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:QSPI:FILTer:DMIN "Data", "Addr", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:QSPI:FILTer:CHKall....1412
SBUS<sb>:QSPI:FILTer:CLR....1412
SBUS<sb>:QSPI:FILTer:INVert....1412
SBUS<sb>:QSPI:FILTer:RST....1412
SBUS<sb>:QSPI:FILTer:BIT....1412
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:BIT....1412
SBUS<sb>:QSPI:FILTer:DMAX....1413
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1413
SBUS<sb>:QSPI:FILTer:DMIN....1413
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1413
SBUS<sb>:QSPI:FILTer:DOPerator....1414
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1414
SBUS<sb>:QSPI:FILTer:IMAX....1414
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1414
SBUS<sb>:QSPI:FILTer:IMIN....1414
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1414
SBUS<sb>:QSPI:FILTer:IOPerator....1415
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1415
SBUS<sb>:QSPI:FILTer:ERENable....1415
SBUS<sb>:QSPI:FILTer:ERRor<n>:ENABle....1415
SBUS<sb>:QSPI:FILTer:FIENable....1415
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1415
SBUS<sb>:QSPI:FILTer:FRENable....1416
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:ENABle....1416

**SBUS<sb>:QSPI:FILTer:CHKall**

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 538

**SBUS<sb>:QSPI:FILTer:CLR**

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 538

**SBUS<sb>:QSPI:FILTer:INVert**

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 538

**SBUS<sb>:QSPI:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 538

**SBUS<sb>:QSPI:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:QSPI:FILTer:BIT?** <Bit>
**SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 538

---

**SBUS<sb>:QSPI:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:QSPI:FILTer:DMAX?** <Data>
**SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 538

---

**SBUS<sb>:QSPI:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:QSPI:FILTer:DMIN?** <Data>
**SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 538

**SBUS<sb>:QSPI:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:QSPI:FILTer:DOPerator?** <Operator>
**SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 538

---

**SBUS<sb>:QSPI:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:QSPI:FILTer:IMAX?** <Data>
**SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max>
Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 538

---

**SBUS<sb>:QSPI:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:QSPI:FILTer:IMIN?** <Data>
**SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 538

---

**SBUS<sb>:QSPI:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:QSPI:FILTer:IOPerator?** <Operator>
**SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 538

---

**SBUS<sb>:QSPI:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:QSPI:FILTer:ERENable?** <Enabler>
**SBUS<sb>:QSPI:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<n> index of the error

**Parameters:**

<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 540

---

**SBUS<sb>:QSPI:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:QSPI:FILTer:FIENable?** <Enabler>
**SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 538

---

**SBUS<sb>:QSPI:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:QSPI:FILTer:FRENable?** <Enabler>
**SBUS<sb>:QSPI:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

### 18.18.3.3 Format

SBUS<sb>:QSPI:OPCode:APPend ... 1417
SBUS<sb>:QSPI:OPCode:DALL ... 1417
SBUS<sb>:QSPI:OPCode:DELete ... 1417
SBUS<sb>:QSPI:OPCode:RESet ... 1417
SBUS<sb>:QSPI:LDOPcode ... 1417
SBUS<sb>:QSPI:OPCode:SIZE? ... 1418
SBUS<sb>:QSPI:OPCode:ITEM<n>:ADBYtes ... 1418
SBUS<sb>:QSPI:OPCode:ITEM<n>:ADLanes ... 1418
SBUS<sb>:QSPI:OPCode:ITEM<n>:ALT ... 1419
SBUS<sb>:QSPI:OPCode:ITEM<n>:CODE ... 1419
SBUS<sb>:QSPI:OPCode:ITEM<n>:DATA ... 1419
SBUS<sb>:QSPI:OPCode:ITEM<n>:DDR ... 1420
SBUS<sb>:QSPI:OPCode:ITEM<n>:DMCYcles ... 1420
SBUS<sb>:QSPI:OPCode:ITEM<n>:DTLanes ... 1420
SBUS<sb>:QSPI:OPCode:ITEM<n>:NAME ... 1421
SBUS<sb>:QSPI:SVOP ... 1421

**SBUS<sb>:QSPI:OPCode:APPend**

Adds a new item to the opcode list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add" on page 540

**SBUS<sb>:QSPI:OPCode:DALL**

Deletes all currently defined opcodes from the list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Delete all" on page 541

**SBUS<sb>:QSPI:OPCode:DELete** <Index>

Deletes the opcode with the selected index.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<Index>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Delete" on page 540

**SBUS<sb>:QSPI:OPCode:RESet**

Resets the opcode fields to the predefined values.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Preset" on page 541

**SBUS<sb>:QSPI:LDOPcode** <FileName>

Loads an opcode file from the selected file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open, Save as" on page 541

---

**SBUS<sb>:QSPI:OPCode:SIZE?**

Sets the size of the opcode, hence the number of opcodes currently defined on the table in the "Format" tab of QUAD-SPI.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:QSPI:OPCode:ITEM<n>:ADBYtes** <AddressBytes>

Sets the address byte. It specifies the location in the flash memory where the operation (e.g., read, write) is performed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<AddressBytes> Range: 0 to 4
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Address Bytes" on page 543

---

**SBUS<sb>:QSPI:OPCode:ITEM<n>:ADLanes** <AddressLanes>

Selects how many lines or lanes are used to send the address bytes to the flash memory.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<AddressLanes> SINGle | DUAL | QUAD
*RST: SINGle

**Usage:** Asynchronous command

**Manual operation:** See "Address Lanes" on page 543

---

**SBUS<sb>:QSPI:OPCode:ITEM<n>:ALT** <Alt>

Enable, if an alternative field is available.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<Alt> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Alternative Field" on page 543

---

**SBUS<sb>:QSPI:OPCode:ITEM<n>:CODE** <Opcode>

Sets the opcode value.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<Opcode> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Opcode" on page 542

---

**SBUS<sb>:QSPI:OPCode:ITEM<n>:DATA** <Data>

Enable, if data is being transferred in the opcode operation.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<Data> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 543

---

**SBUS<sb>:QSPI:OPCode:ITEM<n>:DDR** <DDR>

If enabled, the data is transferred on both the rising and falling edges of the clock signal. This setting effectively doubles the data transfer rate compared to a single data rate (SDR), which transfers data only on one clock edge.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<DDR> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "DDR" on page 542

---

**SBUS<sb>:QSPI:OPCode:ITEM<n>:DMCYcles** <DummyCycles>

Sets the number of dummy cycles. Dummy cycles are clock cycles inserted after the address or other command sequences but before data transfer begins. These cycles allow the flash memory device additional time to perform internal operations or latch onto the correct data to ensure accurate read or write operations.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<DummyCycles> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Dummy Cycles" on page 543

---

**SBUS<sb>:QSPI:OPCode:ITEM<n>:DTLanes** <DataLanes>

Selects how many lanes are used for the data transfer.

The data lanes refer to the physical connections through which data is transmitted between the main (typically a microcontroller) and the sub (typically a flash memory device). QUADSPI can utilize multiple data lines to increase the speed and efficiency of data transfer.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<DataLanes> SINGle | DUAL | QUAD
*RST: SINGle

**Usage:** Asynchronous command

**Manual operation:** See "Data Lanes" on page 543

---

**SBUS<sb>:QSPI:OPCode:ITEM<n>:NAME** <Name>

Sets the name for the Opcode.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Name" on page 542

---

**SBUS<sb>:QSPI:SVOP** <FileName>

Saves the opcode file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open, Save as" on page 541

### 18.18.3.4 Software trigger

There are two commands for each parameter, that you can use for defining the QUAD SPI settings.

For example, to set the *Frame type =Data > Field =Addr >01100* value you can use one of the following commands:

- `TRIGger:SBSW:QSPI:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:SBSW:QSPI:DMIN "Data", "Addr", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:QSPI:CHKall....1422
TRIGger:SBSW:QSPI:CLR....1422
TRIGger:SBSW:QSPI:INVert....1422
TRIGger:SBSW:QSPI:RST....1423
TRIGger:SBSW:QSPI:FRENable....1423
TRIGger:SBSW:QSPI:FRAMe<fr>:ENABle....1423
TRIGger:SBSW:QSPI:BIT....1423
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:BIT....1423
TRIGger:SBSW:QSPI:DMAX....1424
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DMAX....1424
TRIGger:SBSW:QSPI:DMIN....1424
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DMIN....1424
TRIGger:SBSW:QSPI:DOPerator....1424
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DOPerator....1424
TRIGger:SBSW:QSPI:FIENable....1425
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:ENABle....1425
TRIGger:SBSW:QSPI:IMAX....1425
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IMAX....1425
TRIGger:SBSW:QSPI:IMIN....1426
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IMIN....1426
TRIGger:SBSW:QSPI:IOPerator....1426
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IOPerator....1426
TRIGger:SBSW:QSPI:ERENable....1426
TRIGger:SBSW:QSPI:ERRor<m>:ENABle....1426
SBUS<sb>:QSPI:SWTindex?....1427
SBUS<sb>:QSPI:SWTTime?....1427

### TRIGger:SBSW:QSPI:CHKall

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 544

### TRIGger:SBSW:QSPI:CLR

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 544

### TRIGger:SBSW:QSPI:INVert

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 544

---

**TRIGger:SBSW:QSPI:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 544

---

**TRIGger:SBSW:QSPI:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:QSPI:FRENable?** <Enabler>
**TRIGger:SBSW:QSPI:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 545

---

**TRIGger:SBSW:QSPI:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:QSPI:BIT?** <Bit>
**TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 545

**TRIGger:SBSW:QSPI:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:QSPI:DMAX?** <Data>
**TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 545

**TRIGger:SBSW:QSPI:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:QSPI:DMIN?** <Data>
**TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 545

**TRIGger:SBSW:QSPI:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:QSPI:DOPerator?** <Operator>
**TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 545

---

**TRIGger:SBSW:QSPI:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:QSPI:FIENable?** <Enabler>
**TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 545

---

**TRIGger:SBSW:QSPI:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:QSPI:IMAX?** <Data>
**TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 545

**TRIGger:SBSW:QSPI:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:QSPI:IMIN?** <Data>
**TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 545

---

**TRIGger:SBSW:QSPI:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:QSPI:IOPerator?** <Operator>
**TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 545

---

**TRIGger:SBSW:QSPI:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:QSPI:ERENable?** <Enabler>
**TRIGger:SBSW:QSPI:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 547

**SBUS<sb>:QSPI:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:QSPI:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.3.5 Decode results

SBUS<sb>:QSPI:FCOunt? 1428
SBUS<sb>:QSPI:FRAMe<fr>:ADDR? 1428
SBUS<sb>:QSPI:FRAMe<fr>:ALT? 1428
SBUS<sb>:QSPI:FRAMe<fr>:BITRate? 1428
SBUS<sb>:QSPI:FRAMe<fr>:DATA? 1429
SBUS<sb>:QSPI:FRAMe<fr>:FLDCount? 1429
SBUS<sb>:QSPI:FRAMe<fr>:FDATa? 1429
SBUS<sb>:QSPI:FRAMe<fr>:FLD<fl>:VALue? 1430
SBUS<sb>:QSPI:FRAMe<fr>:FLD<fl>:FVALue? 1430
SBUS<sb>:QSPI:FRAMe<fr>:FLD<fl>:LABel? 1430
SBUS<sb>:QSPI:FRAMe<fr>:FLD<fl>:STATe? 1431
SBUS<sb>:QSPI:FRAMe<fr>:ILBL? 1431
SBUS<sb>:QSPI:FRAMe<fr>:IVALue? 1431
SBUS<sb>:QSPI:FRAMe<fr>:TYPE? 1432

SBUS<sb>:QSPI:FRAMe<fr>:STARt? ..... 1432
SBUS<sb>:QSPI:FRAMe<fr>:STATus? ..... 1432
SBUS<sb>:QSPI:FRAMe<fr>:STOP? ..... 1433

### SBUS<sb>:QSPI:FCOunt?

Returns the number of decoded frames.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

### SBUS<sb>:QSPI:FRAMe<fr>:ADDR?

Returns the address of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameAddress> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

### SBUS<sb>:QSPI:FRAMe<fr>:ALT?

Retruns the value of the alternate byte for the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameAlt> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

### SBUS<sb>:QSPI:FRAMe<fr>:BITRate?

Returns the bit rate of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<PrimaryBitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:QSPI:FRAMe<fr>:DATA?**

Returns the data of the specified frame in comma-separated values.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

<Values>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:QSPI:FRAMe<fr>:FLDCount?**

Returns the field count for the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:QSPI:FRAMe<fr>:FDATa?**

Returns the foramtted data value of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameData>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:QSPI:FRAMe<fr>:FLD<fl>:VALue?**

Returns the value of the specified field of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:QSPI:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:QSPI:FRAMe<fr>:FLD<fl>:LABel?**

Returns the label of the selected field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:QSPI:FRAMe<fr>:FLD<fl>:STATe?**

Returns the state of the selected field in the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**

<State> OK | LENGth | OPCode | INComplete

**OK**
The field is valid.

**LENGth**
The field does not have the expected length.

**OPCode**
The opcode value is not found in your frame description list. See Section 14.3.4, "QUAD-SPI format configuration", on page 540.

**INComplete**
The field is not completely contained in the acquisition. The acquired part of the field is usually valid, but cannot always be trusted.

*RST: OK

**Usage:** Query only
Asynchronous command

**SBUS<sb>:QSPI:FRAMe<fr>:ILBL?**

Returns the label of the instruction.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameLabel>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:QSPI:FRAMe<fr>:IVALue?**

Returns the instruction mode value for the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrmInstruction> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:QSPI:FRAMe<fr>:TYPE?**

Returns the type of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameType>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:QSPI:FRAMe<fr>:STARt?**

Returns the start time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:QSPI:FRAMe<fr>:STATus?**

Returns the overall state of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameState> OK | LENGth | OPCode | INComplete
`OK`: the frame is valid.
`OPCode`: operation code.

INComplete: INComplete word. The word does not have the expected word length.
LENGth: The frame is not contained in the acquisition.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:QSPI:FRAMe<fr>:STOP?**

Returns the end time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

## 18.18.4 OCTO-SPI (option R&S MXO3-K510)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with SBUS<sb>[:STATe].
- The bus is set to the correct type with SBUS<sb>:TYPE.

- Configuration....1433
- Filter....1440
- Format....1446
- Software trigger....1451
- Decode results....1457

### 18.18.4.1 Configuration

SBUS<sb>:OSPI:CSEL:HYSTeresis....1435
SBUS<sb>:OSPI:CSEL:POLarity....1435
SBUS<sb>:OSPI:CSEL:SOURce....1435
SBUS<sb>:OSPI:CSEL:THReshold....1435
SBUS<sb>:OSPI:MODE....1436
SBUS<sb>:OSPI:CK:HYSTeresis....1436
SBUS<sb>:OSPI:CK:SOURce....1436
SBUS<sb>:OSPI:CK:THReshold....1437
SBUS<sb>:OSPI:IOZero:HYSTeresis....1437
SBUS<sb>:OSPI:IOONe:HYSTeresis....1437

SBUS<sb>:OSPI:IOTWo:HYSTeresis....1437
SBUS<sb>:OSPI:IOTHree:HYSTeresis....1437
SBUS<sb>:OSPI:IOFR:HYSTeresis....1437
SBUS<sb>:OSPI:IOFV:HYSTeresis....1437
SBUS<sb>:OSPI:IOSX:HYSTeresis....1437
SBUS<sb>:OSPI:IOSV:HYSTeresis....1437
SBUS<sb>:OSPI:IOZero:POLarity....1437
SBUS<sb>:OSPI:IOONe:POLarity....1437
SBUS<sb>:OSPI:IOTWo:POLarity....1437
SBUS<sb>:OSPI:IOTHree:POLarity....1437
SBUS<sb>:OSPI:IOFR:POLarity....1437
SBUS<sb>:OSPI:IOFV:POLarity....1437
SBUS<sb>:OSPI:IOSX:POLarity....1437
SBUS<sb>:OSPI:IOSV:POLarity....1437
SBUS<sb>:OSPI:IOZero:POSition....1438
SBUS<sb>:OSPI:IOONe:POSition....1438
SBUS<sb>:OSPI:IOTWo:POSition....1438
SBUS<sb>:OSPI:IOTHree:POSition....1438
SBUS<sb>:OSPI:IOFR:POSition....1438
SBUS<sb>:OSPI:IOFV:POSition....1438
SBUS<sb>:OSPI:IOSX:POSition....1438
SBUS<sb>:OSPI:IOSV:POSition....1438
SBUS<sb>:OSPI:IOZero:SCALe....1438
SBUS<sb>:OSPI:IOONe:SCALe....1438
SBUS<sb>:OSPI:IOTWo:SCALe....1438
SBUS<sb>:OSPI:IOTHree:SCALe....1438
SBUS<sb>:OSPI:IOFR:SCALe....1438
SBUS<sb>:OSPI:IOFV:SCALe....1438
SBUS<sb>:OSPI:IOSX:SCALe....1438
SBUS<sb>:OSPI:IOSV:SCALe....1438
SBUS<sb>:OSPI:IOZero:SOURce....1438
SBUS<sb>:OSPI:IOONe:SOURce....1438
SBUS<sb>:OSPI:IOTWo:SOURce....1438
SBUS<sb>:OSPI:IOTHree:SOURce....1438
SBUS<sb>:OSPI:IOFR:SOURce....1439
SBUS<sb>:OSPI:IOFV:SOURce....1439
SBUS<sb>:OSPI:IOSX:SOURce....1439
SBUS<sb>:OSPI:IOSV:SOURce....1439
SBUS<sb>:OSPI:IOZero:THReshold....1439
SBUS<sb>:OSPI:IOONe:THReshold....1439
SBUS<sb>:OSPI:IOTWo:THReshold....1439
SBUS<sb>:OSPI:IOTHree:THReshold....1439
SBUS<sb>:OSPI:IOFR:THReshold....1439
SBUS<sb>:OSPI:IOFV:THReshold....1439
SBUS<sb>:OSPI:IOSX:THReshold....1439
SBUS<sb>:OSPI:IOSV:THReshold....1439
SBUS<sb>:OSPI:DS:HYSTeresis....1439
SBUS<sb>:OSPI:DS:SOURce....1440
SBUS<sb>:OSPI:DS:THReshold....1440

**SBUS<sb>:OSPI:CSEL:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the CS channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "CS/CK/IO<n>/DS hysteresis" on page 554

**SBUS<sb>:OSPI:CSEL:POLarity** <CSPolarity>

Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CSPolarity> ACTLow | ACTHigh
*RST: ACTLow

**Usage:** Asynchronous command

**Manual operation:** See "Polarity: CS, IO<n>" on page 552

**SBUS<sb>:OSPI:CSEL:SOURce** <CSSource>

Sets the input channel of the CS line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CSSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "CS, SCLK, IO n" on page 552

**SBUS<sb>:OSPI:CSEL:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "CS/CK/IO<n>/DS threshold" on page 553

---

**SBUS<sb>:OSPI:MODE** <InstructionMd>

Selects the instruction mode that defines how many lanes are used to transmit data.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<InstructionMd> WR1S | WR2S | WR4S | WR8S | WR1D | WR2D | WR4D | WR8D

**WR1S | WR2S | WR4S | WR8S**
Number of lanes in single data rate mode

**WR1D | WR2D | WR4D | WR8D**
Number of lanes in dual data rate mode

*RST: WR1S

**Usage:** Asynchronous command

**Manual operation:** See "Instruction" on page 552

---

**SBUS<sb>:OSPI:CK:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "CS/CK/IO<n>/DS hysteresis" on page 554

---

**SBUS<sb>:OSPI:CK:SOURce** <CKSource>

Sets the input channel of the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CKSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "CS, SCLK, IO n" on page 552

---

**SBUS<sb>:OSPI:CK:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "CS/CK/IO<n>/DS threshold" on page 553

---

**SBUS<sb>:OSPI:IOZero:HYSTeresis** <Hysteresis>
**SBUS<sb>:OSPI:IOONe:HYSTeresis** <Hysteresis>
**SBUS<sb>:OSPI:IOTWo:HYSTeresis** <Hysteresis>
**SBUS<sb>:OSPI:IOTHree:HYSTeresis** <Hysteresis>
**SBUS<sb>:OSPI:IOFR:HYSTeresis** <Hysteresis>
**SBUS<sb>:OSPI:IOFV:HYSTeresis** <Hysteresis>
**SBUS<sb>:OSPI:IOSX:HYSTeresis** <Hysteresis>
**SBUS<sb>:OSPI:IOSV:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the IO<n> line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "CS/CK/IO<n>/DS hysteresis" on page 554

---

**SBUS<sb>:OSPI:IOZero:POLarity** <IO0Polarity>
**SBUS<sb>:OSPI:IOONe:POLarity** <IO1Polarity>
**SBUS<sb>:OSPI:IOTWo:POLarity** <IO2Polarity>
**SBUS<sb>:OSPI:IOTHree:POLarity** <IO3Polarity>
**SBUS<sb>:OSPI:IOFR:POLarity** <IO4Polarity>
**SBUS<sb>:OSPI:IOFV:POLarity** <IO5Polarity>
**SBUS<sb>:OSPI:IOSX:POLarity** <IO6Polarity>
**SBUS<sb>:OSPI:IOSV:POLarity** <IO7Polarity>

Selects if the transmitted signal for the respective line is active high (high = 1) or active low (low = 1).

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<IO7Polarity> ACTLow | ACTHigh
*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Polarity: CS, IO<n>" on page 552

---

**SBUS<sb>:OSPI:IOZero:POSition** <OCTOSPI IO0 position>
**SBUS<sb>:OSPI:IOONe:POSition** <OCTOSPI IO1 position>
**SBUS<sb>:OSPI:IOTWo:POSition** <OCTOSPI IO2 position>
**SBUS<sb>:OSPI:IOTHree:POSition** <OCTOSPI IO3 position>
**SBUS<sb>:OSPI:IOFR:POSition** <OCTOSPI IO4 position>
**SBUS<sb>:OSPI:IOFV:POSition** <OCTOSPI IO5 position>
**SBUS<sb>:OSPI:IOSX:POSition** <OCTOSPI IO6 position>
**SBUS<sb>:OSPI:IOSV:POSition** <OCTOSPI IO7 position>

Sets the vertical position of the IO<n> signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<OCTOSPI IO7 position>

**Usage:** Asynchronous command

---

**SBUS<sb>:OSPI:IOZero:SCALe** <OCTOSPI IO0 scale>
**SBUS<sb>:OSPI:IOONe:SCALe** <OCTOSPI IO1 scale>
**SBUS<sb>:OSPI:IOTWo:SCALe** <OCTOSPI IO2 scale>
**SBUS<sb>:OSPI:IOTHree:SCALe** <OCTOSPI IO3 scale>
**SBUS<sb>:OSPI:IOFR:SCALe** <OCTOSPI IO4 scale>
**SBUS<sb>:OSPI:IOFV:SCALe** <OCTOSPI IO5 scale>
**SBUS<sb>:OSPI:IOSX:SCALe** <OCTOSPI IO6 scale>
**SBUS<sb>:OSPI:IOSV:SCALe** <OCTOSPI IO7 scale>

Set the vertical scale of the IO<n> signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<OCTOSPI IO7 scale>

**Usage:** Asynchronous command

---

**SBUS<sb>:OSPI:IOZero:SOURce** <IO0Source>
**SBUS<sb>:OSPI:IOONe:SOURce** <IO1Source>
**SBUS<sb>:OSPI:IOTWo:SOURce** <IO2Source>
**SBUS<sb>:OSPI:IOTHree:SOURce** <IO3Source>

**SBUS<sb>:OSPI:IOFR:SOURce** <IO4Source>
**SBUS<sb>:OSPI:IOFV:SOURce** <IO5Source>
**SBUS<sb>:OSPI:IOSX:SOURce** <IO6Source>
**SBUS<sb>:OSPI:IOSV:SOURce** <IO7Source>

Sets the input channel of the IO<n> line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<IO7Source> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "CS, SCLK, IO n" on page 552

---

**SBUS<sb>:OSPI:IOZero:THReshold** <Threshold>
**SBUS<sb>:OSPI:IOONe:THReshold** <Threshold>
**SBUS<sb>:OSPI:IOTWo:THReshold** <Threshold>
**SBUS<sb>:OSPI:IOTHree:THReshold** <Threshold>
**SBUS<sb>:OSPI:IOFR:THReshold** <Threshold>
**SBUS<sb>:OSPI:IOFV:THReshold** <Threshold>
**SBUS<sb>:OSPI:IOSX:THReshold** <Threshold>
**SBUS<sb>:OSPI:IOSV:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "CS/CK/IO<n>/DS threshold" on page 553

---

**SBUS<sb>:OSPI:DS:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the data strobe line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "CS/CK/IO<n>/DS hysteresis" on page 554

**SBUS<sb>:OSPI:DS:SOURce** <DSSource>

Sets the input channel of the data strobe line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DSSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "CS, SCLK, IO n" on page 552

**SBUS<sb>:OSPI:DS:THReshold** <Threshold>

Sets a user-defined threshold value for the line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "CS/CK/IO<n>/DS threshold" on page 553

### 18.18.4.2 Filter

There are two commands for each parameter, that you can use for defining the OCTO-SPI settings.

For example, to set the *Frame type =Data* > *Field =Addr >01100* value you can use one of the following commands:

- `SBUS:OSPI:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:OSPI:FILTer:DMIN "Data", "Addr", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:OSPI:FILTer:CHKall....1441
SBUS<sb>:OSPI:FILTer:CLR....1441
SBUS<sb>:OSPI:FILTer:INVert....1441
SBUS<sb>:OSPI:FILTer:RST....1442
SBUS<sb>:OSPI:FILTer:BIT....1442
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:BIT....1442
SBUS<sb>:OSPI:FILTer:DMAX....1442
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1442
SBUS<sb>:OSPI:FILTer:DMIN....1443

SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1443
SBUS<sb>:OSPI:FILTer:DOPerator....1443
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1443
SBUS<sb>:OSPI:FILTer:IMAX....1443
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1443
SBUS<sb>:OSPI:FILTer:IMIN....1444
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1444
SBUS<sb>:OSPI:FILTer:IOPerator....1444
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1444
SBUS<sb>:OSPI:FILTer:ERENable....1445
SBUS<sb>:OSPI:FILTer:ERRor<n>:ENABle....1445
SBUS<sb>:OSPI:FILTer:FIENable....1445
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1445
SBUS<sb>:OSPI:FILTer:FRENable....1445
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:ENABle....1445

### SBUS<sb>:OSPI:FILTer:CHKall

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 555

### SBUS<sb>:OSPI:FILTer:CLR

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 555

### SBUS<sb>:OSPI:FILTer:INVert

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 555

**SBUS<sb>:OSPI:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 555

**SBUS<sb>:OSPI:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:OSPI:FILTer:BIT?** <Bit>
**SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 556

**SBUS<sb>:OSPI:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:OSPI:FILTer:DMAX?** <Data>
**SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 556

**SBUS<sb>:OSPI:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:OSPI:FILTer:DMIN?** <Data>
**SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 556

**SBUS<sb>:OSPI:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:OSPI:FILTer:DOPerator?** <Operator>
**SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 556

**SBUS<sb>:OSPI:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:OSPI:FILTer:IMAX?** <Data>
**SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 556

---

**SBUS<sb>:OSPI:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:OSPI:FILTer:IMIN?** <Data>
**SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 556

---

**SBUS<sb>:OSPI:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:OSPI:FILTer:IOPerator?** <Operator>
**SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 556

**SBUS<sb>:OSPI:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:OSPI:FILTer:ERENable?** <Enabler>
**SBUS<sb>:OSPI:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 557

**SBUS<sb>:OSPI:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:OSPI:FILTer:FIENable?** <Enabler>
**SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 556

**SBUS<sb>:OSPI:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:OSPI:FILTer:FRENable?** <Enabler>
**SBUS<sb>:OSPI:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

### 18.18.4.3 Format

SBUS<sb>:OSPI:OPCode:APPend ... 1446
SBUS<sb>:OSPI:OPCode:DALL ... 1446
SBUS<sb>:OSPI:OPCode:DELete ... 1446
SBUS<sb>:OSPI:OPCode:RESet ... 1447
SBUS<sb>:OSPI:LDOPcode ... 1447
SBUS<sb>:OSPI:SVOP ... 1447
SBUS<sb>:OSPI:OPCode:SIZE? ... 1447
SBUS<sb>:OSPI:OPCode:ITEM<n>:CODE ... 1448
SBUS<sb>:OSPI:OPCode:ITEM<n>:NAME ... 1448
SBUS<sb>:OSPI:OPCode:ITEM<n>:CMMode ... 1448
SBUS<sb>:OSPI:OPCode:ITEM<n>:CMEXtension ... 1449
SBUS<sb>:OSPI:OPCode:ITEM<n>:ADBYtes ... 1449
SBUS<sb>:OSPI:OPCode:ITEM<n>:ADMode ... 1449
SBUS<sb>:OSPI:OPCode:ITEM<n>:LACYcles ... 1450
SBUS<sb>:OSPI:OPCode:ITEM<n>:DTTYpe ... 1450
SBUS<sb>:OSPI:OPCode:ITEM<n>:DTMode ... 1451

**SBUS<sb>:OSPI:OPCode:APPend**

Adds a new item to the opcode list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add" on page 557

**SBUS<sb>:OSPI:OPCode:DALL**

Deletes all currently defined opcodes from the list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Delete all" on page 558

**SBUS<sb>:OSPI:OPCode:DELete** <Index>

Deletes the opcode with the selected index.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<Index>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Delete" on page 558

---

**SBUS<sb>:OSPI:OPCode:RESet**

Resets the opcode fields to the predefined values.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Preset" on page 558

---

**SBUS<sb>:OSPI:LDOPcode** <FileName>

Loads an opcode file from the selected file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open, Save as" on page 558

---

**SBUS<sb>:OSPI:SVOP** <FileName>

Saves the opcode file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open, Save as" on page 558

---

**SBUS<sb>:OSPI:OPCode:SIZE?**

Sets the size of the opcode, hence the number of opcodes currently defined on the table in the "Format" tab of "OCTO-SPI".

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:OPCode:ITEM<n>:CODE** <Opcode>

Sets the opcode value.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<Opcode> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Command" on page 559

---

**SBUS<sb>:OSPI:OPCode:ITEM<n>:NAME** <Name>

Sets the name for the Opcode.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Name" on page 559

---

**SBUS<sb>:OSPI:OPCode:ITEM<n>:CMMode** <CmdExtensionMd>

Enable to send an extra extension byte immediately after the command opcode.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<CmdExtensionMd> WR1S | WR2S | WR4S | WR8S | WR1D | WR2D | WR4D | WR8D

**WR1S | WR2S | WR4S | WR8S**
Number of lanes in single data rate mode

**WR1D | WR2D | WR4D | WR8D**
Number of lanes in dual data rate mode
*RST: WR8D

**Usage:** Asynchronous command

**Manual operation:** See "Command extension" on page 559

---

**SBUS<sb>:OSPI:OPCode:ITEM<n>:CMEXtension** <CmdExtension>

Selects how many lines are used for sending the command extension byte.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<CmdExtension> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Cmd. ext.mode" on page 559

---

**SBUS<sb>:OSPI:OPCode:ITEM<n>:ADBYtes** <AddressBytes>

Sets the address byte. It specifies the location in the flash memory where the operation (e.g., read, write) is performed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<AddressBytes> Range: 0 to 5
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Address Bytes" on page 560

---

**SBUS<sb>:OSPI:OPCode:ITEM<n>:ADMode** <AddressMode>

Selects how many lines are used to send the address bytes to the flash memory.

**Suffix:**
<sb> 1..4

<n> *

**Parameters:**

<AddressMode> WR1S | WR2S | WR4S | WR8S | WR1D | WR2D | WR4D | WR8D

**WR1S | WR2S | WR4S | WR8S**
Number of lanes in single data rate mode

**WR1D | WR2D | WR4D | WR8D**
Number of lanes in dual data rate mode

*RST: WR8D

**Usage:** Asynchronous command

**Manual operation:** See "Address mode" on page 560

---

**SBUS<sb>:OSPI:OPCode:ITEM<n>:LACYcles** <LatencyCycles>

Sets the number of latancy cycles. Latency cycles are clock cycles inserted after the address or other command sequences but before data transfer begins.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**

<LatencyCycles> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Latency Cycles" on page 560

---

**SBUS<sb>:OSPI:OPCode:ITEM<n>:DTTYpe** <DataType>

Determines the direction of the data flow for the payload after the command and address are sent.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**

<DataType> NONE | WRITe | READ

*RST: NONE

**Usage:** Asynchronous command

**Manual operation:** See "Data type" on page 560

**SBUS<sb>:OSPI:OPCode:ITEM<n>:DTMode** <DataMode>

Selects how many lines are used for the data transfer.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the opcode

**Parameters:**
<DataMode> WR1S | WR2S | WR4S | WR8S | WR1D | WR2D | WR4D | WR8D

**WR1S | WR2S | WR4S | WR8S**
Number of lanes in single data rate mode

**WR1D | WR2D | WR4D | WR8D**
Number of lanes in dual data rate mode

*RST: WR8D

**Usage:** Asynchronous command

**Manual operation:** See "Data mode" on page 560

### 18.18.4.4 Software trigger

There are two commands for each parameter, that you can use for defining the OCTO SPI settings.

For example, to set the *Frame type =Data > Field =Addr >01100* value you can use one of the following commands:

- `TRIGger:SBSW:OSPI:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:SBSW:OSPI:DMIN "Data", "Addr", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:OSPI:CHKall....1452
TRIGger:SBSW:OSPI:CLR....1452
TRIGger:SBSW:OSPI:INVert....1452
TRIGger:SBSW:OSPI:RST....1452
TRIGger:SBSW:OSPI:FRENable....1453
TRIGger:SBSW:OSPI:FRAMe<fr>:ENABle....1453
TRIGger:SBSW:OSPI:BIT....1453
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:BIT....1453
TRIGger:SBSW:OSPI:DMAX....1453
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:DMAX....1453
TRIGger:SBSW:OSPI:DMIN....1454
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:DMIN....1454
TRIGger:SBSW:OSPI:DOPerator....1454
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:DOPerator....1454
TRIGger:SBSW:OSPI:FIENable....1454
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:ENABle....1454

TRIGger:SBSW:OSPI:IMAX ..... 1455
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:IMAX ..... 1455
TRIGger:SBSW:OSPI:IMIN ..... 1455
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:IMIN ..... 1455
TRIGger:SBSW:OSPI:IOPerator ..... 1456
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:IOPerator ..... 1456
TRIGger:SBSW:OSPI:ERENable ..... 1456
TRIGger:SBSW:OSPI:ERRor<m>:ENABle ..... 1456
SBUS<sb>:OSPI:SWTindex? ..... 1456
SBUS<sb>:OSPI:SWTTime? ..... 1457

### TRIGger:SBSW:OSPI:CHKall

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 561

### TRIGger:SBSW:OSPI:CLR

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 561

### TRIGger:SBSW:OSPI:INVert

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 561

### TRIGger:SBSW:OSPI:RST

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 561

**TRIGger:SBSW:OSPI:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:OSPI:FRENable?** <Enabler>
**TRIGger:SBSW:OSPI:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 562

**TRIGger:SBSW:OSPI:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:OSPI:BIT?** <Bit>
**TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 562

**TRIGger:SBSW:OSPI:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:OSPI:DMAX?** <Data>
**TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 562

---

**TRIGger:SBSW:OSPI:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:OSPI:DMIN?** <Data>
**TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 562

---

**TRIGger:SBSW:OSPI:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:OSPI:DOPerator?** <Operator>
**TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 562

---

**TRIGger:SBSW:OSPI:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:OSPI:FIENable?** <Enabler>
**TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 562

---

**TRIGger:SBSW:OSPI:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:OSPI:IMAX?** <Data>
**TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 562

---

**TRIGger:SBSW:OSPI:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:OSPI:IMIN?** <Data>
**TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 562

**TRIGger:SBSW:OSPI:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:OSPI:IOPerator?** <Operator>
**TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 562

**TRIGger:SBSW:OSPI:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:OSPI:ERENable?** <Enabler>
**TRIGger:SBSW:OSPI:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 563

**SBUS<sb>:OSPI:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:OSPI:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.4.5 Decode results

SBUS<sb>:OSPI:FCOunt? 1457
SBUS<sb>:OSPI:FRAMe<fr>:ADDR? 1458
SBUS<sb>:OSPI:FRAMe<fr>:ADFormatted? 1458
SBUS<sb>:OSPI:FRAMe<fr>:BITRate? 1458
SBUS<sb>:OSPI:FRAMe<fr>:DATA? 1458
SBUS<sb>:OSPI:FRAMe<fr>:FLDCount? 1459
SBUS<sb>:OSPI:FRAMe<fr>:FDATa? 1459
SBUS<sb>:OSPI:FRAMe<fr>:FLD<fl>:VALue? 1459
SBUS<sb>:OSPI:FRAMe<fr>:FLD<fl>:FVALue? 1460
SBUS<sb>:OSPI:FRAMe<fr>:FLD<fl>:LABel? 1460
SBUS<sb>:OSPI:FRAMe<fr>:FLD<fl>:STATe? 1460
SBUS<sb>:OSPI:FRAMe<fr>:CMD? 1461
SBUS<sb>:OSPI:FRAMe<fr>:CMDFormatted? 1461
SBUS<sb>:OSPI:FRAMe<fr>:CEXTension? 1461
SBUS<sb>:OSPI:FRAMe<fr>:CXFormatted? 1462
SBUS<sb>:OSPI:FRAMe<fr>:DCOunt? 1462
SBUS<sb>:OSPI:FRAMe<fr>:TYPE? 1462
SBUS<sb>:OSPI:FRAMe<fr>:STARt? 1463
SBUS<sb>:OSPI:FRAMe<fr>:STATus? 1463
SBUS<sb>:OSPI:FRAMe<fr>:STOP? 1463

**SBUS<sb>:OSPI:FCOunt?**

Returns the number of decoded frames.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:OSPI:FRAMe<fr>:ADDR?**

Returns the address of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameAddress> Range: 0 to 281474976710655
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:OSPI:FRAMe<fr>:ADFormatted?**

Returns the formatted address of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrmFormattedAddr>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:OSPI:FRAMe<fr>:BITRate?**

Returns the bit rate of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<PrimaryBitRate> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

**SBUS<sb>:OSPI:FRAMe<fr>:DATA?**

Returns the data of the specified frame in comma-separated values.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

<Values>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:FLDCount?**

Returns the field count for the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:FDATa?**

Returns the foramtted data value of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameData>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:FLD<fl>:VALue?**

Returns the value of the specified field of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**

<Value> Range: 0 to 18446744073709551615
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**

<FormattedValue>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:FLD<fl>:LABel?**

Returns the label of the selected field in the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**

<Label>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:FLD<fl>:STATe?**

Returns the state of the selected field in the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**

<State> OK | LENGth | OPCode | INComplete

**OK**
The field is valid.

**LENGth**
The field does not have the expected length.

**OPCode**
The opcode value is not found in your frame description list. See Section 14.4.4, "OCTO-SPI format configuration", on page 557.

**INComplete**
The field is not completely contained in the acquisition. The acquired part of the field is usually valid, but cannot always be trusted.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:CMD?**

Returns the opcode of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameCommand> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:CMDFormatted?**

Returns the formatted opcode of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrmFormattedCmd>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:CEXTension?**

Returns the extension byte of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrmCmdExt> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:CXFormatted?**

Returns the formatted extension byte of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrmFormattedCmdExt>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:DCOunt?**

Returns the number of data payload frames.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameDataCount> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:TYPE?**

Returns the type of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameType>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:STARt?**

Returns the start time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:STATus?**

Returns the overall state of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameState> OK | LENGth | OPCode | INComplete

`OK`: the frame is valid.
`OPCode`: operation code.
`INComplete`: INComplete word. The word does not have the expected word length.
`LENGth`: The frame is not contained in the acquisition.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:OSPI:FRAMe<fr>:STOP?**

Returns the end time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

## 18.18.5 I²C (option R&S MXO3-K510)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.


- Configuration....1464
- Filter....1467
- Format....1473
- Hardware trigger....1482
- Software trigger....1486
- Decode results....1491


### 18.18.5.1 Configuration


SBUS<sb>:I2C:SCL:HYSTeresis....1464
SBUS<sb>:I2C:SCL:SOURce....1465
SBUS<sb>:I2C:SCL:THReshold....1465
SBUS<sb>:I2C:SDA:HYSTeresis....1465
SBUS<sb>:I2C:SDA:SOURce....1465
SBUS<sb>:I2C:SDA:THReshold....1466
SBUS<sb>:I2C:POSition....1466
SBUS<sb>:I2C:SCALe....1466
SBUS<sb>:I2C:SYMBols....1466
SBUS<sb>:I2C:NEWList....1467


---

**SBUS<sb>:I2C:SCL:HYSTeresis** <Hysteresis>

Sets a hysteresis value for the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "SDA/SCL threshold, SDA/SCL hysteresis" on page 570

**SBUS<sb>:I2C:SCL:SOURce** SCLSource

Selects the waveform source of the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
SCLSource C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "SCL source" on page 569

**SBUS<sb>:I2C:SCL:THReshold** <Threshold>

Sets a user-defined threshold value for the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "SDA/SCL threshold, SDA/SCL hysteresis" on page 570

**SBUS<sb>:I2C:SDA:HYSTeresis** <Hysteresis>

Sets a hysteresis value for the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "SDA/SCL threshold, SDA/SCL hysteresis" on page 570

**SBUS<sb>:I2C:SDA:SOURce** SDASource

Sets the source channel to which the data line is connected.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
SDASource C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "SDA source" on page 569

---

**SBUS<sb>:I2C:SDA:THReshold** <Threshold>

Sets a user-defined threshold value for the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "SDA/SCL threshold, SDA/SCL hysteresis" on page 570

---

**SBUS<sb>:I2C:POSition** <Position>

Sets the vertical position of the $I^2C$ signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Position>

**Usage:** Asynchronous command

---

**SBUS<sb>:I2C:SCALe** <Scale>

Set the vertical scale of the $I^2C$ signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Scale>

**Usage:** Asynchronous command

---

**SBUS<sb>:I2C:SYMBols** <ShowSymbols>

Activates the symbol list to be used for decoding.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ShowSymbols> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show symbols" on page 571

---

**SBUS<sb>:I2C:NEWList** <FileName>

Loads a symbol list file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName> String parameter with path and file name.

**Example:**
```
SBUS1:I2C:NEWList
'/home/storage/userData/Protocol/I2C.csv'
SBUS1:I2C:SYMBols ON
```

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Show symbols" on page 571

### 18.18.5.2 Filter

There are two commands for each parameter, that you can use for defining the I2C settings.

For example, to set the *Frame type =WRITe > Field =Address >Data* value you can use one of the following commands:

- `SBUS:I2C:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:I2C:FILTer:DMIN "WRITe", "Address", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:I2C:FILTer:CHKall ... 1468
SBUS<sb>:I2C:FILTer:CLR ... 1468
SBUS<sb>:I2C:FILTer:INVert ... 1468
SBUS<sb>:I2C:FILTer:RST ... 1468
SBUS<sb>:I2C:FILTer:FRENable ... 1469
SBUS<sb>:I2C:FILTer:FRAMe<fr>:ENABle ... 1469
SBUS<sb>:I2C:FILTer:DMAX ... 1469
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1469
SBUS<sb>:I2C:FILTer:DMIN ... 1469
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1469
SBUS<sb>:I2C:FILTer:DOPerator ... 1470
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1470

SBUS<sb>:I2C:FILTer:ERENable ... 1470
SBUS<sb>:I2C:FILTer:ERRor<n>:ENABle ... 1470
SBUS<sb>:I2C:FILTer:IMAX ... 1471
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1471
SBUS<sb>:I2C:FILTer:IMIN ... 1471
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1471
SBUS<sb>:I2C:FILTer:IOPerator ... 1471
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1471
SBUS<sb>:I2C:FILTer:BIT ... 1472
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1472
SBUS<sb>:I2C:FILTer:FIENable ... 1472
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1472

### SBUS<sb>:I2C:FILTer:CHKall

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 573

### SBUS<sb>:I2C:FILTer:CLR

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 573

### SBUS<sb>:I2C:FILTer:INVert

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 573

### SBUS<sb>:I2C:FILTer:RST

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 573

---

**SBUS<sb>:I2C:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:I2C:FILTer:FRENable?** <Enabler>
**SBUS<sb>:I2C:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables or disables the specific frame to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 573

---

**SBUS<sb>:I2C:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:I2C:FILTer:DMAX?** <Data>
**SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:I2C:FILTer:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 574

---

**SBUS<sb>:I2C:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:I2C:FILTer:DMIN?** <Data>
**SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 574

---

**SBUS<sb>:I2C:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:I2C:FILTer:DOPerator?** <Operator>
**SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 574

---

**SBUS<sb>:I2C:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:I2C:FILTer:ERENable?** <Enabler>
**SBUS<sb>:I2C:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> index of the error

**Parameters:**
<Enable> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 575

**SBUS<sb>:I2C:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:I2C:FILTer:IMAX?** <Data>
**SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

You can set the operator with `SBUS<sb>:I2C:FILTer:IOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 574

**SBUS<sb>:I2C:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:I2C:FILTer:IMIN?** <Data>
**SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 574

**SBUS<sb>:I2C:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:I2C:FILTer:IOPerator?** <Operator>
**SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 574

---

**SBUS<sb>:I2C:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:I2C:FILTer:BIT?** <Bit>
**SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 574

---

**SBUS<sb>:I2C:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:I2C:FILTer:FIENable?** <Enabler>
**SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 574

### 18.18.5.3 Format

This section describes the commands to specify and query frame formats for the "I2C" protocol.

SBUS<sb>:I2C:FORMat:SELect....1473
SBUS<sb>:I2C:FORMat:ADDFrame....1473
SBUS<sb>:I2C:FORMat:LOAD....1474
SBUS<sb>:I2C:FORMat:SAVE....1474
SBUS<sb>:I2C:FORMat:CLR....1474
SBUS<sb>:I2C:FORMat:FCOunt?....1474
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ADDField....1475
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLDCount?....1475
SBUS<sb>:I2C:FORMat:FRAMe<fr>:NAME....1475
SBUS<sb>:I2C:FORMat:FRAMe<fr>:COLor....1476
SBUS<sb>:I2C:FORMat:FRAMe<fr>:TYPE....1476
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID1Field....1476
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID2Field....1476
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID3Field....1476
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID1Value....1477
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID2Value....1477
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID3Value....1477
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:NAME....1477
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:COLor....1477
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1478
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1478
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1478
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1479
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1479
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1480
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1480
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1481
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1481
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:PARA....1481
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:PARB....1481

---

**SBUS<sb>:I2C:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 575

---

**SBUS<sb>:I2C:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 575

---

**SBUS<sb>:I2C:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 576

---

**SBUS<sb>:I2C:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 576

---

**SBUS<sb>:I2C:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 576

---

**SBUS<sb>:I2C:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 576

---

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 577

---

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 577

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 577

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 577

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 577

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 577

---

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 578

---

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 578

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<FirstBit> Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 578

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 578

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 579

---

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 579

---

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 579

---

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 580

---

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 580

---

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 580

---

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 580

---

**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)

- Parameter B: Offset

Available if `SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<ParameterB> Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 580

### 18.18.5.4 Hardware trigger

Trigger commands for serial buses affect the current trigger source. Therefore, set the trigger source to the required serial bus with `TRIGger:EVENt<ev>:SOURce` before sending the bus-specific trigger commands.

TRIGger:SBHW:I2C:TYPE....1482
TRIGger:SBHW:I2C:ACCess....1483
TRIGger:SBHW:I2C:ACONdition....1483
TRIGger:SBHW:I2C:ADDRess....1483
TRIGger:SBHW:I2C:ADDTo....1484
TRIGger:SBHW:I2C:ADNack....1484
TRIGger:SBHW:I2C:AMODe....1484
TRIGger:SBHW:I2C:DCONdition....1484
TRIGger:SBHW:I2C:DMIN....1485
TRIGger:SBHW:I2C:DPOSition....1485
TRIGger:SBHW:I2C:DRNack....1485
TRIGger:SBHW:I2C:DWNack....1485

---

**TRIGger:SBHW:I2C:TYPE** <Type>

Selects the trigger type for I²C analysis.

**Parameters:**
<Type> STARt | REPStart | STOP | NACK | ADDRess | DATA | ADAT

**STARt**
Start condition

**REPStart**
Repeated start - the start condition occurs without previous stop condition.

**STOP**
Stop condition, end of frame

**NACK**
Missing acknowledge bit. To localize specific missing acknowledge bits, use:
`TRIGger:SBHW:I2C:ADNack`
`TRIGger:SBHW:I2C:DWNack`
`TRIGger:SBHW:I2C:DRNack`

**ADDRess**
Triggers on one specific address

**DATA**
Triggers on a specific data

**ADAT**
Triggers on a combination of address and data condition.

*RST: STARt

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 581

---

**TRIGger:SBHW:I2C:ACCess** <RWBitAddress>

Toggles the trigger condition between read and write access of the primary. Select "Either" if the transfer direction is not relevant for the trigger condition.

**Parameters:**
<RWBitAddress> READ | WRITe | EITHer
*RST: EITHer

**Usage:** Asynchronous command

**Manual operation:** See "R/W bit" on page 583

---

**TRIGger:SBHW:I2C:ACONdition** <AddrOptor>

Sets the operator to set a specific address or an address range. The address values are set with `TRIGger:SBHW:I2C:ADDRess` and `TRIGger:SBHW:I2C:ADDTo`.

**Parameters:**
<AddrOptor> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange
*RST: EQUal

**Usage:** Asynchronous command

---

**TRIGger:SBHW:I2C:ADDRess** <Address>

Triggers on the specified address, or sets the start value of an address range depending on the condition set with `TRIGger:SBHW:I2C:ACONdition`.

**Parameters:**
<Address>

**Usage:** Asynchronous command

**Manual operation:** See "Address" on page 583

---

**TRIGger:SBHW:I2C:ADDTo** <AddressTo>

Sets the end value of an address range if the condition is set to an address range with `TRIGger:SBHW:I2C:ACONdition`.

**Parameters:**
<AddressTo>

**Usage:** Asynchronous command

**Manual operation:** See "Address" on page 583

---

**TRIGger:SBHW:I2C:ADNack** <AddressNack>

Triggers if the address acknowledge bit is missing - no target recognizes the address.

**Parameters:**
<AddressNack> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "No Ack conditions" on page 583

---

**TRIGger:SBHW:I2C:AMODe** <AddressType>

Sets the address length to be triggered on: 7 bit or 10 bit.

**Parameters:**
<AddressType> BIT7 | BIT7RW | BIT7_RW | BIT10
Note that `BIT7RW` is the same address type as `BIT7_RW`.
**BIT7 | BIT10**
Enter only the seven or ten address bits in the address pattern.
**BIT7RW | BIT7_RW**
Enter seven address bits and also the read/write bit.
*RST: BIT7

**Usage:** Asynchronous command

**Manual operation:** See "Address type" on page 583

---

**TRIGger:SBHW:I2C:DCONdition** <DataOperator>

Sets the operator to set a specific data value or a data range.

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan
*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 584

---

**TRIGger:SBHW:I2C:DMIN** <Data>

Specifies the data bit pattern, or sets the start value of a data pattern range. Enter the bytes in MSB first bit order. The maximum pattern length is 64 bit. Waveform data is compared with the pattern byte-by-byte.

**Parameters:**
<Data>

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 584

---

**TRIGger:SBHW:I2C:DPOSition** <DataPosition>

Sets the number of data bytes to be skipped after the address.

**Parameters:**
<DataPosition> Range: 1 to 4096
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Position" on page 584

---

**TRIGger:SBHW:I2C:DRNack** <DataReadNack>

Triggers on the end of the read process when the controller reads data from the target. This NACK is sent according to the protocol definition, it is not an error.

**Parameters:**
<DataReadNack> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "No Ack conditions" on page 583

---

**TRIGger:SBHW:I2C:DWNack** <DataWriteNack>

Triggers if a date acknowledge bit is missing - the addressed target does not accept the data.

**Parameters:**
<DataWriteNack> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "No Ack conditions" on page 583

### 18.18.5.5 Software trigger

There are two commands for each parameter, that you can use for defining the I2C settings.

For example, to set the *Frame type =WRITe > Field =Address >Data* value you can use one of the following commands:

- `TRIGger:SBSW:I2C:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `STRIGger:SBSW:I2C:DMIN "WRITe", "Address", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:I2C:CHKall....1486
TRIGger:SBSW:I2C:CLR....1487
TRIGger:SBSW:I2C:INVert....1487
TRIGger:SBSW:I2C:RST....1487
TRIGger:SBSW:I2C:FRENable....1487
TRIGger:SBSW:I2C:FRAMe<fr>:ENABle....1487
TRIGger:SBSW:I2C:BIT....1487
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:BIT....1487
TRIGger:SBSW:I2C:DMAX....1488
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DMAX....1488
TRIGger:SBSW:I2C:DMIN....1488
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DMIN....1488
TRIGger:SBSW:I2C:DOPerator....1489
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DOPerator....1489
TRIGger:SBSW:I2C:FIENable....1489
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:ENABle....1489
TRIGger:SBSW:I2C:IMAX....1489
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IMAX....1489
TRIGger:SBSW:I2C:IMIN....1490
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IMIN....1490
TRIGger:SBSW:I2C:IOPerator....1490
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IOPerator....1490
TRIGger:SBSW:I2C:ERENable....1490
TRIGger:SBSW:I2C:ERRor<m>:ENABle....1490
SBUS<sb>:I2C:SWTindex?....1491
SBUS<sb>:I2C:SWTTime?....1491

---

**TRIGger:SBSW:I2C:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 585

**TRIGger:SBSW:I2C:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 585

**TRIGger:SBSW:I2C:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 585

**TRIGger:SBSW:I2C:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 585

**TRIGger:SBSW:I2C:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:I2C:FRENable?** <Enabler>
**TRIGger:SBSW:I2C:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 586

**TRIGger:SBSW:I2C:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:I2C:BIT?** <Bit>
**TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 586

---

**TRIGger:SBSW:I2C:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:I2C:DMAX?** <Data>
**TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 586

---

**TRIGger:SBSW:I2C:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:I2C:DMIN?** <Data>
**TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 586

**TRIGger:SBSW:I2C:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:I2C:DOPerator?** <Operator>
**TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 586

**TRIGger:SBSW:I2C:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:I2C:FIENable?** <Enabler>
**TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 586

**TRIGger:SBSW:I2C:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:I2C:IMAX?** <Data>
**TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 586

---

**TRIGger:SBSW:I2C:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:I2C:IMIN?** <Data>
**TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 586

---

**TRIGger:SBSW:I2C:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:I2C:IOPerator?** <Operator>
**TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 586

---

**TRIGger:SBSW:I2C:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:I2C:ERENable?** <Enabler>
**TRIGger:SBSW:I2C:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 587

---

**SBUS<sb>:I2C:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.5.6 Decode results

SBUS<sb>:I2C:FCOunt? 1492
SBUS<sb>:I2C:FRAMe<fr>:AACCess? 1492
SBUS<sb>:I2C:FRAMe<fr>:ACCess? 1492
SBUS<sb>:I2C:FRAMe<fr>:ACOMplete? 1493
SBUS<sb>:I2C:FRAMe<fr>:ADBStart? 1493
SBUS<sb>:I2C:FRAMe<fr>:ADDRess? 1493

SBUS<sb>:I2C:FRAMe<fr>:ADEVice? ..... 1494
SBUS<sb>:I2C:FRAMe<fr>:AMODe? ..... 1494
SBUS<sb>:I2C:FRAMe<fr>:ASTart? ..... 1494
SBUS<sb>:I2C:FRAMe<fr>:BITRate? ..... 1495
SBUS<sb>:I2C:FRAMe<fr>:FLDCount? ..... 1495
SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:FVALue? ..... 1495
SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:LABel? ..... 1496
SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:STARt? ..... 1496
SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:STATe? ..... 1496
SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:VALue? ..... 1497
SBUS<sb>:I2C:FRAMe<fr>:DATA? ..... 1497
SBUS<sb>:I2C:FRAMe<fr>:RWBStart? ..... 1498
SBUS<sb>:I2C:FRAMe<fr>:STARt? ..... 1498
SBUS<sb>:I2C:FRAMe<fr>:STATus? ..... 1498
SBUS<sb>:I2C:FRAMe<fr>:STOP? ..... 1499
SBUS<sb>:I2C:FRAMe<fr>:SYMBol? ..... 1499

### SBUS<sb>:I2C:FCOunt?

Returns the number of decoded frames.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count> Total number of decoded frames.

**Usage:** Query only
Asynchronous command

### SBUS<sb>:I2C:FRAMe<fr>:AACCess?

Returns the address acknowledge bit value for the indicated frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<AddressAckBit> ACK | NACK | EITHer
*RST: EITHer

**Usage:** Query only
Asynchronous command

### SBUS<sb>:I2C:FRAMe<fr>:ACCess?

Returns the value of the R/W bit of the indicated frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<RWBit> UNDefined | READ | WRITe | EITHer
*RST: UNDefined

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:ACOMplete?**

Returns if the address is completely contained in the acquisition.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<AddrComplete> OFF | ON
*RST: OFF

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:ADBStart?**

Returns the start time of the address acknowledge bit.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<AddrAckBtStrt> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:ADDRess?**

Returns the device address value of the indicated frame. That is, the address value that is shown in the decoded cells and in the decode results table.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<AddressValue> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:ADEVice?**

Returns the pure device address of the indicated frame *without* the R/W bit.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<DeviceAddress> Range: 0 to 1023
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:AMODe?**

Returns the address length.

See also `TRIGger:SBHW:I2C:AMODe` on page 1484.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<AddressType> BIT7 | BIT7RW | BIT7_RW | BIT10
*RST: BIT7

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:ASTart?**

Returns the start time of the address for the indicated frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<AddressStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:BITRate?**

Returns the primary bit rate.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<PrimaryBitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:FLDCount?**

Returns the number of fields in the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**

<FormattedValue>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:LABel?**

Returns the name (label) of the specified field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:STARt?**

Returns the start time of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Start> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:STATe?**

Returns the overall state of the specified field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | ADDifferent | NOSTop | UNKNown | INComplete

**OK**
The field has no error, it is valid.

**ADDifferent**
10-bit read address different. For a read access on a 10-bit address, the first address byte is sent twice, first as write, the second as read. The first 7 bits of the byte must be identical. If they are not identical, the ADDiffernt error is indicated.

**NOSTop**
The stop bit is missing.

**UNKNown**
Unknown field type detected.

**INComplete**
The field is not completely contained in the acquisition. The acquired part of the field is usually valid, but cannot always be trusted.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:VALue?**

Returns the data value of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:DATA?**

Returns the data words of the specified frame as comma-separated values.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

<Values>

**Example:**
```
BUS:I2C:FRAMe4:DATA?
<-- 3,74,164,18
```

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:RWBStart?**

Returns the start time of the R/W bit

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<RWBitStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:STARt?**

Returns the start time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:STATus?**

Returns the overall state of the frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameState> OK | ADDifferent | NOSTop | UNKNown | INComplete

**OK**
The frame is valid.

**ADDifferent**
10-bit read address different. For a read access on a 10-bit address, the first address byte is sent twice, first as write, the second as read. The first 7 bits of the byte must be identical. If they are not identical, the ADDiffernt error is indicated.

**NOSTop**
The stop bit is missing.

**UNKNown**
Unknown frame type detected.

**INComplete**
The frame is not completely contained in the acquisition. The acquired part of the frame is usually valid, but cannot always be trusted.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:STOP?**

Returns the end time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I2C:FRAMe<fr>:SYMBol?**

Returns the symbolic label of the specified frame if the label list is enabled.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Translation> String with the symbolic name of the frame

**Usage:** Query only
Asynchronous command

## 18.18.6 I3C (option R&S MXO3-K550)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.


- Configuration....1500
- Filter....1504
- Format....1509
- Software trigger....1518
- Decode results....1524


### 18.18.6.1 Configuration


SBUS<sb>:I3C:SCL:HYSTeresis....1500
SBUS<sb>:I3C:SCL:SOURce....1500
SBUS<sb>:I3C:SCL:THReshold....1501
SBUS<sb>:I3C:SDA:HYSTeresis....1501
SBUS<sb>:I3C:SDA:SOURce....1501
SBUS<sb>:I3C:SDA:THReshold....1502
SBUS<sb>:I3C:MINGap:SELect....1502
SBUS<sb>:I3C:MINGap:WIDTh....1502
SBUS<sb>:I3C:POSition....1502
SBUS<sb>:I3C:SCALe....1503
SBUS<sb>:I3C:SYMBols....1503
SBUS<sb>:I3C:NEWList....1503


---

**SBUS<sb>:I3C:SCL:HYSTeresis** <SCLHysteresis>

Sets a hysteresis value for the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SCLHysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "SDA/SCL threshold, SDA/SCL hysteresis" on page 594

---

**SBUS<sb>:I3C:SCL:SOURce** SCLSource

Selects the waveform source of the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
SCLSource C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "SCL source" on page 593

---

**SBUS<sb>:I3C:SCL:THReshold** <SCLThreshold>

Sets a user-defined threshold value for the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SCLThreshold>

**Usage:** Asynchronous command

**Manual operation:** See "SDA/SCL threshold, SDA/SCL hysteresis" on page 594

---

**SBUS<sb>:I3C:SDA:HYSTeresis** <SDAHysteresis>

Sets a hysteresis value for the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SDAHysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "SDA/SCL threshold, SDA/SCL hysteresis" on page 594

---

**SBUS<sb>:I3C:SDA:SOURce** SDASource

Sets the source channel to which the data line is connected.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
SDASource C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "SDA source" on page 593

**SBUS<sb>:I3C:SDA:THReshold** <SDAThreshold>

Sets a user-defined threshold value for the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SDAThreshold>

**Usage:** Asynchronous command

**Manual operation:** See "SDA/SCL threshold, SDA/SCL hysteresis" on page 594

**SBUS<sb>:I3C:MINGap:SELect** <MingapSelect>

Specifies whether a minimum gap before the 1st decoded frame is required.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MingapSelect> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Require min gap" on page 594

**SBUS<sb>:I3C:MINGap:WIDTh** <MingapWidth>

Specifies the minimum duration of the gap.

Available, if `SBUS<sb>:I3C:MINGap:SELect` is set to `ON`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MingapWidth> Range: 1E-11 to 1
Increment: 1E-11
*RST: 1E-06
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Min gap width" on page 594

**SBUS<sb>:I3C:POSition** <Position>

Sets the vertical position of the I3C signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Position>

**Usage:** Asynchronous command

---

**SBUS<sb>:I3C:SCALe** <Scale>

Sets the vertical scale of the I3C signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Scale>

**Usage:** Asynchronous command

---

**SBUS<sb>:I3C:SYMBols** <ShowSymbols>

Activates the symbol list to be used for decoding.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ShowSymbols> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show symbols" on page 595

---

**SBUS<sb>:I3C:NEWList** <FileName>

Loads a symbol list file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName> String parameter with path and file name.

**Example:**
```
SBUS1:I3C:NEWList
'/home/storage/userData/Protocol/I3C.csv'
SBUS1:I3C:SYMBols ON
```

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Show symbols" on page 595

### 18.18.6.2 Filter

There are two commands for each parameter, that you can use for defining the I3C settings.

For example, to set the *Frame type =WRITe > Field =Address >Data* value you can use one of the following commands:

- `SBUS:I3C:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:I3C:FILTer:DMIN "WRITe", "Address", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:I3C:FILTer:CHKall ..... 1504
SBUS<sb>:I3C:FILTer:CLR ..... 1505
SBUS<sb>:I3C:FILTer:INVert ..... 1505
SBUS<sb>:I3C:FILTer:RST ..... 1505
SBUS<sb>:I3C:FILTer:FRENable ..... 1505
SBUS<sb>:I3C:FILTer:FRAMe<fr>:ENABle ..... 1505
SBUS<sb>:I3C:FILTer:DMAX ..... 1506
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DMAX ..... 1506
SBUS<sb>:I3C:FILTer:DMIN ..... 1506
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DMIN ..... 1506
SBUS<sb>:I3C:FILTer:DOPerator ..... 1506
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ..... 1506
SBUS<sb>:I3C:FILTer:ERENable ..... 1507
SBUS<sb>:I3C:FILTer:ERRor<n>:ENABle ..... 1507
SBUS<sb>:I3C:FILTer:IMAX ..... 1507
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:IMAX ..... 1507
SBUS<sb>:I3C:FILTer:IMIN ..... 1508
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:IMIN ..... 1508
SBUS<sb>:I3C:FILTer:IOPerator ..... 1508
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ..... 1508
SBUS<sb>:I3C:FILTer:BIT ..... 1508
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:BIT ..... 1508
SBUS<sb>:I3C:FILTer:FIENable ..... 1509
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:ENABle ..... 1509

---

**SBUS<sb>:I3C:FILTer:CHKall**

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 597

**SBUS<sb>:I3C:FILTer:CLR**

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 597

**SBUS<sb>:I3C:FILTer:INVert**

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 597

**SBUS<sb>:I3C:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 597

**SBUS<sb>:I3C:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:I3C:FILTer:FRENable?** <Enabler>
**SBUS<sb>:I3C:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables or disables the specific frame to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 597

**SBUS<sb>:I3C:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:I3C:FILTer:DMAX?** <Data>
**SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 597

**SBUS<sb>:I3C:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:I3C:FILTer:DMIN?** <Data>
**SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 597

**SBUS<sb>:I3C:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:I3C:FILTer:DOPerator?** <Operator>
**SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**

<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 597

---

**SBUS<sb>:I3C:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:I3C:FILTer:ERENable?** <Enabler>
**SBUS<sb>:I3C:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<n> Index of the error

**Parameters:**

<Enable> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 599

---

**SBUS<sb>:I3C:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:I3C:FILTer:IMAX?** <Data>
**SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

You can set the operator with `SBUS<sb>:I3C:FILTer:IOPerator`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 597

**SBUS<sb>:I3C:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:I3C:FILTer:IMIN?** <Data>
**SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 597

**SBUS<sb>:I3C:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:I3C:FILTer:IOPerator?** <Operator>
**SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 597

**SBUS<sb>:I3C:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:I3C:FILTer:BIT?** <Bit>
**SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 597

---

**SBUS<sb>:I3C:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:I3C:FILTer:FIENable?** <Enabler>
**SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 597

### 18.18.6.3 Format

This section describes the commands to specify and query frame formats for the "I3C" protocol.

SBUS<sb>:I3C:FORMat:SELect....1510
SBUS<sb>:I3C:FORMat:ADDFrame....1510
SBUS<sb>:I3C:FORMat:LOAD....1510
SBUS<sb>:I3C:FORMat:SAVE....1510
SBUS<sb>:I3C:FORMat:CLR....1511
SBUS<sb>:I3C:FORMat:FCOunt?....1511
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ADDField....1511
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLDCount?....1512
SBUS<sb>:I3C:FORMat:FRAMe<fr>:NAME....1512
SBUS<sb>:I3C:FORMat:FRAMe<fr>:COLor....1512
SBUS<sb>:I3C:FORMat:FRAMe<fr>:TYPE....1512
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID1Field....1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID2Field....1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID3Field....1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID1Value....1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID2Value....1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID3Value....1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:NAME....1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:COLor....1514

SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ..... 1514
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:BITCount ..... 1514
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:REPetition ..... 1515
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:BITorder ..... 1515
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FORMat ..... 1516
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ..... 1516
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ..... 1517
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ..... 1517
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:UNIT ..... 1517
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:PARA ..... 1518
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:PARB ..... 1518

**SBUS<sb>:I3C:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 599

**SBUS<sb>:I3C:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 599

**SBUS<sb>:I3C:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 600

**SBUS<sb>:I3C:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 600

---

**SBUS<sb>:I3C:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 600

---

**SBUS<sb>:I3C:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 600

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 601

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 601

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 601

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 601

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 601

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 601

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 602

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 602

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<FirstBit>

Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 602

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 602

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 603

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 603

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 603

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 604

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**
<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 604

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**
<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 604

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 604

---

**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

Available if `SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<ParameterB> Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 604

### 18.18.6.4 Software trigger

There are two commands for each parameter, that you can use for defining the I3C settings.

For example, to set the *Frame type =WRITe > Field =Address >Data* value you can use one of the following commands:

- `TRIGger:SBSW:I3C:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `STRIGger:SBSW:I3C:DMIN "WRITe", "Address", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:I3C:CHKall....1519
TRIGger:SBSW:I3C:CLR....1519
TRIGger:SBSW:I3C:INVert....1519

TRIGger:SBSW:I3C:RST ..... 1520
TRIGger:SBSW:I3C:FRENable ..... 1520
TRIGger:SBSW:I3C:FRAMe<fr>:ENABle ..... 1520
TRIGger:SBSW:I3C:BIT ..... 1520
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:BIT ..... 1520
TRIGger:SBSW:I3C:DMAX ..... 1520
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DMAX ..... 1520
TRIGger:SBSW:I3C:DMIN ..... 1521
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DMIN ..... 1521
TRIGger:SBSW:I3C:DOPerator ..... 1521
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DOPerator ..... 1521
TRIGger:SBSW:I3C:FIENable ..... 1522
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:ENABle ..... 1522
TRIGger:SBSW:I3C:IMAX ..... 1522
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IMAX ..... 1522
TRIGger:SBSW:I3C:IMIN ..... 1522
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IMIN ..... 1522
TRIGger:SBSW:I3C:IOPerator ..... 1523
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IOPerator ..... 1523
TRIGger:SBSW:I3C:ERENable ..... 1523
TRIGger:SBSW:I3C:ERRor<m>:ENABle ..... 1523
SBUS<sb>:I3C:SWTindex? ..... 1523
SBUS<sb>:I3C:SWTTime? ..... 1524

### TRIGger:SBSW:I3C:CHKall

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 605

### TRIGger:SBSW:I3C:CLR

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 605

### TRIGger:SBSW:I3C:INVert

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 605

**TRIGger:SBSW:I3C:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 605

---

**TRIGger:SBSW:I3C:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:I3C:FRENable?** <Enabler>
**TRIGger:SBSW:I3C:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 606

---

**TRIGger:SBSW:I3C:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:I3C:BIT?** <Bit>
**TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 606

---

**TRIGger:SBSW:I3C:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:I3C:DMAX?** <Data>
**TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max> List of comma-separated values

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 606

---

**TRIGger:SBSW:I3C:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:I3C:DMIN?** <Data>
**TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min> List of comma-separated values

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 606

---

**TRIGger:SBSW:I3C:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:I3C:DOPerator?** <Operator>
**TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 606

**TRIGger:SBSW:I3C:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:I3C:FIENable?** <Enabler>
**TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 606

**TRIGger:SBSW:I3C:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:I3C:IMAX?** <Data>
**TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 606

**TRIGger:SBSW:I3C:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:I3C:IMIN?** <Data>
**TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 606

---

**TRIGger:SBSW:I3C:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:I3C:IOPerator?** <Operator>
**TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 606

---

**TRIGger:SBSW:I3C:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:I3C:ERENable?** <Enabler>
**TRIGger:SBSW:I3C:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**

<m> Index of the error

**Parameters:**

<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 607

---

**SBUS<sb>:I3C:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Return values:**

<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Return values:**

<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.6.5 Decode results

SBUS<sb>:I3C:FCOunt? 1525
SBUS<sb>:I3C:FRAMe<fr>:ADDRess? 1525
SBUS<sb>:I3C:FRAMe<fr>:ASTart? 1525
SBUS<sb>:I3C:FRAMe<fr>:FLDCount? 1526
SBUS<sb>:I3C:FRAMe<fr>:BITRate? 1526
SBUS<sb>:I3C:FRAMe<fr>:DATA? 1526
SBUS<sb>:I3C:FRAMe<fr>:RWBStart? 1526
SBUS<sb>:I3C:FRAMe<fr>:STARt? 1527
SBUS<sb>:I3C:FRAMe<fr>:STATus? 1527
SBUS<sb>:I3C:FRAMe<fr>:STOP? 1528
SBUS<sb>:I3C:FRAMe<fr>:SYMBol? 1528
SBUS<sb>:I3C:FRAMe<fr>:ACK? 1528
SBUS<sb>:I3C:FRAMe<fr>:ACKStart? 1529
SBUS<sb>:I3C:FRAMe<fr>:FLD<fl>:FVALue? 1529
SBUS<sb>:I3C:FRAMe<fr>:FLD<fl>:LABel? 1529
SBUS<sb>:I3C:FRAMe<fr>:FLD<fl>:STATe? 1530
SBUS<sb>:I3C:FRAMe<fr>:FLD<fl>:VALue? 1530
SBUS<sb>:I3C:FRAMe<fr>:COMMand? 1531
SBUS<sb>:I3C:FRAMe<fr>:CRC? 1531
SBUS<sb>:I3C:FRAMe<fr>:FADDress? 1531
SBUS<sb>:I3C:FRAMe<fr>:FCOMmand? 1532

SBUS<sb>:I3C:FRAMe<fr>:FCRC?....1532
SBUS<sb>:I3C:FRAMe<fr>:RWBit?....1532
SBUS<sb>:I3C:FRAMe<fr>:TYPE?....1532

**SBUS<sb>:I3C:FCOunt?**

Returns the number of decoded frames.

**Suffix:**
<sb> 1 to 4, index of the serial bus
Selects the serial bus.

**Return values:**
<Count> Total number of decoded frames.

**Usage:** Query only
Asynchronous command

**SBUS<sb>:I3C:FRAMe<fr>:ADDRess?**

Returns the device address value of the indicated frame. That is, the address value that is shown in the decoded cells and in the decode results table.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<AddressValue> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:I3C:FRAMe<fr>:ASTart?**

Returns the start time of the address for the indicated frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<AddressStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

**SBUS<sb>:I3C:FRAMe<fr>:FLDCount?**

Returns the number of fields in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:I3C:FRAMe<fr>:BITRate?**

Returns the primary bit rate.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<BitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

**SBUS<sb>:I3C:FRAMe<fr>:DATA?**

Returns the data words of the specified frame in comma-separated values.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

<Values>

**Example:**
```
BUS:I3C:FRAMe4:DATA?
<-- 3,74,164,18
```

**Usage:** Query only
Asynchronous command

**SBUS<sb>:I3C:FRAMe<fr>:RWBStart?**

Returns the start time of the R/W bit

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<RWBitStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:STARt?**

Returns the start time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:STATus?**

Returns the overall state of the frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameState> OK | ACK | PAR | CRC | LENGth | UNKNown | INComplete

**OK**
The frame is valid.

**ACK**
Wrong acknowledgment (ACK) delimiter occurred.

**PAR**
Wrong parity detected.

**CRC**
Cyclic redundancy check failed.

**LENGth**
Wrong length of frame.

**UNKNown**
Unknown frame type detected.

**INComplete**
The frame is not completely contained in the acquisition. The acquired part of the frame is usually valid, but cannot always be trusted.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:STOP?**

Returns the end time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:SYMBol?**

Returns the symbolic label of the specified frame if the label list is enabled.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Translation> String with the symbolic name of the frame

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:ACK?**

Returns the value of the acknowledge bit for the selected frame. Because this ACK bit is transmitted right after the address field, it acknowledges receiving the address.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<AckBitValue> ACK | NACK | EITHer
*RST: EITHer

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:ACKStart?**

Returns the start time of the acknowledge bit for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<AckBitStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:FLD<fl>:LABel?**

Returns the name (label) of the specified field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:FLD<fl>:STATe?**

Returns the overall state of the specified field for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | ACK | PAR | CRC | LENGth | UNKNown | INComplete

**OK**
The field has no error, it is valid.

**ACK**
Wrong acknowledgment (ACK) delimiter occurred.

**PAR**
Wrong parity detected.

**CRC**
Cyclic redundancy check failed.

**LENGth**
Wrong length of the field.

**UNKNown**
Unknown field type detected.

**INComplete**
The field is not completely contained in the acquisition. The acquired part of the field is usually valid, but cannot always be trusted.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:FLD<fl>:VALue?**

Returns the data value of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:COMMand?**

Returns the selected frame's command response, which is an integer.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Response> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:CRC?**

Returns the cyclic redundancy check (CRC) checksum of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<CRCValue> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:I3C:FRAMe<fr>:FADDress?**

Returns the formatted address for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedAddrVal>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:I3C:FRAMe<fr>:FCOMmand?**

Returns the formatted command from the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedCmd>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:I3C:FRAMe<fr>:FCRC?**

Returns the formatted CRC value for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedCRCVal>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:I3C:FRAMe<fr>:RWBit?**

Returns the value of the R/W bit of the selected frame. The value "0" denotes "Write", the value "1" denotes "Read".

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<RWBitValue> UNDefined | READ | WRITe | EITHer
*RST: UNDefined

**Usage:** Query only
Asynchronous command

**SBUS<sb>:I3C:FRAMe<fr>:TYPE?**

Returns the frame type of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameType> UNKNown | PROB | READ | WRIT | BRDC | DRCT | HDDR | HTSX

*RST: UNKNown

**Usage:** Query only
Asynchronous command

## 18.18.7 UART / RS-232 (option R&S MXO3- K510)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.


- Configuration ..... 1533
- Filter ..... 1539
- Format ..... 1544
- Hardware trigger ..... 1553
- Software trigger ..... 1555
- Decode results ..... 1561


### 18.18.7.1 Configuration


SBUS<sb>:UART:BITRate ..... 1533
SBUS<sb>:UART:BORDer ..... 1534
SBUS<sb>:UART:PACKets ..... 1534
SBUS<sb>:UART:PARity ..... 1534
SBUS<sb>:UART:POLarity ..... 1535
SBUS<sb>:UART:RX:HYSTeresis ..... 1535
SBUS<sb>:UART:RX:SOURce ..... 1535
SBUS<sb>:UART:RX:THReshold ..... 1536
SBUS<sb>:UART:SBIT ..... 1536
SBUS<sb>:UART:SSIZe ..... 1536
SBUS<sb>:UART:TOUT ..... 1537
SBUS<sb>:UART:TX:HYSTeresis ..... 1537
SBUS<sb>:UART:TX:SOURce ..... 1537
SBUS<sb>:UART:TX:THReshold ..... 1538
SBUS<sb>:UART:RX:POSition ..... 1538
SBUS<sb>:UART:RX:SCALe ..... 1538
SBUS<sb>:UART:TX:POSition ..... 1538
SBUS<sb>:UART:TX:SCALe ..... 1538


---

**SBUS<sb>:UART:BITRate** <Bitrate>

Sets the number of transmitted bits per second.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<Bitrate>
Range: 300 to 20000000
Increment: 1
*RST: 9600
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "Bit rate" on page 613

---

**SBUS<sb>:UART:BORDer** <BitOrder>

Selects the bit order, which determines if the data of the messages starts with MSB (most significant bit) or LSB (least significant bit).

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<BitOrder> LSBF | MSBF

*RST: LSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 613

---

**SBUS<sb>:UART:PACKets** <FrmSeparation>

Defines the method of packet separation. A packet is a number of subsequent words in a date stream.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<FrmSeparation> NONE | TOUT

**NONE**
Packets are not considered.

**TOUT**
Defines a timeout between the packets. To set the timeout, use `SBUS<sb>:UART:TOUT`.

*RST: TOUT

**Usage:** Asynchronous command

**Manual operation:** See "Packets" on page 613

---

**SBUS<sb>:UART:PARity** <Parity>

Defines the optional parity bit that is used for error detection.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Parity> NONE | ODD | EVEN | MARK | SPC | DC

**MARK**
The parity bit is always a logic 1.

**SPC**
SPaCe: The parity bit is always a logic 0.

**DC**
Do not care: the parity is ignored.

*RST: NONE

**Usage:** Asynchronous command

**Manual operation:** See "Parity" on page 613

---

**SBUS<sb>:UART:POLarity** <Polarity>

Defines the logic levels of the bus. The idle state corresponds to a logic 1. The start bit corresponds to a logic 0. "Idle high" (high=1) is used, for example, for control signals, while "Idle low" (low=1) is defined for data lines (RS-232).

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Polarity> IDLLow | IDLHigh

*RST: IDLHigh

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 613

---

**SBUS<sb>:UART:RX:HYSTeresis** <Rx hysteresis>

Sets the hysteresis for the Tx line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Rx hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 614

---

**SBUS<sb>:UART:RX:SOURce** <RxSource>

Selects the input channel for the receiver signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<RxSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Source: Tx, Rx" on page 612

---

**SBUS<sb>:UART:RX:THReshold** <Rx threshold>

Sets a user-defined threshold value for the Rx line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Rx threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 614

---

**SBUS<sb>:UART:SBIT** <StopBits>

Sets the number of stop bits: 1 or 1.5 or 2 stop bits are possible.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<StopBits> B1 | B15 | B2
*RST: B1

**Usage:** Asynchronous command

**Manual operation:** See "Stop bits" on page 613

---

**SBUS<sb>:UART:SSIZe** <DataBits>

Sets the number of data bits of a word in a range from 5 bits to 8 bits. If no parity bit is used, then 9 data bits are possible.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DataBits> Range: 5 to 9
Increment: 1
*RST: 8

**Usage:** Asynchronous command

**Manual operation:** See "Data bits" on page 613

---

**SBUS<sb>:UART:TOUT** <Timeout>

Sets the timeout between packets in a UART data stream. A new packet starts with the first start bit after the timeout.

The command is relevant if `SBUS<sb>:UART:PACKets` is set to `TOUT`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Timeout>
Range: 1E-06 to 1
Increment: 1
*RST: 0.000625
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Packets" on page 613

---

**SBUS<sb>:UART:TX:HYSTeresis** <Tx hysteresis>

Sets the hysteresis for the TX line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Tx hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 614

---

**SBUS<sb>:UART:TX:SOURce** <TxSource>

Selects the input channel for the transmitter signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<TxSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Source: Tx, Rx" on page 612

**SBUS<sb>:UART:TX:THReshold** <Tx threshold>

Sets a user-defined threshold value for the Tx line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Tx threshold> User-defined clock threshold

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 614

**SBUS<sb>:UART:RX:POSition** <Rx position>

Sets the vertical position of the RX UART signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Rx position>

**Usage:** Asynchronous command

**SBUS<sb>:UART:RX:SCALe** <Rx scale>

Set the vertical scale of the RX UART signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Rx scale>

**Usage:** Asynchronous command

**SBUS<sb>:UART:TX:POSition** <Tx position>

Sets the vertical position of the TX UART signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Tx position>

**Usage:** Asynchronous command

**SBUS<sb>:UART:TX:SCALe** <Tx scale>

Set the vertical scale of the TX UART signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Tx scale>

**Usage:** Asynchronous command

### 18.18.7.2 Filter

There are two commands for each parameter, that you can use for defining the UART settings.

For example, to set the *Frame type =Tx > Field =Data >01100* value you can use one of the following commands:

- `SBUS:UART:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:UART:FILTer:DMIN "Tx", "Data", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:UART:FILTer:CHKall ..... 1539
SBUS<sb>:UART:FILTer:CLR ..... 1540
SBUS<sb>:UART:FILTer:INVert ..... 1540
SBUS<sb>:UART:FILTer:RST ..... 1540
SBUS<sb>:UART:FILTer:IMIN ..... 1540
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:IMIN ..... 1540
SBUS<sb>:UART:FILTer:IMAX ..... 1541
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:IMAX ..... 1541
SBUS<sb>:UART:FILTer:IOPerator ..... 1541
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ..... 1541
SBUS<sb>:UART:FILTer:ERENable ..... 1542
SBUS<sb>:UART:FILTer:ERRor<n>:ENABle ..... 1542
SBUS<sb>:UART:FILTer:DOPerator ..... 1542
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ..... 1542
SBUS<sb>:UART:FILTer:DMIN ..... 1542
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:DMIN ..... 1542
SBUS<sb>:UART:FILTer:DMAX ..... 1543
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:DMAX ..... 1543
SBUS<sb>:UART:FILTer:BIT ..... 1543
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:BIT ..... 1543
SBUS<sb>:UART:FILTer:FRENable ..... 1543
SBUS<sb>:UART:FILTer:FRAMe<fr>:ENABle ..... 1543
SBUS<sb>:UART:FILTer:FIENable ..... 1544
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:ENABle ..... 1544

---

**SBUS<sb>:UART:FILTer:CHKall**

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 615

---

**SBUS<sb>:UART:FILTer:CLR**

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 615

---

**SBUS<sb>:UART:FILTer:INVert**

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 615

---

**SBUS<sb>:UART:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 615

---

**SBUS<sb>:UART:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:UART:FILTer:IMIN?** <Data>
**SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 616

---

**SBUS<sb>:UART:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:UART:FILTer:IMAX?** <Data>
**SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 616

---

**SBUS<sb>:UART:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:UART:FILTer:IOPerator?** <Operator>
**SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 616

**SBUS<sb>:UART:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:UART:FILTer:ERENable?** <Enabler>
**SBUS<sb>:UART:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 617

**SBUS<sb>:UART:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:UART:FILTer:DOPerator?** <Operator>
**SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange
*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 616

**SBUS<sb>:UART:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:UART:FILTer:DMIN?** <Data>
**SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1..4
Selects the serial bus.

<fr> *
Specifies the frame number.

<fl> *
Specifies the field number within the frame.

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 616

---

**SBUS<sb>:UART:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:UART:FILTer:DMAX?** <Data>
**SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 616

---

**SBUS<sb>:UART:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:UART:FILTer:BIT?** <Bit>
**SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO

*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 616

---

**SBUS<sb>:UART:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:UART:FILTer:FRENable?** <Enabler>
**SBUS<sb>:UART:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 616

---

**SBUS<sb>:UART:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:UART:FILTer:FIENable?** <Enabler>
**SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 616

### 18.18.7.3 Format

This section describes the commands to specify and query frame formats for the "UART" protocol.

SBUS<sb>:UART:FORMat:SELect ... 1545
SBUS<sb>:UART:FORMat:ADDFrame ... 1545
SBUS<sb>:UART:FORMat:LOAD ... 1545
SBUS<sb>:UART:FORMat:SAVE ... 1546
SBUS<sb>:UART:FORMat:CLR ... 1546
SBUS<sb>:UART:FORMat:FCOunt? ... 1546
SBUS<sb>:UART:FORMat:FRAMe<fr>:ADDField ... 1546
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLDCount? ... 1547
SBUS<sb>:UART:FORMat:FRAMe<fr>:NAME ... 1547
SBUS<sb>:UART:FORMat:FRAMe<fr>:COLor ... 1547
SBUS<sb>:UART:FORMat:FRAMe<fr>:TYPE ... 1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID1Field ... 1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID2Field ... 1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID3Field ... 1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID1Value ... 1548

SBUS<sb>:UART:FORMat:FRAMe<fr>:ID2Value ..... 1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID3Value ..... 1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:NAME ..... 1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:COLor ..... 1549
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ..... 1549
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:BITCount ..... 1550
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:REPetition ..... 1550
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:BITorder ..... 1550
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FORMat ..... 1551
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ..... 1551
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ..... 1552
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ..... 1552
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:UNIT ..... 1552
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:PARA ..... 1553
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:PARB ..... 1553

---

**SBUS<sb>:UART:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 617

---

**SBUS<sb>:UART:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 618

---

**SBUS<sb>:UART:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 618

---

**SBUS<sb>:UART:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 619

---

**SBUS<sb>:UART:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 618

---

**SBUS<sb>:UART:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 618

---

**SBUS<sb>:UART:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 620

---

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:UART:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 619

---

**SBUS<sb>:UART:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 619

**SBUS<sb>:UART:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 619

**SBUS<sb>:UART:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:UART:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:UART:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 619

**SBUS<sb>:UART:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:UART:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:UART:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 619

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 620

---

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 620

---

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<FirstBit>

Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 620

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 621

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 621

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first
*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 621

---

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first
**MSBF**
Most significant bit first
*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 621

---

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 622

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value>
Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 622

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 622

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 622

---

**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

Available if `SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<ParameterB>
Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 622

### 18.18.7.4 Hardware trigger

Trigger commands for serial buses affect the current trigger source. Therefore, set the trigger source to the required serial bus with `TRIGger:EVENt<ev>:SOURce` before sending the bus-specific trigger commands.

TRIGger:SBHW:UART:DATA....1553
TRIGger:SBHW:UART:DPOSition....1554
TRIGger:SBHW:UART:FCONdition....1554
TRIGger:SBHW:UART:OPERator....1554
TRIGger:SBHW:UART:SOURce....1554
TRIGger:SBHW:UART:TYPE....1555

---

**TRIGger:SBHW:UART:DATA** <DataPattern>

Specifies the data pattern to be found on the specified trigger source. Enter the words in MSB first bit order.

**Parameters:**
<DataPattern>

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 623

---

**TRIGger:SBHW:UART:DPOSition** <DataPosition>

Sets the number of words before the first word of interest. These offset words are ignored.

**Parameters:**

<DataPosition> Range: 1 to 4096
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Position" on page 623

---

**TRIGger:SBHW:UART:FCONdition** <DataOperator>

Selects the operator for the "Data" pattern.

**Parameters:**

<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan
*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 623

---

**TRIGger:SBHW:UART:OPERator** <DataOperator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Parameters:**

<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan
*RST: EQUal

**Usage:** Asynchronous command

---

**TRIGger:SBHW:UART:SOURce** <Source>

Selects the transmitter or receiver line as trigger source.

**Parameters:**

<Source> TX | RX
*RST: TX

**Usage:** Asynchronous command

**TRIGger:SBHW:UART:TYPE** <Type>

Selects the trigger type condition.

**Parameters:**

<Type> STBT | PCKS | DATA | PRER | BRKC | STPerror

STBT: Start bit
PCKS: Packet start
DATA: Serial pattern
PRER: Parity error
BRKC: Break condition
STPerror: Stop error

*RST: STBT

**Usage:** Asynchronous command

### 18.18.7.5 Software trigger

There are two commands for each parameter, that you can use for defining the UART settings.

For example, to set the *Frame type =Tx > Field =Data >01100* value you can use one of the following commands:

- `TRIGger:SBSW:UART:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:SBSW:UART:DMIN "Tx", "Data", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:UART:CHKall ... 1556
TRIGger:SBSW:UART:CLR ... 1556
TRIGger:SBSW:UART:INVert ... 1556
TRIGger:SBSW:UART:RST ... 1556
TRIGger:SBSW:UART:FRENable ... 1556
TRIGger:SBSW:UART:FRAMe<fr>:ENABle ... 1556
TRIGger:SBSW:UART:BIT ... 1557
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:BIT ... 1557
TRIGger:SBSW:UART:DMAX ... 1557
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DMAX ... 1557
TRIGger:SBSW:UART:DMIN ... 1557
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DMIN ... 1557
TRIGger:SBSW:UART:DOPerator ... 1558
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DOPerator ... 1558
TRIGger:SBSW:UART:FIENable ... 1558
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:ENABle ... 1558
TRIGger:SBSW:UART:IMAX ... 1559
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IMAX ... 1559
TRIGger:SBSW:UART:IMIN ... 1559
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IMIN ... 1559
TRIGger:SBSW:UART:IOPerator ... 1559

TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IOPerator ..... 1559
TRIGger:SBSW:UART:ERENable ..... 1560
TRIGger:SBSW:UART:ERRor<m>:ENABle ..... 1560
SBUS<sb>:UART:SWTindex? ..... 1560
SBUS<sb>:UART:SWTTime? ..... 1560

**TRIGger:SBSW:UART:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 624

**TRIGger:SBSW:UART:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 624

**TRIGger:SBSW:UART:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 624

**TRIGger:SBSW:UART:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 624

**TRIGger:SBSW:UART:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:UART:FRENable?** <Enabler>
**TRIGger:SBSW:UART:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 625

---

**TRIGger:SBSW:UART:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:UART:BIT?** <Bit>
**TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 625

---

**TRIGger:SBSW:UART:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:UART:DMAX?** <Data>
**TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 625

---

**TRIGger:SBSW:UART:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:UART:DMIN?** <Data>
**TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 625

---

**TRIGger:SBSW:UART:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:UART:DOPerator?** <Operator>
**TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 625

---

**TRIGger:SBSW:UART:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:UART:FIENable?** <Enabler>
**TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 625

**TRIGger:SBSW:UART:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:UART:IMAX?** <Data>
**TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max>
Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 625

**TRIGger:SBSW:UART:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:UART:IMIN?** <Data>
**TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min>
Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 625

**TRIGger:SBSW:UART:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:UART:IOPerator?** <Operator>
**TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 625

---

**TRIGger:SBSW:UART:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:UART:ERENable?** <Enabler>
**TRIGger:SBSW:UART:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 626

---

**SBUS<sb>:UART:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:UART:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**

<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.7.6 Decode results

SBUS<sb>:UART:WCOunt? 1561
SBUS<sb>:UART:WORD<w>:FLDCount? 1561
SBUS<sb>:UART:WORD<w>:BITRate? 1562
SBUS<sb>:UART:WORD<w>:FLD<fl>:STATe? 1562
SBUS<sb>:UART:WORD<w>:RXValue? 1562
SBUS<sb>:UART:WORD<w>:TXValue? 1563
SBUS<sb>:UART:WORD<w>:SOURce? 1563
SBUS<sb>:UART:WORD<w>:STARt? 1563
SBUS<sb>:UART:WORD<w>:STOP? 1564
SBUS<sb>:UART:WORD<w>:STATe? 1564
SBUS<sb>:UART:WORD<w>:FLD<fl>:LABel? 1565
SBUS<sb>:UART:WORD<w>:FLD<fl>:VALue? 1565
SBUS<sb>:UART:WORD<w>:FLD<fl>:FVALue? 1565

---

**SBUS<sb>:UART:WCOunt?**

Returns the number of words in the acquisition.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Return values:**

<Count> Total number of words.

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:UART:WORD<w>:FLDCount?**

Returns the number of fields available in the seleced word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**

<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:UART:WORD<w>:BITRate?**

Returns the primary bit rate.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> Index of the word

**Return values:**
<PrimaryBitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

**SBUS<sb>:UART:WORD<w>:FLD<fl>:STATe?**

Returns the overall state of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | BREak | STERror | SPERror | PRERror | INComplete

OK: the frame is valid.
`BREak`: stop bit error with 0x00 word
`STERror`: start error, incorrect start bit
`SPERror`: stop error, incorrect stop bit
`PRERror`: parity error, incorrect parity bit.
`INComplete`: The frame is not completely contained in the acquisition. The acquired part of the frame is valid.

*RST: OK

**Usage:** Query only
Asynchronous command

**SBUS<sb>:UART:WORD<w>:RXValue?**

Returns the value of the specified word on the Rx line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> Index of the word

**Return values:**

<RxValue> Range: 0 to 511
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:UART:WORD<w>:TXValue?**

Returns the value of the specified word on the TX line.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> Index of the word

**Return values:**

<TxValue> Range: 0 to 511
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:UART:WORD<w>:SOURce?**

Returns the line on which the specified word was transferred.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> Index of the word

**Return values:**

<WordSource> TX | RX
*RST: TX

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:UART:WORD<w>:STARt?**

Returns the start time of the specified word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> Index of the word

**Return values:**

<WordStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:UART:WORD<w>:STOP?**

Returns the stop time of the specified word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> Index of the word

**Return values:**

<WordStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:UART:WORD<w>:STATe?**

Returns the status of the specified word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> Index of the word

**Return values:**

<WordState> OK | BREak | STERror | SPERror | PRERror | INComplete

OK: the frame is valid.
`BREak`: stop bit error with 0x00 word
`STERror`: start error, incorrect start bit
`SPERror`: stop error, incorrect stop bit
`PRERror`: parity error, incorrect parity bit.
`INComplete`: The frame is not completely contained in the acquisition. The acquired part of the frame is valid.

*RST: OK

**Usage:** Query only
Asynchronous command

**SBUS<sb>:UART:WORD<w>:FLD<fl>:LABel?**

Returns the label of the specified field in the specified word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:UART:WORD<w>:FLD<fl>:VALue?**

Returns the value of the specified field in the specified word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

<fl> Index of the field

**Return values:**
<Value> Range: 0 to 18446744073709551615
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:UART:WORD<w>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field in the specified word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

## 18.18.8 NRZ Clocked (option R&S MXO3-K510)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.


- Configuration....1566
- Filter....1571
- Format....1576
- Software trigger....1585
- Decode results....1590


### 18.18.8.1 Configuration


SBUS<sb>:NRZC:CLK:SOURce....1566
SBUS<sb>:NRZC:CLK:POLarity....1566
SBUS<sb>:NRZC:DATA:SOURce....1567
SBUS<sb>:NRZC:DATA:POLarity....1567
SBUS<sb>:NRZC:ENABle:SOURce....1568
SBUS<sb>:NRZC:ENABle:POLarity....1568
SBUS<sb>:NRZC:MINGap:SELect....1568
SBUS<sb>:NRZC:MINGap:WIDTh....1569
SBUS<sb>:NRZC:CLK:THReshold....1569
SBUS<sb>:NRZC:CLK:HYSTeresis....1569
SBUS<sb>:NRZC:DATA:THReshold....1569
SBUS<sb>:NRZC:DATA:HYSTeresis....1570
SBUS<sb>:NRZC:ENABle:THReshold....1570
SBUS<sb>:NRZC:ENABle:HYSTeresis....1570
SBUS<sb>:NRZC:POSition....1570
SBUS<sb>:NRZC:SCALe....1571


---

**SBUS<sb>:NRZC:CLK:SOURce** Source

Selects the source for the clock signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
Source C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Clock source" on page 632

---

**SBUS<sb>:NRZC:CLK:POLarity** <CLKPolarity>

Sets the polarity for the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CLKPolarity> RISE | FALL | BOTH

**RISE**
Data is sampled at the rising edges of the clock signal.

**FALL**
Data is sampled at the falling edges of the clock signal.

**BOTH**
Data is sampled at the rising and falling edges of the clock signal.

*RST: RISE

**Usage:** Asynchronous command

**Manual operation:** See "Clock polarity" on page 632

---

**SBUS<sb>:NRZC:DATA:SOURce** Source

Selects the channel for the data signal source.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
Source C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Data source" on page 632

---

**SBUS<sb>:NRZC:DATA:POLarity** <DATAPolarity>

Sets the polarity for the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DATAPolarity> ACTLow | ACTHigh

**ACTLow**
The transmitted signal for the data line is active high (high = 1).

**ACTHigh**
The transmitted signal for the data line is active low (low = 1).

*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Data polarity" on page 632

**SBUS<sb>:NRZC:ENABle:SOURce** Source

Selects the channel for the enable signal source.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
Source C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Enable source" on page 633

**SBUS<sb>:NRZC:ENABle:POLarity** <ENABPolarity>

Sets the polarity for the enable line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ENABPolarity> ACTLow | ACTHigh

**ACTLow**
The transmitted signal for the enable line is active high (high = 1).

**ACTHigh**
The transmitted signal for the enable line is active low (low = 1).

*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Enable source" on page 633

**SBUS<sb>:NRZC:MINGap:SELect** <MingapSelect>

Enables checking the minimum idle time between two frames during decoding.

To set the idle time, use the command `SBUS<sb>:NRZC:MINGap:WIDTh`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MingapSelect> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Check gap time" on page 633

**SBUS<sb>:NRZC:MINGap:WIDTh** <MingapWidth>

Sets the minimum duration of the idle time. Any inactivity greater than this time is interpreted as a gap and leads to a resynchronization to the signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MingapWidth>
Range: 1E-09 to 1
Increment: 1E-09
*RST: 1E-05
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Check gap time" on page 633

**SBUS<sb>:NRZC:CLK:THReshold** <CLKThreshold>

Sets the threshold value for the digitization of the clock signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CLKThreshold>

**Usage:** Asynchronous command

**Manual operation:** See "Clock threshold, Clock hysteresis" on page 634

**SBUS<sb>:NRZC:CLK:HYSTeresis** <CLKHysteresis>

Sets a value for the hysteresis of the clock signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CLKHysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Clock threshold, Clock hysteresis" on page 634

**SBUS<sb>:NRZC:DATA:THReshold** <SDAThreshold>

Sets the threshold for the data channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SDAThreshold>

**Usage:** Asynchronous command

**Manual operation:** See "Data threshold, Data hysteresis" on page 634

---

**SBUS<sb>:NRZC:DATA:HYSTeresis** <SDAHysteresis>

Sets the hysteresis for the data channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SDAHysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Data threshold, Data hysteresis" on page 634

---

**SBUS<sb>:NRZC:ENABle:THReshold** <ENABThreshold>

Sets the threshold for the enable channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ENABThreshold>

**Usage:** Asynchronous command

**Manual operation:** See "Enable threshold, Enable hysteresis" on page 634

---

**SBUS<sb>:NRZC:ENABle:HYSTeresis** <ENABHysteresis>

Sets the hysteresis for the enable channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ENABHysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Enable threshold, Enable hysteresis" on page 634

---

**SBUS<sb>:NRZC:POSition** <Position>

Sets the vertical position of the NRZ clocked signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Position>

**Usage:** Asynchronous command

---

**SBUS<sb>:NRZC:SCALe** <Scale>

Sets the vertical scale of the NRZ clocked signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Scale>

**Usage:** Asynchronous command

### 18.18.8.2 Filter

There are two commands for each parameter, that you can use for defining the NRZ clocked settings.

For example, to set the *Frame type =NRZC-Frame > Field =Data >Data* value you can use one of the following commands:

- `SBUS:NRZC:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:NRZC:FILTer:DMIN "NRZC-Frame", "Data", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:NRZC:FILTer:CHKall....1572
SBUS<sb>:NRZC:FILTer:CLR....1572
SBUS<sb>:NRZC:FILTer:INVert....1572
SBUS<sb>:NRZC:FILTer:RST....1572
SBUS<sb>:NRZC:FILTer:BIT....1572
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:BIT....1572
SBUS<sb>:NRZC:FILTer:ERENable....1573
SBUS<sb>:NRZC:FILTer:ERRor<n>:ENABle....1573
SBUS<sb>:NRZC:FILTer:DMAX....1573
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1573
SBUS<sb>:NRZC:FILTer:DMIN....1574
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1574
SBUS<sb>:NRZC:FILTer:DOPerator....1574
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1574
SBUS<sb>:NRZC:FILTer:FIENable....1574
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1574
SBUS<sb>:NRZC:FILTer:FRENable....1575
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:ENABle....1575
SBUS<sb>:NRZC:FILTer:IMAX....1575
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1575
SBUS<sb>:NRZC:FILTer:IMIN....1576
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1576
SBUS<sb>:NRZC:FILTer:IOPerator....1576
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1576

**SBUS<sb>:NRZC:FILTer:CHKall**

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 636

**SBUS<sb>:NRZC:FILTer:CLR**

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 636

**SBUS<sb>:NRZC:FILTer:INVert**

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 636

**SBUS<sb>:NRZC:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 636

**SBUS<sb>:NRZC:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:NRZC:FILTer:BIT?** <Bit>
**SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

---

**SBUS<sb>:NRZC:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:NRZC:FILTer:ERENable?** <Enabler>
**SBUS<sb>:NRZC:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<n> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 639

---

**SBUS<sb>:NRZC:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:NRZC:FILTer:DMAX?** <Data>
**SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

**SBUS<sb>:NRZC:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:NRZC:FILTer:DMIN?** <Data>
**SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

**SBUS<sb>:NRZC:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:NRZC:FILTer:DOPerator?** <Operator>
**SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

**SBUS<sb>:NRZC:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:NRZC:FILTer:FIENable?** <Enabler>
**SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

---

**SBUS<sb>:NRZC:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:NRZC:FILTer:FRENable?** <Enabler>
**SBUS<sb>:NRZC:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables the filtering on NRZ clocked frames. Only the frames that match the selected filter conditions are displayed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 636

---

**SBUS<sb>:NRZC:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:NRZC:FILTer:IMAX?** <Data>
**SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

You can set the operator with `SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

**SBUS<sb>:NRZC:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:NRZC:FILTer:IMIN?** <Data>
**SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

**SBUS<sb>:NRZC:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:NRZC:FILTer:IOPerator?** <Operator>
**SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

### 18.18.8.3 Format

This section describes the commands to specify and query frame formats for the "NRZ Clocked" protocol.

For manual operation, refer to Section 14.8.4, "NRZ format configuration", on page 639.

SBUS<sb>:NRZC:FORMat:ADDFrame ..... 1577
SBUS<sb>:NRZC:FORMat:LOAD ..... 1577
SBUS<sb>:NRZC:FORMat:SAVE ..... 1577
SBUS<sb>:NRZC:FORMat:CLR ..... 1578
SBUS<sb>:NRZC:FORMat:FCOunt? ..... 1578

SBUS<sb>:NRZC:FORMat:FRAMe<fr>:ADDField....1578
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLDCount?....1578
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:NAME....1579
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:COLor....1579
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:NAME....1579
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:COLor....1580
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1580
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:CONDition....1580
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1581
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1582
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:CLMN....1582
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1583
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1583
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1583
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1584
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:PARA....1584
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:PARB....1584

**SBUS<sb>:NRZC:FORMat:ADDFrame**

Appendes a new frame description to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add Frame" on page 639

**SBUS<sb>:NRZC:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame descriptions.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 640

**SBUS<sb>:NRZC:FORMat:SAVE** <FileName>

Saves the current list of frame descriptions to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 641

---

**SBUS<sb>:NRZC:FORMat:CLR**

Erases from the volatile memory all field and frame format descriptions that you have created for a specific custom protocol.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit List" on page 640

---

**SBUS<sb>:NRZC:FORMat:FCOunt?**

Returns the number of frame format descriptions available for a specific custom protocol.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit List" on page 640

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame description.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add Field" on page 641

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit List" on page 642

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the frame description of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 641

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the frame description of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame Color" on page 641

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 642

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field Color" on page 642

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitCount>

Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 644

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:CONDition** <Condition>

Specifies a user-defined condition operator for the selected field of the selected frame. The various condition operators can identify, for example, a mandatory CRC checksum value or a frame ID.

Set the numeric format of the condition by the command `SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:FORMat`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Condition> The implemented conditions are the "equal" and "array" operators.

**Usage:** Asynchronous command

**Manual operation:** See "Condition" on page 642

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 645

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 644

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:CLMN** <Column>

Specifies, in which result column of the decode table to display the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Column> NONE | COL1 | COL2 | COL3

**NONE**
The result is not displayed.

**COL1**
The result is displayed in column 1.

**COL2**
The result is displayed in column 2.

**COL3**
The result is displayed in column 3.

*RST: NONE

**Usage:** Asynchronous command

**Manual operation:** See "Result column" on page 644

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 645

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 645

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**
<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 645

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 646

---

**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

Available if `SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<ParameterB> Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 646

### 18.18.8.4 Software trigger

There are two commands for each parameter that you can use for defining the "NRZ Clocked" settings.

For example, to set the *Frame type =NRZC-Frame* > *Field =Data* >*Data* value you can use one of the following commands:

- `TRIGger:SBSW:NRZC:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <fr> for the frame number and <fl> for the field number.
- `TRIGger:SBSW:NRZC:DMIN "NRZC-Frame", "Data", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:NRZC:CHKall....1585
TRIGger:SBSW:NRZC:CLR....1586
TRIGger:SBSW:NRZC:INVert....1586
TRIGger:SBSW:NRZC:RST....1586
TRIGger:SBSW:NRZC:FRENable....1586
TRIGger:SBSW:NRZC:FRAMe<fr>:ENABle....1586
TRIGger:SBSW:NRZC:BIT....1586
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:BIT....1586
TRIGger:SBSW:NRZC:DMAX....1587
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DMAX....1587
TRIGger:SBSW:NRZC:DMIN....1587
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DMIN....1587
TRIGger:SBSW:NRZC:DOPerator....1588
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DOPerator....1588
TRIGger:SBSW:NRZC:FIENable....1588
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:ENABle....1588
TRIGger:SBSW:NRZC:IMAX....1588
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IMAX....1588
TRIGger:SBSW:NRZC:IMIN....1589
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IMIN....1589
TRIGger:SBSW:NRZC:IOPerator....1589
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IOPerator....1589
TRIGger:SBSW:NRZC:ERENable....1589
TRIGger:SBSW:NRZC:ERRor<m>:ENABle....1589
SBUS<sb>:NRZC:SWTindex?....1590
SBUS<sb>:NRZC:SWTTime?....1590

---

**TRIGger:SBSW:NRZC:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 647

**TRIGger:SBSW:NRZC:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 647

**TRIGger:SBSW:NRZC:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 647

**TRIGger:SBSW:NRZC:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 647

**TRIGger:SBSW:NRZC:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:NRZC:FRENable?** <Enabler>
**TRIGger:SBSW:NRZC:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 648

**TRIGger:SBSW:NRZC:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:NRZC:BIT?** <Bit>
**TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

---

**TRIGger:SBSW:NRZC:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:NRZC:DMAX?** <Data>
**TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

---

**TRIGger:SBSW:NRZC:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:NRZC:DMIN?** <Data>
**TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

**TRIGger:SBSW:NRZC:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:NRZC:DOPerator?** <Operator>
**TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

**TRIGger:SBSW:NRZC:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:NRZC:FIENable?** <Enabler>
**TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

**TRIGger:SBSW:NRZC:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:NRZC:IMAX?** <Data>
**TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

---

**TRIGger:SBSW:NRZC:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:NRZC:IMIN?** <Data>
**TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

---

**TRIGger:SBSW:NRZC:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:NRZC:IOPerator?** <Operator>
**TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

---

**TRIGger:SBSW:NRZC:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:NRZC:ERENable?** <Enabler>
**TRIGger:SBSW:NRZC:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 650

---

**SBUS<sb>:NRZC:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZC:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.8.5 Decode results

This section describes the commands to query the results of the decoding of the "NRZ Clocked" protocol.

For manual operation, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

SBUS<sb>:NRZC:FCOunt? 1591
SBUS<sb>:NRZC:FRAMe<fr>:FLDCount? 1591
SBUS<sb>:NRZC:FRAMe<fr>:WBRate? 1591
SBUS<sb>:NRZC:FRAMe<fr>:FLD<fl>:STATe? 1592
SBUS<sb>:NRZC:FRAMe<fr>:FLD<fl>:LABel? 1592
SBUS<sb>:NRZC:FRAMe<fr>:FLD<fl>:VALue? 1593
SBUS<sb>:NRZC:FRAMe<fr>:FLD<fl>:FVALue? 1593
SBUS<sb>:NRZC:FRAMe<fr>:STATus? 1593
SBUS<sb>:NRZC:FRAMe<fr>:STARt? 1594
SBUS<sb>:NRZC:FRAMe<fr>:STOP? 1594
SBUS<sb>:NRZC:FRAMe<fr>:TYPE? 1594
SBUS<sb>:NRZC:FRAMe<fr>:CONE? 1595
SBUS<sb>:NRZC:FRAMe<fr>:CTWO? 1595
SBUS<sb>:NRZC:FRAMe<fr>:CTHRee? 1595

### SBUS<sb>:NRZC:FCOunt?

Returns the number of decoded frames.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

### SBUS<sb>:NRZC:FRAMe<fr>:FLDCount?

Returns the number of fields available in the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<DataCount>

**Usage:** Query only
Asynchronous command

### SBUS<sb>:NRZC:FRAMe<fr>:WBRate?

Returns the bit rate of the frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameWBR> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**SBUS<sb>:NRZC:FRAMe<fr>:FLD<fl>:STATe?**

Returns the overall state of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | LENGth | CRC | PARity | INComplete

**OK**
The field has no error, it is valid.

**LENGth**
The length of the field is not as expected, indicating an error.

**CRC**
The CRC of the frame is not as expected, indicating an error.

**PARity**
Parity is not as expected, indicating an error

**INComplete**
The frame is incomplete.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZC:FRAMe<fr>:FLD<fl>:LABel?**

Returns the label of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Name>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:NRZC:FRAMe<fr>:FLD<fl>:VALue?**

Returns the value of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> *RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:NRZC:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:NRZC:FRAMe<fr>:STATus?**

Returns the overall state of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameState> OK | LENGth | CRC | PARity | INComplete

**OK**
The frame is valid.

**LENGth**
Length of the frame is not as expected, indicating an error.

**CRC**
The checksum of the frame is not as expected, indicating an error.

**PARity**
Parity is not as expected, indicating an error.

**INComplete**
The frame is incomplete.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZC:FRAMe<fr>:STARt?**

Returns the start time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZC:FRAMe<fr>:STOP?**

Returns the end time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZC:FRAMe<fr>:TYPE?**

Returns the type of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameType> The frame type is undefined, because it is custom specific.

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZC:FRAMe<fr>:CONE?**

Returns the content of the 1st result column of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Result1>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZC:FRAMe<fr>:CTWO?**

Returns the content of the 2nd result column of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Result2>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZC:FRAMe<fr>:CTHRee?**

Returns the content of the 3rd result column of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Result3>

**Usage:** Query only
Asynchronous command

### 18.18.9 NRZ Unclocked (option R&S MXO3-K510)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.

- The bus is set to the correct type with `SBUS<sb>:TYPE`.

- Configuration....1596
- Filter....1600
- Format....1606
- Software trigger....1614
- Decode results....1620

### 18.18.9.1 Configuration

SBUS<sb>:NRZU:DATA:SOURce....1596
SBUS<sb>:NRZU:DATA:POLarity....1596
SBUS<sb>:NRZU:ENABle:SOURce....1597
SBUS<sb>:NRZU:ENABle:POLarity....1597
SBUS<sb>:NRZU:MINGap:SELect....1597
SBUS<sb>:NRZU:MINGap:WIDTh....1598
SBUS<sb>:NRZU:BITRate:SELect....1598
SBUS<sb>:NRZU:BITRate:WIDTh....1598
SBUS<sb>:NRZU:DATA:THReshold....1599
SBUS<sb>:NRZU:DATA:HYSTeresis....1599
SBUS<sb>:NRZU:ENABle:THReshold....1599
SBUS<sb>:NRZU:ENABle:HYSTeresis....1599
SBUS<sb>:NRZU:POSition....1600
SBUS<sb>:NRZU:SCALe....1600

---

**SBUS<sb>:NRZU:DATA:SOURce** Source

Selects the channel for the data signal source.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
Source C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Data source" on page 632

---

**SBUS<sb>:NRZU:DATA:POLarity** <DATAPolarity>

Sets the polarity for the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DATAPolarity> ACTLow | ACTHigh

**ACTLow**
The transmitted signal for the data line is active high (high = 1).

**ACTHigh**
The transmitted signal for the data line is active low (low = 1).

*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Data polarity" on page 632

---

**SBUS<sb>:NRZU:ENABle:SOURce** Source

Selects the channel for the enable signal source.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
Source C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Enable source" on page 633

---

**SBUS<sb>:NRZU:ENABle:POLarity** <ENABPolarity>

Sets the polarity for the enable line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ENABPolarity> ACTLow | ACTHigh

**ACTLow**
The transmitted signal for the enable line is active high (high = 1).

**ACTHigh**
The transmitted signal for the enable line is active low (low = 1).

*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Enable source" on page 633

---

**SBUS<sb>:NRZU:MINGap:SELect** <MingapSelect>

Enables checking the minimum idle time between two frames during decoding.

To set the idle time, use the command `SBUS<sb>:NRZU:MINGap:WIDTh`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MingapSelect> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Check gap time" on page 633

---

**SBUS<sb>:NRZU:MINGap:WIDTh** <MingapWidth>

Sets the minimum duration of the idle time. Any inactivity greater than this time is interpreted as a gap and leads to a resynchronization to the signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MingapWidth> Range: 1E-09 to 1
Increment: 1E-09
*RST: 1E-05
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Check gap time" on page 633

---

**SBUS<sb>:NRZU:BITRate:SELect** <BitrateSelect>

Enables setting the bit rate, which is required to trigger and decode unclocked NRZ signals.

You can set the bit rate with `SBUS<sb>:NRZU:BITRate:WIDTh`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<BitrateSelect> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Check bit rate" on page 633

---

**SBUS<sb>:NRZU:BITRate:WIDTh** <BitrateWidth>

Specifies the bit rate, if enabled with `SBUS<sb>:NRZU:BITRate:SELect`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**

<BitrateWidth>

Range: 300 to 2000000000
Increment: 100000
*RST: 10000000
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "Check bit rate" on page 633

---

**SBUS<sb>:NRZU:DATA:THReshold** <SDAThreshold>

Sets the threshold for the data channel.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<SDAThreshold>

**Usage:** Asynchronous command

**Manual operation:** See "Data threshold, Data hysteresis" on page 634

---

**SBUS<sb>:NRZU:DATA:HYSTeresis** <SDAHysteresis>

Sets the hysteresis for the data channel.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<SDAHysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Data threshold, Data hysteresis" on page 634

---

**SBUS<sb>:NRZU:ENABle:THReshold** <ENABThreshold>

Sets the threshold for the enable channel.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<ENABThreshold>

**Usage:** Asynchronous command

---

**SBUS<sb>:NRZU:ENABle:HYSTeresis** <ENABHysteresis>

Sets the hysteresis for the enable channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ENABHysteresis>

**Usage:** Asynchronous command

---

**SBUS<sb>:NRZU:POSition** <Position>

Sets the vertical position of the NRZ unclocked signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Position>

**Usage:** Asynchronous command

---

**SBUS<sb>:NRZU:SCALe** <Scale>

Sets the vertical scale of the NRZ unclocked signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Scale>

**Usage:** Asynchronous command

### 18.18.9.2 Filter

There are two commands for each parameter, that you can use for defining the NRZ unclocked settings.

For example, to set the *Frame type =NRZU-Frame > Field =Data >Data* value you can use one of the following commands:

- `SBUS:NRZU:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:NRZU:FILTer:DMIN "NRZC-Frame", "Data", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:NRZU:FILTer:CHKall....1601
SBUS<sb>:NRZU:FILTer:CLR....1601
SBUS<sb>:NRZU:FILTer:INVert....1601
SBUS<sb>:NRZU:FILTer:RST....1602
SBUS<sb>:NRZU:FILTer:BIT....1602
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:BIT....1602
SBUS<sb>:NRZU:FILTer:ERENable....1602
SBUS<sb>:NRZU:FILTer:ERRor<n>:ENABle....1602

SBUS<sb>:NRZU:FILTer:DMAX ... 1603
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1603
SBUS<sb>:NRZU:FILTer:DMIN ... 1603
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1603
SBUS<sb>:NRZU:FILTer:DOPerator ... 1603
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1603
SBUS<sb>:NRZU:FILTer:FIENable ... 1604
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1604
SBUS<sb>:NRZU:FILTer:FRENable ... 1604
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:ENABle ... 1604
SBUS<sb>:NRZU:FILTer:IMAX ... 1604
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1604
SBUS<sb>:NRZU:FILTer:IMIN ... 1605
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1605
SBUS<sb>:NRZU:FILTer:IOPerator ... 1605
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1605

### SBUS<sb>:NRZU:FILTer:CHKall

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 636

### SBUS<sb>:NRZU:FILTer:CLR

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 636

### SBUS<sb>:NRZU:FILTer:INVert

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 636

**SBUS<sb>:NRZU:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 636

**SBUS<sb>:NRZU:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:NRZU:FILTer:BIT?** <Bit>
**SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

**SBUS<sb>:NRZU:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:NRZU:FILTer:ERENable?** <Enabler>
**SBUS<sb>:NRZU:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 639

**SBUS<sb>:NRZU:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:NRZU:FILTer:DMAX?** <Data>
**SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

**SBUS<sb>:NRZU:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:NRZU:FILTer:DMIN?** <Data>
**SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

**SBUS<sb>:NRZU:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:NRZU:FILTer:DOPerator?** <Operator>
**SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**

<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

---

**SBUS<sb>:NRZU:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:NRZU:FILTer:FIENable?** <Enabler>
**SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

---

**SBUS<sb>:NRZU:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:NRZU:FILTer:FRENable?** <Enabler>
**SBUS<sb>:NRZU:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables the filtering on NRZ unclocked frames. Only the frames that match the selected filter conditions are displayed.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**

<Enable> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 636

---

**SBUS<sb>:NRZU:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:NRZU:FILTer:IMAX?** <Data>
**SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

You can set the operator with `SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

---

**SBUS<sb>:NRZU:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:NRZU:FILTer:IMIN?** <Data>
**SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

---

**SBUS<sb>:NRZU:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:NRZU:FILTer:IOPerator?** <Operator>
**SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 637

### 18.18.9.3 Format

This section describes the commands to specify and query frame formats for the "NRZ Unclocked" protocol.

For manual operation, refer to Section 14.8.4, "NRZ format configuration", on page 639.

SBUS<sb>:NRZU:FORMat:ADDFrame ..... 1606
SBUS<sb>:NRZU:FORMat:LOAD ..... 1606
SBUS<sb>:NRZU:FORMat:SAVE ..... 1607
SBUS<sb>:NRZU:FORMat:CLR ..... 1607
SBUS<sb>:NRZU:FORMat:FCOunt? ..... 1607
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:ADDField ..... 1608
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLDCount? ..... 1608
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:NAME ..... 1608
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:COLor ..... 1608
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:NAME ..... 1609
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:COLor ..... 1609
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:BITCount ..... 1609
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:CONDition ..... 1610
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:FORMat ..... 1610
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:BITorder ..... 1611
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:CLMN ..... 1611
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ..... 1612
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ..... 1612
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ..... 1613
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:UNIT ..... 1613
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:PARA ..... 1613
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:PARB ..... 1613

---

**SBUS<sb>:NRZU:FORMat:ADDFrame**

Appendes a new frame description to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add Frame" on page 639

---

**SBUS<sb>:NRZU:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame descriptions.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 640

---

**SBUS<sb>:NRZU:FORMat:SAVE** <FileName>

Saves the current list of frame descriptions to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 641

---

**SBUS<sb>:NRZU:FORMat:CLR**

Erases from the volatile memory all field and frame format descriptions that you have created for a specific custom protocol.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit List" on page 640

---

**SBUS<sb>:NRZU:FORMat:FCOunt?**

Returns the number of frame format descriptions available for a specific custom protocol.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit List" on page 640

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame description.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add Field" on page 641

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit List" on page 642

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the frame description of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 641

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the frame description of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**

<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame Color" on page 641

---

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 642

---

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field Color" on page 642

---

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 644

---

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:CONDition** <Condition>

Specifies a user-defined condition operator for the selected field of the selected frame. The various condition operators can identify, for example, a mandatory CRC checksum value or a frame ID.

Set the numeric format of the condition by the command `SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:FORMat`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Condition> The implemented conditions are the "equal" and "array" operators.

**Usage:** Asynchronous command

**Manual operation:** See "Condition" on page 642

---

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 645

---

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 644

---

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:CLMN** <Column>

Specifies, in which result column of the decode table to display the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Column> NONE | COL1 | COL2 | COL3

**NONE**
The result is not displayed.

**COL1**
The result is displayed in column 1.

**COL2**
The result is displayed in column 2.

**COL3**
The result is displayed in column 3.

*RST: NONE

**Usage:** Asynchronous command

**Manual operation:** See "Result column" on page 644

---

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 645

---

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 645

---

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 645

---

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 646

---

**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)

- Parameter B: Offset

Available if `SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

| | |
|---|---|
| <sb> | 1 to 4, index of the serial bus |
| <fr> | Index of the frame |
| <fl> | Index of the field |

**Parameters:**

| | | |
|---|---|---|
| <ParameterB> | Range: | -1E+20 to 1E+20 |
| | Increment: | 1E-12 |
| | *RST: | 0 |

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 646

### 18.18.9.4 Software trigger

There are two commands for each parameter that you can use for defining the "NRZ Unclocked" settings.

For example, to set the *Frame type =NRZU-Frame > Field =Data >Data* value you can use one of the following commands:

- `TRIGger:SBSW:NRZU:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <fr> for the frame number and <fl> for the field number.
- `TRIGger:SBSW:NRZU:DMIN "NRZC-Frame", "Data", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:NRZU:CHKall....1615
TRIGger:SBSW:NRZU:CLR....1615
TRIGger:SBSW:NRZU:INVert....1615
TRIGger:SBSW:NRZU:RST....1615
TRIGger:SBSW:NRZU:FRENable....1615
TRIGger:SBSW:NRZU:FRAMe<fr>:ENABle....1615
TRIGger:SBSW:NRZU:BIT....1616
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:BIT....1616
TRIGger:SBSW:NRZU:DMAX....1616
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DMAX....1616
TRIGger:SBSW:NRZU:DMIN....1616
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DMIN....1616
TRIGger:SBSW:NRZU:DOPerator....1617
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DOPerator....1617
TRIGger:SBSW:NRZU:FIENable....1617
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:ENABle....1617
TRIGger:SBSW:NRZU:IMAX....1618
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IMAX....1618
TRIGger:SBSW:NRZU:IMIN....1618
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IMIN....1618

TRIGger:SBSW:NRZU:IOPerator....1618
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IOPerator....1618
TRIGger:SBSW:NRZU:ERENable....1619
TRIGger:SBSW:NRZU:ERRor<m>:ENABle....1619
SBUS<sb>:NRZU:SWTindex?....1619
SBUS<sb>:NRZU:SWTTime?....1619

**TRIGger:SBSW:NRZU:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 647

**TRIGger:SBSW:NRZU:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 647

**TRIGger:SBSW:NRZU:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 647

**TRIGger:SBSW:NRZU:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 647

**TRIGger:SBSW:NRZU:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:NRZU:FRENable?** <Enabler>
**TRIGger:SBSW:NRZU:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 648

---

**TRIGger:SBSW:NRZU:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:NRZU:BIT?** <Bit>
**TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

---

**TRIGger:SBSW:NRZU:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:NRZU:DMAX?** <Data>
**TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

---

**TRIGger:SBSW:NRZU:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:NRZU:DMIN?** <Data>
**TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

---

**TRIGger:SBSW:NRZU:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:NRZU:DOPerator?** <Operator>
**TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

---

**TRIGger:SBSW:NRZU:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:NRZU:FIENable?** <Enabler>
**TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

**TRIGger:SBSW:NRZU:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:NRZU:IMAX?** <Data>
**TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max>
Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

**TRIGger:SBSW:NRZU:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:NRZU:IMIN?** <Data>
**TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min>
Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

**TRIGger:SBSW:NRZU:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:NRZU:IOPerator?** <Operator>
**TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 648

---

**TRIGger:SBSW:NRZU:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:NRZU:ERENable?** <Enabler>
**TRIGger:SBSW:NRZU:ERROr<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 650

---

**SBUS<sb>:NRZU:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZU:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**

<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.9.5 Decode results

This section describes the commands to query the results of the decoding of the "NRZ Unclocked" protocol.

For manual operation, see Section 14.1.9.3, "Decoding protocol signals", on page 512.

SBUS<sb>:NRZU:FCOunt? 1620
SBUS<sb>:NRZU:FRAMe<fr>:FLDCount? 1620
SBUS<sb>:NRZU:FRAMe<fr>:WBRate? 1621
SBUS<sb>:NRZU:FRAMe<fr>:FLD<fl>:STATe? 1621
SBUS<sb>:NRZU:FRAMe<fr>:FLD<fl>:LABel? 1622
SBUS<sb>:NRZU:FRAMe<fr>:FLD<fl>:VALue? 1622
SBUS<sb>:NRZU:FRAMe<fr>:FLD<fl>:FVALue? 1622
SBUS<sb>:NRZU:FRAMe<fr>:STATus? 1622
SBUS<sb>:NRZU:FRAMe<fr>:STARt? 1623
SBUS<sb>:NRZU:FRAMe<fr>:STOP? 1623
SBUS<sb>:NRZU:FRAMe<fr>:TYPE? 1624
SBUS<sb>:NRZU:FRAMe<fr>:CONE? 1624
SBUS<sb>:NRZU:FRAMe<fr>:CTWO? 1624
SBUS<sb>:NRZU:FRAMe<fr>:CTHRee? 1625

---

**SBUS<sb>:NRZU:FCOunt?**

Returns the number of decoded frames.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Return values:**

<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZU:FRAMe<fr>:FLDCount?**

Returns the number of fields available in the seleced frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<DataCount>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZU:FRAMe<fr>:WBRate?**

Returns the bit rate of the frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameWBR> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**SBUS<sb>:NRZU:FRAMe<fr>:FLD<fl>:STATe?**

Returns the overall state of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | LENGth | CRC | PARity | INComplete

**OK**
The field has no error, it is valid.

**LENGth**
The length of the field is not as expected, indicating an error.

**CRC**
The CRC of the frame is not as expected, indicating an error.

**PARity**
Parity is not as expected, indicating an error

**INComplete**
The frame is incomplete.

*RST: OK

**Usage:** Query only
Asynchronous command

**SBUS<sb>:NRZU:FRAMe<fr>:FLD<fl>:LABel?**

Returns the label of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Name>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:NRZU:FRAMe<fr>:FLD<fl>:VALue?**

Returns the value of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> *RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:NRZU:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:NRZU:FRAMe<fr>:STATus?**

Returns the overall state of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameState> OK | LENGth | CRC | PARity | INComplete

**OK**
The frame is valid.

**LENGth**
Length of the frame is not as expected, indicating an error.

**CRC**
The checksum of the frame is not as expected, indicating an error.

**PARity**
Parity is not as expected, indicating an error.

**INComplete**
The frame is incomplete.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZU:FRAMe<fr>:STARt?**

Returns the start time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZU:FRAMe<fr>:STOP?**

Returns the end time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZU:FRAMe<fr>:TYPE?**

Returns the type of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameType> The frame type is undefined, because it is custom specific.

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZU:FRAMe<fr>:CONE?**

Returns the content of the 1st result column of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Result1>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:NRZU:FRAMe<fr>:CTWO?**

Returns the content of the 2nd result column of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Result2>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:NRZU:FRAMe<fr>:CTHRee?**

Returns the content of the 3rd result column of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Result3>

**Usage:** Query only
Asynchronous command

## 18.18.10 Manchester (option R&S MXO3-K510)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.


- Configuration....1625
- Filter....1630
- Format....1636
- Software trigger....1644
- Decode results....1650


### 18.18.10.1 Configuration


SBUS<sb>:MANCh:DATA:SOURce....1625
SBUS<sb>:MANCh:DATA:POLarity....1626
SBUS<sb>:MANCh:DATA:PHASe....1626
SBUS<sb>:MANCh:ENABle:SOURce....1627
SBUS<sb>:MANCh:ENABle:POLarity....1627
SBUS<sb>:MANCh:MINGap:SELect....1627
SBUS<sb>:MANCh:MINGap:WIDTh....1628
SBUS<sb>:MANCh:BITRate:SELect....1628
SBUS<sb>:MANCh:BITRate:WIDTh....1628
SBUS<sb>:MANCh:DATA:THUPper....1628
SBUS<sb>:MANCh:DATA:THLower....1629
SBUS<sb>:MANCh:DATA:HYSTeresis....1629
SBUS<sb>:MANCh:ENABle:THReshold....1629
SBUS<sb>:MANCh:ENABle:HYSTeresis....1629
SBUS<sb>:MANCh:POSition....1630
SBUS<sb>:MANCh:SCALe....1630


**SBUS<sb>:MANCh:DATA:SOURce** <DATASource>

Selects the channel for the data signal source.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DATASource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Data source" on page 657

---

**SBUS<sb>:MANCh:DATA:POLarity** <DataPolarity>

Sets the polarity for the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DataPolarity> MANC | MANT

**MANC**
Selects the "Manchester" data representation convention as per G. E. Thomas: High-to-low transition for logical 1.

**MANT**
Selects the "Manchester II" (Manchester Two) data representation convention as per IEEE 802.3: Low-to-high transition for logical 1.

*RST: MANC

**Usage:** Asynchronous command

**Manual operation:** See "Data polarity" on page 657

---

**SBUS<sb>:MANCh:DATA:PHASe** <DataPhase>

Sets the phase for the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DataPhase> FEDGe | SEDGe

**FEDGe**
Selects capturing data bits on the clock's first (rising) edge.

**SEDGe**
Selects capturing data bits on the clock's second (falling) edge.

*RST: SEDGe

**Usage:** Asynchronous command

**Manual operation:** See "Data phase" on page 657

**SBUS<sb>:MANCh:ENABle:SOURce** <DATASource>

Selects the channel for the enable signal source.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DATASource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Enable source" on page 657

**SBUS<sb>:MANCh:ENABle:POLarity** <EnablePolarity>

Sets the polarity for the enable line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<EnablePolarity> ACTLow | ACTHigh

**ACTLow**
The transmitted signal for the enable line is active high (high = 1).

**ACTHigh**
The transmitted signal for the enable line is active low (low = 1).

*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Enable source" on page 657

**SBUS<sb>:MANCh:MINGap:SELect** <MingapSelect>

Enables checking the minimum idle time between two frames during decoding.

To set the idle time, use the command `SBUS<sb>:MANCh:MINGap:WIDTh`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MingapSelect> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Check gap time" on page 658

**SBUS<sb>:MANCh:MINGap:WIDTh** <MingapWidth>

Sets the minimum duration of the idle time. Any inactivity greater than this time is interpreted as a gap and leads to a resynchronization to the signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MingapWidth> Range: 1E-09 to 1
Increment: 1E-09
*RST: 1E-05
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Check gap time" on page 658

**SBUS<sb>:MANCh:BITRate:SELect** <BitrateSelect>

Enables setting the bit rate.

To set the bit rate, use the command `SBUS<sb>:MANCh:BITRate:WIDTh`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<BitrateSelect> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Check bit rate" on page 658

**SBUS<sb>:MANCh:BITRate:WIDTh** <BitrateWidth>

Sets the bit rate.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<BitrateWidth> Range: 300 to 2000000000
Increment: 100000
*RST: 10000000
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "Check bit rate" on page 658

**SBUS<sb>:MANCh:DATA:THUPper** <DatThresUpp>

Sets the upper threshold for the data channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DatThresUpp>

**Usage:** Asynchronous command

**Manual operation:** See "Upper threshold, Lower threshold, Hysteresis" on page 658

---

**SBUS<sb>:MANCh:DATA:THLower** <DatThresLow>

Sets the lower threshold for the data channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DatThresLow>

**Usage:** Asynchronous command

**Manual operation:** See "Upper threshold, Lower threshold, Hysteresis" on page 658

---

**SBUS<sb>:MANCh:DATA:HYSTeresis** <DataHysteresis>

Sets the hysteresis for the data channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DataHysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Upper threshold, Lower threshold, Hysteresis" on page 658

---

**SBUS<sb>:MANCh:ENABle:THReshold** <EnabThres>

Sets the threshold for the enable channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<EnabThres>

**Usage:** Asynchronous command

**Manual operation:** See "Enable threshold, Enable hysteresis" on page 659

---

**SBUS<sb>:MANCh:ENABle:HYSTeresis** <EnabHyst>

Sets the hysteresis for the enable channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<EnabHyst>

**Usage:** Asynchronous command

**Manual operation:** See "Enable threshold, Enable hysteresis" on page 659

---

**SBUS<sb>:MANCh:POSition** <Position>

Sets the vertical position of the Manchester signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Position>

**Usage:** Asynchronous command

---

**SBUS<sb>:MANCh:SCALe** <Scale>

Sets the vertical scale of the Manchester signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Scale>

**Usage:** Asynchronous command

### 18.18.10.2 Filter

There are two commands for each parameter, that you can use for defining the Manchester settings.

For example, to set the *Frame type =Manchester-Frame* > *Field =Data* >*Data* value you can use one of the following commands:

- `SBUS:MANCh:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:MANCh:FILTer:DMIN "Manchester-Frame", "Data", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:MANCh:FILTer:CHKall ..... 1631
SBUS<sb>:MANCh:FILTer:CLR ..... 1631
SBUS<sb>:MANCh:FILTer:INVert ..... 1631
SBUS<sb>:MANCh:FILTer:RST ..... 1632
SBUS<sb>:MANCh:FILTer:BIT ..... 1632
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:BIT ..... 1632

SBUS<sb>:MANCh:FILTer:ERENable ..... 1632
SBUS<sb>:MANCh:FILTer:ERRor<n>:ENABle ..... 1632
SBUS<sb>:MANCh:FILTer:DMAX ..... 1633
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DMAX ..... 1633
SBUS<sb>:MANCh:FILTer:DMIN ..... 1633
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DMIN ..... 1633
SBUS<sb>:MANCh:FILTer:DOPerator ..... 1633
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ..... 1633
SBUS<sb>:MANCh:FILTer:FIENable ..... 1634
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:ENABle ..... 1634
SBUS<sb>:MANCh:FILTer:FRENable ..... 1634
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:ENABle ..... 1634
SBUS<sb>:MANCh:FILTer:IMAX ..... 1634
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IMAX ..... 1634
SBUS<sb>:MANCh:FILTer:IMIN ..... 1635
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IMIN ..... 1635
SBUS<sb>:MANCh:FILTer:IOPerator ..... 1635
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ..... 1635

### SBUS<sb>:MANCh:FILTer:CHKall

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 660

### SBUS<sb>:MANCh:FILTer:CLR

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 660

### SBUS<sb>:MANCh:FILTer:INVert

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 660

---

**SBUS<sb>:MANCh:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 660

---

**SBUS<sb>:MANCh:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:MANCh:FILTer:BIT?** <Bit>
**SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 661

---

**SBUS<sb>:MANCh:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:MANCh:FILTer:ERENable?** <Enabler>
**SBUS<sb>:MANCh:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 662

**SBUS<sb>:MANCh:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:MANCh:FILTer:DMAX?** <Data>
**SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 661

**SBUS<sb>:MANCh:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:MANCh:FILTer:DMIN?** <Data>
**SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 661

**SBUS<sb>:MANCh:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:MANCh:FILTer:DOPerator?** <Operator>
**SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**

<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 661

---

**SBUS<sb>:MANCh:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:MANCh:FILTer:FIENable?** <Enabler>
**SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 661

---

**SBUS<sb>:MANCh:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:MANCh:FILTer:FRENable?** <Enabler>
**SBUS<sb>:MANCh:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables the filtering on Manchester frames. Only the frames that match the selected filter conditions are displayed.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**

<Enable> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 661

---

**SBUS<sb>:MANCh:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:MANCh:FILTer:IMAX?** <Data>
**SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

You can set the operator with `SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 661

---

**SBUS<sb>:MANCh:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:MANCh:FILTer:IMIN?** <Data>
**SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 661

---

**SBUS<sb>:MANCh:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:MANCh:FILTer:IOPerator?** <Operator>
**SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 661

### 18.18.10.3 Format

This section describes the commands to specify and query frame formats for the "Manchester" protocol.

For manual operation, refer to Section 14.9.4, "Manchester format configuration", on page 663.

SBUS<sb>:MANCh:FORMat:ADDFrame....1636
SBUS<sb>:MANCh:FORMat:LOAD....1636
SBUS<sb>:MANCh:FORMat:SAVE....1637
SBUS<sb>:MANCh:FORMat:CLR....1637
SBUS<sb>:MANCh:FORMat:FCOunt?....1637
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:ADDField....1638
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLDCount?....1638
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:NAME....1638
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:COLor....1638
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:NAME....1639
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:COLor....1639
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1639
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:CONDition....1640
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1640
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1641
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:CLMN....1641
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1642
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1642
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1643
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1643
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:PARA....1643
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:PARB....1643

---

**SBUS<sb>:MANCh:FORMat:ADDFrame**

Appendes a new frame description to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add Frame" on page 663

---

**SBUS<sb>:MANCh:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame descriptions.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 663

---

**SBUS<sb>:MANCh:FORMat:SAVE** <FileName>

Saves the current list of frame descriptions to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 664

---

**SBUS<sb>:MANCh:FORMat:CLR**

Erases from the volatile memory all field and frame format descriptions that you have created for a specific custom protocol.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit List" on page 663

---

**SBUS<sb>:MANCh:FORMat:FCOunt?**

Returns the number of frame format descriptions available for a specific custom protocol.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit List" on page 663

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame description.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add Field" on page 664

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit List" on page 665

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the frame description of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 664

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the frame description of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**

<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame Color" on page 664

---

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 665

---

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field Color" on page 665

---

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 666

---

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:CONDition** <Condition>

Specifies a user-defined condition operator for the selected field of the selected frame. The various condition operators can identify, for example, a mandatory CRC checksum value or a frame ID.

Set the numeric format of the condition by the command `SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:FORMat`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Condition> The implemented conditions are the "equal" and "array" operators.

**Usage:** Asynchronous command

**Manual operation:** See "Condition" on page 665

---

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 667

---

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 667

---

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:CLMN** <Column>

Specifies, in which result column of the decode table to display the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Column> NONE | COL1 | COL2 | COL3

**NONE**
The result is not displayed.

**COL1**
The result is displayed in column 1.

**COL2**
The result is displayed in column 2.

**COL3**
The result is displayed in column 3.

*RST: NONE

**Usage:** Asynchronous command

**Manual operation:** See "Result column" on page 667

---

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 668

---

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 668

---

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 668

---

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 668

---

**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)

- Parameter B: Offset

Available if `SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<ParameterB> Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 668

### 18.18.10.4 Software trigger

There are two commands for each parameter that you can use for defining the "Manchester" settings.

For example, to set the *Frame type =Manchester-Frame* > *Field =Data* >*Data* value you can use one of the following commands:

- `TRIGger:SBSW:MANCh:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <fr> for the frame number and <fl> for the field number.
- `TRIGger:SBSW:MANCh:DMIN "Manchester-Frame", "Data", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:MANCh:CHKall ... 1645
TRIGger:SBSW:MANCh:CLR ... 1645
TRIGger:SBSW:MANCh:INVert ... 1645
TRIGger:SBSW:MANCh:RST ... 1645
TRIGger:SBSW:MANCh:FRENable ... 1645
TRIGger:SBSW:MANCh:FRAMe<fr>:ENABle ... 1645
TRIGger:SBSW:MANCh:BIT ... 1646
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:BIT ... 1646
TRIGger:SBSW:MANCh:DMAX ... 1646
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DMAX ... 1646
TRIGger:SBSW:MANCh:DMIN ... 1646
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DMIN ... 1646
TRIGger:SBSW:MANCh:DOPerator ... 1647
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DOPerator ... 1647
TRIGger:SBSW:MANCh:FIENable ... 1647
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:ENABle ... 1647
TRIGger:SBSW:MANCh:IMAX ... 1648
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IMAX ... 1648
TRIGger:SBSW:MANCh:IMIN ... 1648
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IMIN ... 1648

TRIGger:SBSW:MANCh:IOPerator ..... 1648
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IOPerator ..... 1648
TRIGger:SBSW:MANCh:ERENable ..... 1649
TRIGger:SBSW:MANCh:ERROr<m>:ENABle ..... 1649
SBUS<sb>:MANCh:SWTindex? ..... 1649
SBUS<sb>:MANCh:SWTTime? ..... 1649

**TRIGger:SBSW:MANCh:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 669

**TRIGger:SBSW:MANCh:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 669

**TRIGger:SBSW:MANCh:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 669

**TRIGger:SBSW:MANCh:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 669

**TRIGger:SBSW:MANCh:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:MANCh:FRENable?** <Enabler>
**TRIGger:SBSW:MANCh:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 670

---

**TRIGger:SBSW:MANCh:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:MANCh:BIT?** <Bit>
**TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 670

---

**TRIGger:SBSW:MANCh:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:MANCh:DMAX?** <Data>
**TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 670

---

**TRIGger:SBSW:MANCh:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:MANCh:DMIN?** <Data>
**TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 670

---

**TRIGger:SBSW:MANCh:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:MANCh:DOPerator?** <Operator>
**TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 670

---

**TRIGger:SBSW:MANCh:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:MANCh:FIENable?** <Enabler>
**TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 670

**TRIGger:SBSW:MANCh:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:MANCh:IMAX?** <Data>
**TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max>
Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 670

**TRIGger:SBSW:MANCh:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:MANCh:IMIN?** <Data>
**TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min>
Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 670

**TRIGger:SBSW:MANCh:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:MANCh:IOPerator?** <Operator>
**TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 670

---

**TRIGger:SBSW:MANCh:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:MANCh:ERENable?** <Enabler>
**TRIGger:SBSW:MANCh:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 671

---

**SBUS<sb>:MANCh:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**

<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.10.5 Decode results

This section describes the commands to query the results of the decoding of the "Manchester" protocol.

SBUS<sb>:MANCh:FCOunt? 1650
SBUS<sb>:MANCh:FRAMe<fr>:FLDCount? 1650
SBUS<sb>:MANCh:FRAMe<fr>:WBRate? 1651
SBUS<sb>:MANCh:FRAMe<fr>:FLD<fl>:STATe? 1651
SBUS<sb>:MANCh:FRAMe<fr>:FLD<fl>:LABel? 1651
SBUS<sb>:MANCh:FRAMe<fr>:FLD<fl>:VALue? 1652
SBUS<sb>:MANCh:FRAMe<fr>:FLD<fl>:FVALue? 1652
SBUS<sb>:MANCh:FRAMe<fr>:STATus? 1652
SBUS<sb>:MANCh:FRAMe<fr>:STARt? 1653
SBUS<sb>:MANCh:FRAMe<fr>:STOP? 1653
SBUS<sb>:MANCh:FRAMe<fr>:TYPE? 1654
SBUS<sb>:MANCh:FRAMe<fr>:CONE? 1654
SBUS<sb>:MANCh:FRAMe<fr>:CTWO? 1654
SBUS<sb>:MANCh:FRAMe<fr>:CTHRee? 1654

---

**SBUS<sb>:MANCh:FCOunt?**

Returns the number of decoded frames.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Return values:**

<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:FRAMe<fr>:FLDCount?**

Returns the number of fields available in the seleced frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<DataCount>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:FRAMe<fr>:WBRate?**

Returns the bit rate of the frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameWBR> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**SBUS<sb>:MANCh:FRAMe<fr>:FLD<fl>:STATe?**

Returns the overall state of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | LENGth | CRC | PARity | INComplete

**OK**
The field has no error, it is valid.

**LENGth**
The length of the field is not as expected, indicating an error.

**CRC**
The CRC of the frame is not as expected, indicating an error.

**PARity**
Parity is not as expected, indicating an error

**INComplete**
The frame is incomplete.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:FRAMe<fr>:FLD<fl>:LABel?**

Returns the label of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Name>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:FRAMe<fr>:FLD<fl>:VALue?**

Returns the value of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> *RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:FRAMe<fr>:STATus?**

Returns the overall state of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameState> OK | LENGth | CRC | PARity | INComplete

**OK**
The frame is valid.

**LENGth**
Length of the frame is not as expected, indicating an error.

**CRC**
The checksum of the frame is not as expected, indicating an error.

**PARity**
Parity is not as expected, indicating an error.

**INComplete**
The frame is incomplete.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:FRAMe<fr>:STARt?**

Returns the start time of the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:FRAMe<fr>:STOP?**

Returns the end time of the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:FRAMe<fr>:TYPE?**

Returns the type of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameType> The frame type is undefined, because it is custom specific.

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:FRAMe<fr>:CONE?**

Returns the content of the 1st result column of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Result1>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:FRAMe<fr>:CTWO?**

Returns the content of the 2nd result column of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Result2>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MANCh:FRAMe<fr>:CTHRee?**

Returns the content of the 3rd result column of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Result3>

**Usage:** Query only
Asynchronous command

## 18.18.11 CAN (option R&S MXO3-K520)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.


- Configuration....1655
- Filter....1660
- Format....1666
- Hardware trigger....1675
- Software trigger....1685
- Decode results....1690


### 18.18.11.1 Configuration


SBUS<sb>:CAN:TYPE....1655
SBUS<sb>:CAN:BITRate....1656
SBUS<sb>:CAN:SAMPlepoint....1656
SBUS<sb>:CAN:SOURce....1656
SBUS<sb>:CAN:FDATa:DBITrate....1657
SBUS<sb>:CAN:FDATa:SAMPlepoint....1657
SBUS<sb>:CAN:SIC:HYSTeresis....1657
SBUS<sb>:CAN:SIC:THReshold....1658
SBUS<sb>:CAN:FAST:THReshold....1658
SBUS<sb>:CAN:FAST:HYSTeresis....1658
SBUS<sb>:CAN:XDATa:DBITrate....1658
SBUS<sb>:CAN:XDATa:SAMPlepoint....1659
SBUS<sb>:CAN:TRCVmode....1659
SBUS<sb>:CAN:SYMBols....1659
SBUS<sb>:CAN:NEWList....1660
SBUS<sb>:CAN:POSition....1660
SBUS<sb>:CAN:SCALe....1660


---

**SBUS<sb>:CAN:TYPE** <SignalType>

Selects the CAN-High or CAN-Low line. Both lines are required for differential signal transmission used by CAN.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**

<SignalType> CANL | CANH
*RST: CANL

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 676

---

**SBUS<sb>:CAN:BITRate** <Bitrate>

Sets the number of transmitted bits per second.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<Bitrate>
Range: 10000 to 5000000
Increment: 100
*RST: 100000
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "Nominal bit rate" on page 676

---

**SBUS<sb>:CAN:SAMPlepoint** <SamplePoint>

Sets the position of the sample point within the bit in percent of the nominal bit time.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<SamplePoint>
Range: 30 to 90
Increment: 5
*RST: 66
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Sample point" on page 676

---

**SBUS<sb>:CAN:SOURce** Source

Sets the source channel to which the line is connected.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

Source With digital channels in SIC mode:

C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8
Without digital channels in FAST mode:
C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 675

---

**SBUS<sb>:CAN:FDATa:DBITrate** <FDBitrate>

Sets the bit rate of the data phase.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<FDBitrate>
Range: 10000 to 15000000
Increment: 100
*RST: 1000000
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "FD bit rate" on page 676

---

**SBUS<sb>:CAN:FDATa:SAMPlepoint** <FDSamplePoint>

Sets the position of the sample point within the bit in percent of the nominal bit time.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<FDSamplePoint>
Range: 30 to 90
Increment: 5
*RST: 66
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "FD sample point" on page 676

---

**SBUS<sb>:CAN:SIC:HYSTeresis** <Hysteresis>

Sets a hysteresis value for the SIC transceiver mode.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 677

---

**SBUS<sb>:CAN:SIC:THReshold** <SIC Threshold>

Sets a threshold value for the SIC transceiver mode.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SIC Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 677

---

**SBUS<sb>:CAN:FAST:THReshold** <SIC Threshold>

Sets a threshold value for the fast transceiver mode.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SIC Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 677

---

**SBUS<sb>:CAN:FAST:HYSTeresis** <Hysteresis>

Sets a hysteresis value for the fast transceiver mode.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 677

---

**SBUS<sb>:CAN:XDATa:DBITrate** <XLBitrate>

Sets the bit rate of the data phase for the CAN XL frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**

<XLBitrate>
Range: 10000 to 30000000
Increment: 100
*RST: 10000000
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "XL bit rate" on page 676

---

**SBUS<sb>:CAN:XDATa:SAMPlepoint** <XLSamplePoint>

Sets the position of the sample point within the bit in percent of the nominal bit time for the CAN XL frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<XLSamplePoint>
Range: 30 to 90
Increment: 5
*RST: 66
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "XL sample point" on page 677

---

**SBUS<sb>:CAN:TRCVmode** <TransceiverMd>

Selects the transceiver mode for the CAN decoding.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<TransceiverMd> SIC | FAST
*RST: SIC

**Usage:** Asynchronous command

**Manual operation:** See "Transceiver mode" on page 676

---

**SBUS<sb>:CAN:SYMBols** <ShowSymbols>

Activates the symbol list to be used for decoding.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<ShowSymbols> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show symbols" on page 678

---

**SBUS<sb>:CAN:NEWList** <FileName>

Loads a symbol list file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName> String parameter with path and file name.

**Example:**
```
SBUS1:CAN:NEWList
'/home/storage/userData/Protocol/CAN.csv'
SBUS1:CAN:SYMBols ON
```

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Show symbols" on page 678

---

**SBUS<sb>:CAN:POSition** <CAN position>

Sets the vertical position of the CAN signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CAN position>

**Usage:** Asynchronous command

---

**SBUS<sb>:CAN:SCALe** <CAN scale>

Set the vertical scale of the indicated CAN signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CAN scale>

**Usage:** Asynchronous command

### 18.18.11.2 Filter

There are two commands for each parameter, that you can use for defining the CAN settings.

For example, to set the *Frame type =CBFF* > *Field =Data* >*Data* value you can use one of the following commands:

- `SBUS:CAN:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:CAN:FILTer:DMIN "CBFF", "Data", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:CAN:FILTer:CHKall....1661
SBUS<sb>:CAN:FILTer:CLR....1661
SBUS<sb>:CAN:FILTer:INVert....1662
SBUS<sb>:CAN:FILTer:RST....1662
SBUS<sb>:CAN:FILTer:BIT....1662
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:BIT....1662
SBUS<sb>:CAN:FILTer:DMAX....1662
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1662
SBUS<sb>:CAN:FILTer:DMIN....1663
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1663
SBUS<sb>:CAN:FILTer:DOPerator....1663
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1663
SBUS<sb>:CAN:FILTer:ERENable....1664
SBUS<sb>:CAN:FILTer:ERRor<n>:ENABle....1664
SBUS<sb>:CAN:FILTer:FIENable....1664
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1664
SBUS<sb>:CAN:FILTer:FRENable....1664
SBUS<sb>:CAN:FILTer:FRAMe<fr>:ENABle....1664
SBUS<sb>:CAN:FILTer:IMAX....1665
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1665
SBUS<sb>:CAN:FILTer:IMIN....1665
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1665
SBUS<sb>:CAN:FILTer:IOPerator....1665
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1665

### SBUS<sb>:CAN:FILTer:CHKall

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 680

### SBUS<sb>:CAN:FILTer:CLR

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 680

---

**SBUS<sb>:CAN:FILTer:INVert**

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 680

---

**SBUS<sb>:CAN:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 680

---

**SBUS<sb>:CAN:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:CAN:FILTer:BIT?** <Bit>
**SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 681

---

**SBUS<sb>:CAN:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:CAN:FILTer:DMAX?** <Data>
**SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 681

---

**SBUS<sb>:CAN:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:CAN:FILTer:DMIN?** <Data>
**SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 681

---

**SBUS<sb>:CAN:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:CAN:FILTer:DOPerator?** <Operator>
**SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 681

**SBUS<sb>:CAN:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:CAN:FILTer:ERENable?** <Enabler>
**SBUS<sb>:CAN:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 683

**SBUS<sb>:CAN:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:CAN:FILTer:FIENable?** <Enabler>
**SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 681

**SBUS<sb>:CAN:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:CAN:FILTer:FRENable?** <Enabler>
**SBUS<sb>:CAN:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables the filtering on CAN frames. Only the frames that match the selected filter conditions are displayed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 681

---

**SBUS<sb>:CAN:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:CAN:FILTer:IMAX?** <Data>
**SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max>
Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 681

---

**SBUS<sb>:CAN:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:CAN:FILTer:IMIN?** <Data>
**SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min>
Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 681

---

**SBUS<sb>:CAN:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:CAN:FILTer:IOPerator?** <Operator>
**SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe

*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 681

### 18.18.11.3 Format

This section describes the commands to specify and query frame formats for the "CAN" protocol.

SBUS<sb>:CAN:FORMat:SELect ... 1666
SBUS<sb>:CAN:FORMat:ADDFrame ... 1667
SBUS<sb>:CAN:FORMat:LOAD ... 1667
SBUS<sb>:CAN:FORMat:SAVE ... 1667
SBUS<sb>:CAN:FORMat:CLR ... 1668
SBUS<sb>:CAN:FORMat:FCOunt? ... 1668
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ADDField ... 1668
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLDCount? ... 1668
SBUS<sb>:CAN:FORMat:FRAMe<fr>:NAME ... 1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:COLor ... 1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:TYPE ... 1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID1Field ... 1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID2Field ... 1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID3Field ... 1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID1Value ... 1670
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID2Value ... 1670
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID3Value ... 1670
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:NAME ... 1670
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:COLor ... 1670
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ... 1671
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:BITCount ... 1671
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:REPetition ... 1671
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:BITorder ... 1672
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FORMat ... 1672
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ... 1673
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ... 1673
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ... 1674
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:UNIT ... 1674
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:PARA ... 1675
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:PARB ... 1675

---

**SBUS<sb>:CAN:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 683

---

**SBUS<sb>:CAN:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 683

---

**SBUS<sb>:CAN:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 684

---

**SBUS<sb>:CAN:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 684

**SBUS<sb>:CAN:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 684

**SBUS<sb>:CAN:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 684

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 685

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 685

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 685

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 685

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 685

---

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 685

---

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 686

---

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 686

---

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<FirstBit>

Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 686

---

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<BitCount>

Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 686

---

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 687

---

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 687

---

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 687

---

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 688

---

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field
<p> Index of the value

**Parameters:**
<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 688

---

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field
<p> Index of the value

**Parameters:**
<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 688

---

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 688

**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

Available if `SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

| | |
|---|---|
| <sb> | 1 to 4, index of the serial bus |
| <fr> | Index of the frame |
| <fl> | Index of the field |

**Parameters:**

| | | |
|---|---|---|
| <ParameterB> | Range: | -1E+20 to 1E+20 |
| | Increment: | 1E-12 |
| | *RST: | 0 |

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 688

### 18.18.11.4 Hardware trigger

Trigger commands for serial buses affect the current trigger source. Therefore, set the trigger source to the required serial bus with `TRIGger:EVENt<ev>:SOURce` before sending the bus-specific trigger commands.

TRIGger:SBHW:CAN:ACKerror ... 1676
TRIGger:SBHW:CAN:BITSterror ... 1676
TRIGger:SBHW:CAN:BORDer ... 1676
TRIGger:SBHW:CAN:CRCerror ... 1677
TRIGger:SBHW:CAN:DCONdition ... 1677
TRIGger:SBHW:CAN:DLC ... 1677
TRIGger:SBHW:CAN:DLCCondition ... 1677
TRIGger:SBHW:CAN:DMIN ... 1678
TRIGger:SBHW:CAN:FDATa:BRS ... 1678
TRIGger:SBHW:CAN:FDATa:DPOSition ... 1678
TRIGger:SBHW:CAN:FDATa:ESI ... 1678
TRIGger:SBHW:CAN:FDATa:SCERror ... 1679
TRIGger:SBHW:CAN:FORMerror ... 1679
TRIGger:SBHW:CAN:FTYPe ... 1679
TRIGger:SBHW:CAN:ICONdition ... 1680
TRIGger:SBHW:CAN:IMAX ... 1680
TRIGger:SBHW:CAN:IMIN ... 1680
TRIGger:SBHW:CAN:ITYPe ... 1681
TRIGger:SBHW:CAN:TYPE ... 1681
TRIGger:SBHW:CAN:XDATa:AF:CONDition ... 1682
TRIGger:SBHW:CAN:XDATa:AF:MAX ... 1682
TRIGger:SBHW:CAN:XDATa:AF:MIN ... 1682

TRIGger:SBHW:CAN:XDATa:SDT:CONDition ..... 1683
TRIGger:SBHW:CAN:XDATa:SDT:MAX ..... 1683
TRIGger:SBHW:CAN:XDATa:SDT:MIN ..... 1683
TRIGger:SBHW:CAN:XDATa:SEC ..... 1684
TRIGger:SBHW:CAN:XDATa:VCID:CONDition ..... 1684
TRIGger:SBHW:CAN:XDATa:VCID:MAX ..... 1684
TRIGger:SBHW:CAN:XDATa:VCID:MIN ..... 1684

**TRIGger:SBHW:CAN:ACKerror** <AckError>

Triggers when the transmitter does not receive an acknowledgment - a dominant bit during the ACK Slot.

Available, if `TRIGger:SBHW:CAN:TYPE` is set to `ERRC`.

**Parameters:**

<AckError> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error conditions" on page 695

**TRIGger:SBHW:CAN:BITSterror** <BitStuffError>

Triggers if a stuff error occurs - when the 6th consecutive equal bit level in the mentioned fields is detected.

Available, if `TRIGger:SBHW:CAN:TYPE` is set to `ERRC`.

**Parameters:**

<BitStuffError> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error conditions" on page 695

**TRIGger:SBHW:CAN:BORDer** <Endianness>

Sets the byte order (endianness) of the data transfer. Only for CAN protocol.

**Parameters:**

<Endianness> BENDian | LENDian

**BENDian**
Big endian, data is analyzed and evaluated in the order of reception.

**LENDian**
Little endian, the instrument reads the complete data, reverses the byte order of the data, and compares it with the specified data word.

*RST: BENDian

**Usage:** Asynchronous command

---

**TRIGger:SBHW:CAN:CRCerror** <ChecksumError>

Triggers on CRC errors. A CRC error occurs when the CRC calculated by the receiver differs from the received value in the CRC sequence.

Available, if `TRIGger:SBHW:CAN:TYPE` is set to `ERRC`.

**Parameters:**

<ChecksumError> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error conditions" on page 695

---

**TRIGger:SBHW:CAN:DCONdition** <DataOperator>

Sets the operator to set a specific data pattern or a data pattern range.

**Parameters:**

<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, not equal, less than, less or equal than, greater than, greater or equal than. These conditions require one data pattern to be set with `TRIGger:SBHW:CAN:DMIN`.

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Data pattern" on page 694

---

**TRIGger:SBHW:CAN:DLC** <DLC>

Sets the data length code, the number of data bytes to be found. For complete definition, set also the operator with `TRIGger:SBHW:CAN:DLCCondition`.

**Parameters:**

<DLC> Range: CAN: 1 to 8, CAN FD: 1 to 15 (64 bytes)
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "DLC" on page 693

---

**TRIGger:SBHW:CAN:DLCCondition** <DLCOperator>

Operator to set the data length code.

The number of data bytes to be found is set with `TRIGger:SBHW:CAN:DLC`.

**Parameters:**

<DLCOperator> EQUal | GETHan

For little endian transfer direction, EQUal must be set.

*RST: GETHan

**Usage:** Asynchronous command

**Manual operation:** See "DLC" on page 693

---

**TRIGger:SBHW:CAN:DMIN** <DataPattern>

Sets a data pattern, or sets the start value of a data pattern range.

**Parameters:**

<DataPattern> List of comma separated values

**Usage:** Asynchronous command

**Manual operation:** See "Data pattern" on page 694

---

**TRIGger:SBHW:CAN:FDATa:BRS** <BRS_Bit>

Sets the bit rate switch bit.

**Parameters:**

<BRS_Bit> ONE | ZERO | DC

ONE: the bit rate switches from the bit rate of the arbitration phase to the faster data rate.

*RST: ONE

**Usage:** Asynchronous command

**Manual operation:** See "BRS, ESI" on page 694

---

**TRIGger:SBHW:CAN:FDATa:DPOSition** <DataPosition>

Defines the number of the first data byte at which the data pattern may start.

**Parameters:**

<DataPosition> Range: 1 to 2048
Increment: 1
*RST: 1

**Usage:** Asynchronous command

---

**TRIGger:SBHW:CAN:FDATa:ESI** <ESI_Bit>

Sets the error state indicator bit.

**Parameters:**

<ESI_Bit> ONE | ZERO | DC

DC: do not care, bit is nor relevant.

*RST: DC

**Usage:** Asynchronous command

**Manual operation:** See "BRS, ESI" on page 694

---

**TRIGger:SBHW:CAN:FDATa:SCERror** <StuffCntErr>

Triggers on stuff count errors. A stuff bit error occurs if more than five consecutive bits of the same level occur on the bus.

Available, if TRIGger:SBHW:CAN:TYPE is set to ERRor.

**Parameters:**

<StuffCntErr> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error conditions" on page 695

---

**TRIGger:SBHW:CAN:FORMerror** <FormError>

Triggers when a fixed-form bit field contains one or more illegal bits.

Available, if TRIGger:SBHW:CAN:TYPE is set to ERRC.

**Parameters:**

<FormError> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error conditions" on page 695

---

**TRIGger:SBHW:CAN:FTYPe** <FrameType>

Sets the CAN frame type.

**Parameters:**

<FrameType> CBFF | CBFR | CEFF | CEFR | FBFF | FEFF | XLFF | ERRor | OVERload

CBFF: classical base frame format data
CBFR: classical base frame format remote
CEFF: classical extended frame format data
CEFR: classical extended frame format remote
FBFF: FD base frame format
FEFF: FD extended frame format
XLFF: XL frame format

ERR: error
OVLD: overload

*RST: CBFF

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 691

---

**TRIGger:SBHW:CAN:ICONdition** <IdOperator>

Sets the operator to set a specific identifier or an identifier range.

**Parameters:**

<IdOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, not equal, less than, less or equal than, greater than, greater or equal than. These conditions require one identifier pattern to be set with TRIGger:SBHW:CAN:IMIN.

**INRange | OORange**
In range / out of range: Set the minimum and maximum value of the range with TRIGger:SBHW:CAN:IMIN and TRIGger:SBHW:CAN:IMAX.

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Identifier" on page 693

---

**TRIGger:SBHW:CAN:IMAX** <IdPattern>

Sets the end value of an identifier range if TRIGger:SBHW:CAN:ICONdition is set to INRange or OORange.

**Parameters:**

<IdPattern> List of comma separated values

**Usage:** Asynchronous command

**Manual operation:** See "Identifier" on page 693

---

**TRIGger:SBHW:CAN:IMIN** <IdPattern>

Specifies a message identifier pattern, or sets the start value of an identifier range.

**Parameters:**

<IdPattern>

**Usage:** Asynchronous command

**Manual operation:** See "Identifier" on page 693

**TRIGger:SBHW:CAN:ITYPe** <IdentifierType>

Selects the format of data and remote frames.

Remote frames are not available in the CAN FD protocol.

**Parameters:**

<IdentifierType> B11 | B29

**B11**
11-bit identifier (standard format). The instrument triggers on the sample point of the IDE bit.

**B29**
29-bit identifier (extended format). The instrument triggers on the sample point of the RTR bit.

*RST: B11

**Usage:** Asynchronous command

**Manual operation:** See "ID type" on page 694

**TRIGger:SBHW:CAN:TYPE** <Type>

Selects the trigger type for CAN analysis.

**Parameters:**

<Type> STOF | EDOF | FTYP | ID | IDDT | ERRC

**STOF**
STart of Frame: triggers on the first edge of the dominant SOF bit (synchronization bit).

**FTYP**
Frame type: triggers on a specified frame type and on the identifier format.

**ID**
Identifier: Sets the trigger to one specific identifier or an identifier range.
To set the identifier, use `TRIGger:SBHW:CAN:ICONdition`, `TRIGger:SBHW:CAN:IMAX`, and `TRIGger:SBHW:CAN:IMIN`.

**IDDT**
Identifier and data: Combination of identifier and data conditions. To set the identifier condition, use `TRIGger:SBHW:CAN:ICONdition`, `TRIGger:SBHW:CAN:IMIN`, and `TRIGger:SBHW:CAN:IMAX`.
To set the data condition, use `TRIGger:SBHW:CAN:DCONdition` and `TRIGger:SBHW:CAN:DMIN`.

**ERRC**
Error condition: Define the error types with:
`TRIGger:SBHW:CAN:ACKerror`
`TRIGger:SBHW:CAN:BITSterror`
`TRIGger:SBHW:CAN:CRCerror`
`TRIGger:SBHW:CAN:FORMerror`
`TRIGger:SBHW:CAN:FDATa:SCERror`
*RST: STOF

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 689

---

**TRIGger:SBHW:CAN:XDATa:AF:CONDition** <AfOperator>

Sets the comparison condition for the acceptance field to a specific value or a range.

**Parameters:**
<AfOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, not equal, less than, less or equal than, greater than, greater or equal than. These conditions require one pattern to be set with `TRIGger:SBHW:CAN:XDATa:AF:MIN`.

**INRange | OORange**
In range / out of range: Set the minimum and maximum value of the range with `TRIGger:SBHW:CAN:XDATa:AF:MIN` and `TRIGger:SBHW:CAN:XDATa:AF:MAX`.

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "AF" on page 695

---

**TRIGger:SBHW:CAN:XDATa:AF:MAX** <AfPatternTo>

Sets the end value of an acceptance field if `TRIGger:SBHW:CAN:XDATa:AF:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<AfPatternTo> List of comma separated values

**Usage:** Asynchronous command

**Manual operation:** See "AF" on page 695

---

**TRIGger:SBHW:CAN:XDATa:AF:MIN** <AfPattern>

Specifies an acceptance field pattern, or sets the start value of a range.

**Parameters:**
<AfPattern> List of comma separated values

**Usage:** Asynchronous command

**Manual operation:** See "AF" on page 695

---

**TRIGger:SBHW:CAN:XDATa:SDT:CONDition** <SdtOperator>

Sets the comparison condition for the service data unit type to a specific value or a range.

**Parameters:**
<SdtOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, not equal, less than, less or equal than, greater than, greater or equal than. These conditions require one pattern to be set with `TRIGger:SBHW:CAN:XDATa:SDT:MIN`.

**INRange | OORange**
In range / out of range: Set the minimum and maximum value of the range with `TRIGger:SBHW:CAN:XDATa:SDT:MIN` and `TRIGger:SBHW:CAN:XDATa:SDT:MAX`.

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "SDT" on page 694

---

**TRIGger:SBHW:CAN:XDATa:SDT:MAX** <SdtPatternTo>

Sets the end value of a service data unit type range if `TRIGger:SBHW:CAN:XDATa:SDT:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<SdtPatternTo> List of comma separated values

**Usage:** Asynchronous command

**Manual operation:** See "SDT" on page 694

---

**TRIGger:SBHW:CAN:XDATa:SDT:MIN** <SdtPattern>

Specifies a service data unit type pattern, or sets the start value of a range.

**Parameters:**
<SdtPattern> List of comma separated values

**Usage:** Asynchronous command

**Manual operation:** See "SDT" on page 694

**TRIGger:SBHW:CAN:XDATa:SEC** <SEC_Bit>

Sets a value for the simple extended content (SEC) field. It indicates, if the CAN XL data frame uses the CADsec protocol.

**Parameters:**

<SEC_Bit> ONE | ZERO | DC

*RST: DC

**Usage:** Asynchronous command

**Manual operation:** See "SEC" on page 695

**TRIGger:SBHW:CAN:XDATa:VCID:CONDition** <VcidOperator>

Sets the comparison condition for the VCID to a specific value or a range.

**Parameters:**

<VcidOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, not equal, less than, less or equal than, greater than, greater or equal than. These conditions require one pattern to be set with `TRIGger:SBHW:CAN:XDATa:VCID:MIN`.

**INRange | OORange**
In range / out of range: Set the minimum and maximum value of the range with `TRIGger:SBHW:CAN:XDATa:VCID:MIN` and `TRIGger:SBHW:CAN:XDATa:VCID:MAX`.

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "VCID" on page 695

**TRIGger:SBHW:CAN:XDATa:VCID:MAX** <VcidPatternTo>

Sets the end value of a VCID range if `TRIGger:SBHW:CAN:XDATa:VCID:CONDition` is set to `INRange` or `OORange`.

**Parameters:**

<VcidPatternTo> List of comma separated values

**Usage:** Asynchronous command

**Manual operation:** See "VCID" on page 695

**TRIGger:SBHW:CAN:XDATa:VCID:MIN** <VcidPattern>

Specifies a VCID pattern, or sets the start value of a range.

**Parameters:**

<VcidPattern> List of comma separated values

**Usage:** Asynchronous command

**Manual operation:** See "VCID" on page 695

### 18.18.11.5 Software trigger

There are two commands for each parameter, that you can use for defining the CAN settings.

For example, to set the *Frame type =CBFF > Field =Data >Data* value you can use one of the following commands:

- `TRIGger:SBSW:CAN:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:SBSW:CAN:DMIN "CBFF", "Data", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:CAN:CHKall....1685
TRIGger:SBSW:CAN:CLR....1686
TRIGger:SBSW:CAN:INVert....1686
TRIGger:SBSW:CAN:RST....1686
TRIGger:SBSW:CAN:FRENable....1686
TRIGger:SBSW:CAN:FRAMe<fr>:ENABle....1686
TRIGger:SBSW:CAN:BIT....1686
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:BIT....1686
TRIGger:SBSW:CAN:DMAX....1687
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DMAX....1687
TRIGger:SBSW:CAN:DMIN....1687
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DMIN....1687
TRIGger:SBSW:CAN:DOPerator....1688
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DOPerator....1688
TRIGger:SBSW:CAN:FIENable....1688
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:ENABle....1688
TRIGger:SBSW:CAN:IMAX....1688
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IMAX....1688
TRIGger:SBSW:CAN:IMIN....1689
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IMIN....1689
TRIGger:SBSW:CAN:IOPerator....1689
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IOPerator....1689
TRIGger:SBSW:CAN:ERENable....1689
TRIGger:SBSW:CAN:ERRor<m>:ENABle....1689
SBUS<sb>:CAN:SWTindex?....1690
SBUS<sb>:CAN:SWTTime?....1690

---

**TRIGger:SBSW:CAN:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 698

---

**TRIGger:SBSW:CAN:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 698

---

**TRIGger:SBSW:CAN:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 698

---

**TRIGger:SBSW:CAN:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 698

---

**TRIGger:SBSW:CAN:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:CAN:FRENable?** <Enabler>
**TRIGger:SBSW:CAN:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 698

---

**TRIGger:SBSW:CAN:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:CAN:BIT?** <Bit>
**TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 698

---

**TRIGger:SBSW:CAN:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:CAN:DMAX?** <Data>
**TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 698

---

**TRIGger:SBSW:CAN:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:CAN:DMIN?** <Data>
**TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 698

**TRIGger:SBSW:CAN:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:CAN:DOPerator?** <Operator>
**TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 698

**TRIGger:SBSW:CAN:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:CAN:FIENable?** <Enabler>
**TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 698

**TRIGger:SBSW:CAN:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:CAN:IMAX?** <Data>
**TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 698

---

**TRIGger:SBSW:CAN:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:CAN:IMIN?** <Data>
**TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 698

---

**TRIGger:SBSW:CAN:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:CAN:IOPerator?** <Operator>
**TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 698

---

**TRIGger:SBSW:CAN:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:CAN:ERENable?** <Enabler>
**TRIGger:SBSW:CAN:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 700

---

**SBUS<sb>:CAN:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.11.6 Decode results

SBUS<sb>:CAN:FCOunt? 1691
SBUS<sb>:CAN:FRAMe<fr>:FLDCount? 1691
SBUS<sb>:CAN:FRAMe<fr>:ACKState? 1692
SBUS<sb>:CAN:FRAMe<fr>:ACKValue? 1692
SBUS<sb>:CAN:FRAMe<fr>:FLD<fl>:STATe? 1692
SBUS<sb>:CAN:FRAMe<fr>:FLD<fl>:LABel? 1693

SBUS<sb>:CAN:FRAMe<fr>:FLD<fl>:VALue? 1693
SBUS<sb>:CAN:FRAMe<fr>:FLD<fl>:FVALue? 1694
SBUS<sb>:CAN:FRAMe<fr>:CSSTate? 1694
SBUS<sb>:CAN:FRAMe<fr>:CSValue? 1694
SBUS<sb>:CAN:FRAMe<fr>:DATA? 1695
SBUS<sb>:CAN:FRAMe<fr>:DLCState? 1695
SBUS<sb>:CAN:FRAMe<fr>:DLCValue? 1695
SBUS<sb>:CAN:FRAMe<fr>:IDSTate? 1696
SBUS<sb>:CAN:FRAMe<fr>:IDTYpe? 1696
SBUS<sb>:CAN:FRAMe<fr>:IDValue? 1696
SBUS<sb>:CAN:FRAMe<fr>:NDBYtes? 1697
SBUS<sb>:CAN:FRAMe<fr>:STARt? 1697
SBUS<sb>:CAN:FRAMe<fr>:STATus? 1697
SBUS<sb>:CAN:FRAMe<fr>:STOP? 1698
SBUS<sb>:CAN:FRAMe<fr>:SYMBol? 1698
SBUS<sb>:CAN:FRAMe<fr>:TYPE? 1699
SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:AF? 1699
SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:PCRC? 1699
SBUS<sb>:CAN:FRAMe<fr>:SBC? 1700
SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:SDT? 1700
SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:SEC? 1700
SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:VCID? 1701
SBUS<sb>:CAN:FRAMe<fr>:FDATa<o>:ESI? 1701
SBUS<sb>:CAN:FRAMe<fr>:FDATa<o>:BRS? 1702
SBUS<sb>:CAN:FRAMe<fr>:DBITrate? 1702
SBUS<sb>:CAN:FRAMe<fr>:NBITrate? 1702
SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:FCRC? 1703

### SBUS<sb>:CAN:FCOunt?

Returns the number of decoded frames for the CAN protocol analysis.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count> Total number of frames.

**Usage:** Query only
Asynchronous command

### SBUS<sb>:CAN:FRAMe<fr>:FLDCount?

Returns the number of fields in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:ACKState?**

Return the states of the acknowledgment field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameAckState> OK | ERRor | UNDF

UNDF: Undefined

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:ACKValue?**

Returns the value of the acknowledge slot for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameAckValue> Range: 0 to 1
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:FLD<fl>:STATe?**

Returns the state of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> *
Selects the field number.

**Return values:**
<State> OK | EOFD | ACKD | NOACk | CRCD | CRC | SERRror | FORM | BTST | UNKNown | INComplete

**OK**
The field has no error, it is valid.

**EOFD**
Wrong end of field

**ACKD**
Acknowledgment delimiter error

**NOACk**
Field acknowledgment is missing

**CRCD**
Checksum delimiter error

**CRC**
Checksum error, cyclic redundancy check failed

**SERRror**
Stuff count error (CAN-FD ISO only)

**FORM**
Fixed-bit form error

**BTST**
Bit stuffing error

**UNKNown**
State unknown

**INComplete**
Field not completely contained in the acquisition.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:FLD<fl>:LABel?**

Returns the label of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:FLD<fl>:VALue?**

Returns the value of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> *
Index of the field

**Return values:**
<Value> Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:CSSTate?**

Return the states of the checksum field (CRC).

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<ChecksumState> OK | ERRor | UNDF
*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:CSValue?**

Returns the CRC sequence value of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<ChecksumValue> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:DATA?**

Returns the data of the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<Count> Comma-separated list of values. The first value is the number of bytes, followed by the values of the data bytes.

<Values>

**Example:**
```
BUS1:CAN:FRAMe2:DATA?
--> 3,208,231,32
```
Returns the data of the 2nd frame: the number of data bytes is 3 (see 1st value).

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:DLCState?**

Return the states of the state of data length code.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameDLCState> OK | ERRor | UNDF
*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:DLCValue?**

Returns the data length code of the selected frame - the number of data bytes in the frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameDLCValue> Range: 0 to 2047
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:IDSTate?**

Return the states of the identifier state.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<IdtfSt> OK | ERRor | UNDF
UNDF: Undefined
*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:IDTYpe?**

Returns the identifier type of the selected frame, the identifier format of data and remote frames.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<IdentifierType> B11 | B29
B11: standard format, 11 bit
B29: extended format, 29 bit
*RST: B11

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:IDValue?**

Returns the identifier value of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<IdtfVal> Range: 0 to 536870911
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:NDBYtes?**

Returns the number of data bytes.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<NDBytes> Range: 0 to 2048
Increment: 1
*RST: 1

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:STARt?**

Return the start time of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:STATus?**

Returns the overall state of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameState> OK | EOFD | ACKD | NOACk | CRCD | CRC | SERRror | FORM | BTST | UNKNown | INComplete

`OK`: the frame is valid.
`EOFD`: Wrong end of frame.
`ACKD`: Wrong ACK delimiter occurred.
`NOACK`: Acknowledge is missing.
`CRCD`: Wrong CRC delimiter occurred.
`CRC`: Cyclic redundancy check failed.
`SERRror`: Stuff count error (CAN FD ISO only).
`FORM`: Fixed-bit form error.
`BTST`: Bit stuffing error occurred.
`INSufficient`: The frame is not completely contained in the acquisition. The acquired part of the frame is valid.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:STOP?**

Return the stop time of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:SYMBol?**

Returns the symbol of the specified frame if the label list is enabled.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<Label> String with symbol of the identifier

**Example:** `BUS:CAN:FRAMe:SYMBol?`
Response: `Temperature`

**Usage:** Query only

---

**SBUS<sb>:CAN:FRAMe<fr>:TYPE?**

Returns the frame type of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameType> CBFF | CBFR | CEFF | CEFR | FBFF | FEFF | XLFF | ERRor | OVERload | UNDefined

`CBFF`: classical base frame format data
`CBFRemote`: classical base frame format remote
`CEFF`: classical extended frame format data
`CEFRemote`: classical extended frame format remote
`FBFF`: FD base frame format
`FEFF`: FD extended frame format
`XLFF`: XL frame format
`ERR`: error
`OVLD`: overload

*RST: CBFF

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:AF?**

Returns the value of the acceptance field for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<o> Index of the field

**Return values:**
<AF> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:PCRC?**

Returns the value of the preamble cyclic redundant check (PCRC) for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<o> Index of the field

**Return values:**
<PCRC> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:SBC?**

Returns the value of the stuff bit count (SBC) field for the selected CAN XL frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<SBC> Range: 0 to 15
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:SDT?**

Returns the value of the service data unit field for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<o> Index of the field

**Return values:**
<SDT> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:SEC?**

Returns the value of the simple extended content (SEC) field for the selected CAN XL frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<o> Index of the field

**Return values:**
<SEC> Range: 0 to 1
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:VCID?**

Returns the value of the VCID field for the selected CAN XL frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<o> Index of the field

**Return values:**
<VCID> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:CAN:FRAMe<fr>:FDATa<o>:ESI?**

Returns the value of the error state indicator (ESI) field for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<o> Index of the field

**Return values:**
<ESI> Range: 0 to 1
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:CAN:FRAMe<fr>:FDATa<o>:BRS?**

Returns the value of the bit rate switch (BRS) field for the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<o> Index of the field

**Return values:**

<BRS> Range: 0 to 1
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:CAN:FRAMe<fr>:DBITrate?**

Returns the data bit rate of the frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<DataBitRate> Increment: 100
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

**SBUS<sb>:CAN:FRAMe<fr>:NBITrate?**

Returns the nominal bit rate of the frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<NominalBitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

**SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:FCRC?**

Returns the value of the frame CRC for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<o> Index of the field

**Return values:**
<FCRC> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

## 18.18.12 LIN (option R&S MXO3-K520)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.

- Configuration....1703
- Filter....1706
- Format....1712
- Hardware trigger....1721
- Software trigger....1724
- Decode results....1730

### 18.18.12.1 Configuration

SBUS<sb>:LIN:BITRate....1703
SBUS<sb>:LIN:DATA:HYSTeresis....1704
SBUS<sb>:LIN:DATA:SOURce....1704
SBUS<sb>:LIN:DATA:THReshold....1704
SBUS<sb>:LIN:POLarity....1705
SBUS<sb>:LIN:STANdard....1705
SBUS<sb>:LIN:POSition....1705
SBUS<sb>:LIN:SCALe....1705
SBUS<sb>:LIN:NEWList....1706
SBUS<sb>:LIN:SYMBols....1706

**SBUS<sb>:LIN:BITRate** <Bitrate>

Sets the number of transmitted bits per second. The maximum bit rate for LIN is 20 kbit/s.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Bitrate> Range: 1000 to 20000000
Increment: 1
*RST: 9600
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "Bit rate" on page 706

---

**SBUS<sb>:LIN:DATA:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the data channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 707

---

**SBUS<sb>:LIN:DATA:SOURce** <DataSource>

Sets the waveform of the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DataSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 706

---

**SBUS<sb>:LIN:DATA:THReshold** <Threshold>

Sets a user-defined threshold value for the data channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 707

---

**SBUS<sb>:LIN:POLarity** <Polarity>

Defines the idle state of the bus. The idle state is the recessive state and corresponds to a logic 1.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Polarity> IDLLow | IDLHigh
*RST: IDLHigh

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 706

---

**SBUS<sb>:LIN:STANdard** <Standard>

Selects the version of the LIN standard.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Standard> V1X | V2X | J2602 | AUTO
*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "LIN standard" on page 706

---

**SBUS<sb>:LIN:POSition** <LIN position>

Sets the vertical position of the LIN signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<LIN position>

**Usage:** Asynchronous command

---

**SBUS<sb>:LIN:SCALe** <LIN scale>

Set the vertical scale of the indicated LIN signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<LIN scale>

**Usage:** Asynchronous command

---

**SBUS<sb>:LIN:NEWList** <FileName>

Loads a symbol list file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName> String parameter with path and file name.

**Example:**
```
SBUS1:LIN:NEWList
'/home/storage/userData/Protocol/LIN.csv'
SBUS1:LIN:SYMBols ON
```

**Usage:** Setting only
Asynchronous command

---

**SBUS<sb>:LIN:SYMBols** <ShowSymbols>

Activates the symbol list to be used for decoding.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ShowSymbols> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

### 18.18.12.2 Filter

There are two commands for each parameter, that you can use for defining the LIN settings.

For example, to set the *Frame type =Data > Field =Id >00* value you can use one of the following commands:

- `SBUS:LIN:FILTer:FRAMe1:FLD1:DMIN 00`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:LIN:FILTer:DMIN "Data", "Id", 00`
  Defines the parameter by using the frame and field name.

SBUS<sb>:LIN:FILTer:CHKall ..... 1707
SBUS<sb>:LIN:FILTer:CLR ..... 1707
SBUS<sb>:LIN:FILTer:INVert ..... 1707
SBUS<sb>:LIN:FILTer:RST ..... 1708
SBUS<sb>:LIN:FILTer:BIT ..... 1708
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:BIT ..... 1708
SBUS<sb>:LIN:FILTer:DMAX ..... 1708

SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1708
SBUS<sb>:LIN:FILTer:DMIN ... 1709
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1709
SBUS<sb>:LIN:FILTer:DOPerator ... 1709
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1709
SBUS<sb>:LIN:FILTer:ERENable ... 1709
SBUS<sb>:LIN:FILTer:ERRor<n>:ENABle ... 1709
SBUS<sb>:LIN:FILTer:FIENable ... 1710
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1710
SBUS<sb>:LIN:FILTer:FRENable ... 1710
SBUS<sb>:LIN:FILTer:FRAMe<fr>:ENABle ... 1710
SBUS<sb>:LIN:FILTer:IMAX ... 1710
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1710
SBUS<sb>:LIN:FILTer:IMIN ... 1711
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1711
SBUS<sb>:LIN:FILTer:IOPerator ... 1711
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1711

### SBUS<sb>:LIN:FILTer:CHKall

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 708

### SBUS<sb>:LIN:FILTer:CLR

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 708

### SBUS<sb>:LIN:FILTer:INVert

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 708

---

**SBUS<sb>:LIN:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 708

---

**SBUS<sb>:LIN:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:LIN:FILTer:BIT?** <Bit>
**SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 709

---

**SBUS<sb>:LIN:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:LIN:FILTer:DMAX?** <Data>
**SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 709

**SBUS<sb>:LIN:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:LIN:FILTer:DMIN?** <Data>
**SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 709

**SBUS<sb>:LIN:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:LIN:FILTer:DOPerator?** <Operator>
**SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 709

**SBUS<sb>:LIN:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:LIN:FILTer:ERENable?** <Enabler>
**SBUS<sb>:LIN:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> index of the error

**Parameters:**
<Enable> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 710

---

**SBUS<sb>:LIN:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:LIN:FILTer:FIENable?** <Enabler>
**SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 709

---

**SBUS<sb>:LIN:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:LIN:FILTer:FRENable?** <Enabler>
**SBUS<sb>:LIN:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables the filtering on LIN frames. Only the frames that match the selected filter conditions are displayed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 709

---

**SBUS<sb>:LIN:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:LIN:FILTer:IMAX?** <Data>
**SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 709

---

**SBUS<sb>:LIN:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:LIN:FILTer:IMIN?** <Data>
**SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 709

---

**SBUS<sb>:LIN:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:LIN:FILTer:IOPerator?** <Operator>
**SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 709

### 18.18.12.3 Format

This section describes the commands to specify and query frame formats for the "LIN" protocol.

SBUS<sb>:LIN:FORMat:SELect....1712
SBUS<sb>:LIN:FORMat:ADDFrame....1713
SBUS<sb>:LIN:FORMat:LOAD....1713
SBUS<sb>:LIN:FORMat:SAVE....1713
SBUS<sb>:LIN:FORMat:CLR....1713
SBUS<sb>:LIN:FORMat:FCOunt?....1714
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ADDField....1714
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLDCount?....1714
SBUS<sb>:LIN:FORMat:FRAMe<fr>:NAME....1714
SBUS<sb>:LIN:FORMat:FRAMe<fr>:COLor....1715
SBUS<sb>:LIN:FORMat:FRAMe<fr>:TYPE....1715
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID1Field....1715
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID2Field....1715
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID3Field....1715
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID1Value....1716
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID2Value....1716
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID3Value....1716
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:NAME....1716
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:COLor....1716
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1717
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1717
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1717
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1718
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1718
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1719
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1719
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1720
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1720
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:PARA....1720
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:PARB....1720

---

**SBUS<sb>:LIN:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 710

**SBUS<sb>:LIN:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 711

**SBUS<sb>:LIN:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 711

**SBUS<sb>:LIN:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 712

**SBUS<sb>:LIN:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 711

**SBUS<sb>:LIN:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 711

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 712

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 712

---

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 712

---

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 712

---

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 712

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 712

---

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 713

---

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 713

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<FirstBit> Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 713

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 713

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 714

---

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 714

---

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 714

---

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 715

---

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 715

---

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 715

---

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 715

---

**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)

- Parameter B: Offset

Available if `SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<ParameterB> Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 715

### 18.18.12.4 Hardware trigger

Trigger commands for serial buses affect the current trigger source. Therefore, set the trigger source to the required serial bus with `TRIGger:EVENt<ev>:SOURce` before sending the bus-specific trigger commands.

TRIGger:SBHW:LIN:TYPE....1721
TRIGger:SBHW:LIN:SYERror....1722
TRIGger:SBHW:LIN:IPERror....1722
TRIGger:SBHW:LIN:IMIN....1722
TRIGger:SBHW:LIN:IMAX....1723
TRIGger:SBHW:LIN:ICONdition....1723
TRIGger:SBHW:LIN:DPOSition....1723
TRIGger:SBHW:LIN:DMIN....1723
TRIGger:SBHW:LIN:DCONdition....1724
TRIGger:SBHW:LIN:CHKSerror....1724

---

**TRIGger:SBHW:LIN:TYPE** <Type>

Selects the trigger type for LIN analysis.

**Parameters:**
<Type> STARtframe | ID | IDDT | WKFR | ERRC

**STARtframe**
Start of the frame. Triggers on the stop bit of the sync field.

**ID**
Sets the trigger to one specific identifier or an identifier range.

**IDDT**
Combination of identifier and data conditions.

**WKFR**
Wake-up frame.

**ERRC**
Error condition. Define the error types with:
`TRIGger:SBHW:LIN:CHKSerror`
`TRIGger:SBHW:LIN:IPERror`
`TRIGger:SBHW:LIN:SYERror`
*RST: STARtframe

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 716

---

**TRIGger:SBHW:LIN:SYERror** <SyncError>

Triggers if a synchronization error occurs.

Available, if `TRIGger:SBHW:LIN:TYPE` is set to `ERRC`.

**Parameters:**
<SyncError> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error conditions" on page 717

---

**TRIGger:SBHW:LIN:IPERror** <IdParityError>

Triggers if an error occurs in the identifier parity bits. The parity bits are the bits 6 and 7 of the identifier.

Available, if `TRIGger:SBHW:LIN:TYPE` is set to `ERRC`.

**Parameters:**
<IdParityError> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error conditions" on page 717

---

**TRIGger:SBHW:LIN:IMIN** <IdPattern>

Specifies a secondary identifier pattern, or sets the start value of an identifier range.

**Parameters:**
<IdPattern>

**Usage:** Asynchronous command

**Manual operation:** See "Identifier setup: Condition, Frame ID min, Frame ID max" on page 717

**TRIGger:SBHW:LIN:IMAX** <IdPattern>

Sets the end value of an identifier range if `TRIGger:SBHW:LIN:ICONdition` is set to `INRange` or `OORange`.

**Parameters:**
<IdPattern>

**Usage:** Asynchronous command

**Manual operation:** See "Identifier setup: Condition, Frame ID min, Frame ID max" on page 717

**TRIGger:SBHW:LIN:ICONdition** <IdOperator>

Sets the operator to set a specific identifier or an identifier range.

**Parameters:**
<IdOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, not equal, less than, less or equal than, greater than, greater or equal than. These conditions require one identifier pattern to be set with `TRIGger:SBHW:LIN:IMIN`.

**INRange | OORange**
In range / out of range: Set the minimum and maximum value of the range with `TRIGger:SBHW:LIN:IMIN` and `TRIGger:SBHW:LIN:IMAX`.

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Identifier setup: Condition, Frame ID min, Frame ID max" on page 717

**TRIGger:SBHW:LIN:DPOSition** <DataPosition>

Sets the number of data events that are ignored, before trigger condition check of the data starts.

**Parameters:**
<DataPosition> Range: 1 to 4096
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Data setup: Condition, Pattern, Position" on page 717

**TRIGger:SBHW:LIN:DMIN** <DataPattern>

Specifies a data pattern, or sets the start value of a data pattern range.

**Parameters:**
<DataPattern>

**Usage:** Asynchronous command

**Manual operation:** See "Data setup: Condition, Pattern, Position" on page 717

---

**TRIGger:SBHW:LIN:DCONdition** <DataOperator>

Sets the operator to set a specific data pattern or a data pattern range.

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan
*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Data setup: Condition, Pattern, Position" on page 717

---

**TRIGger:SBHW:LIN:CHKSerror** <ChecksumError>

Triggers on checksum errors.

Available, if `TRIGger:SBHW:LIN:TYPE` is set to `ERRC`.

**Parameters:**
<ChecksumError> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error conditions" on page 717

### 18.18.12.5 Software trigger

There are two commands for each parameter, that you can use for defining the LIN settings.

For example, to set the *Frame type =Data > Field =Id >00* value you can use one of the following commands:

- `TRIGger:SBSW:LIN:FRAMe1:FLD1:DMIN 00`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:SBSW:LIN:DMIN "Data", "Id", 00`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:LIN:CHKall....1725
TRIGger:SBSW:LIN:CLR....1725
TRIGger:SBSW:LIN:INVert....1725
TRIGger:SBSW:LIN:RST....1725
TRIGger:SBSW:LIN:FRENable....1726
TRIGger:SBSW:LIN:FRAMe<fr>:ENABle....1726
TRIGger:SBSW:LIN:BIT....1726

TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:BIT....1726
TRIGger:SBSW:LIN:DMAX....1726
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DMAX....1726
TRIGger:SBSW:LIN:DMIN....1727
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DMIN....1727
TRIGger:SBSW:LIN:DOPerator....1727
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DOPerator....1727
TRIGger:SBSW:LIN:FIENable....1727
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:ENABle....1727
TRIGger:SBSW:LIN:IMAX....1728
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IMAX....1728
TRIGger:SBSW:LIN:IMIN....1728
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IMIN....1728
TRIGger:SBSW:LIN:IOPerator....1729
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IOPerator....1729
TRIGger:SBSW:LIN:ERENable....1729
TRIGger:SBSW:LIN:ERRor<m>:ENABle....1729
SBUS<sb>:LIN:SWTindex?....1729
SBUS<sb>:LIN:SWTTime?....1730

### TRIGger:SBSW:LIN:CHKall

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 719

### TRIGger:SBSW:LIN:CLR

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 719

### TRIGger:SBSW:LIN:INVert

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 719

### TRIGger:SBSW:LIN:RST

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 719

---

**TRIGger:SBSW:LIN:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:LIN:FRENable?** <Enabler>
**TRIGger:SBSW:LIN:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 719

---

**TRIGger:SBSW:LIN:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:LIN:BIT?** <Bit>
**TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 720

---

**TRIGger:SBSW:LIN:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:LIN:DMAX?** <Data>
**TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 720

---

**TRIGger:SBSW:LIN:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:LIN:DMIN?** <Data>
**TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 720

---

**TRIGger:SBSW:LIN:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:LIN:DOPerator?** <Operator>
**TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 720

---

**TRIGger:SBSW:LIN:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:LIN:FIENable?** <Enabler>
**TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 720

---

**TRIGger:SBSW:LIN:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:LIN:IMAX?** <Data>
**TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 720

---

**TRIGger:SBSW:LIN:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:LIN:IMIN?** <Data>
**TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 720

**TRIGger:SBSW:LIN:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:LIN:IOPerator?** <Operator>
**TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 720

**TRIGger:SBSW:LIN:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:LIN:ERENable?** <Enabler>
**TRIGger:SBSW:LIN:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 721

**SBUS<sb>:LIN:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:LIN:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.12.6 Decode results

The following section describes the commands used for querying the results of the LIN decoding.

SBUS<sb>:LIN:FCOunt? 1730
SBUS<sb>:LIN:FRAMe<fr>:FLDCount? 1731
SBUS<sb>:LIN:FRAMe<fr>:BITRate? 1731
SBUS<sb>:LIN:FRAMe<fr>:FLD<fl>:STATe? 1731
SBUS<sb>:LIN:FRAMe<fr>:FLD<fl>:VALue? 1732
SBUS<sb>:LIN:FRAMe<fr>:FLD<fl>:FVALue? 1732
SBUS<sb>:LIN:FRAMe<fr>:FLD<fl>:LABel? 1732
SBUS<sb>:LIN:FRAMe<fr>:CSValue? 1733
SBUS<sb>:LIN:FRAMe<fr>:DATA? 1733
SBUS<sb>:LIN:FRAMe<fr>:IDPValue? 1733
SBUS<sb>:LIN:FRAMe<fr>:IDValue? 1734
SBUS<sb>:LIN:FRAMe<fr>:STARt? 1734
SBUS<sb>:LIN:FRAMe<fr>:STATus? 1734
SBUS<sb>:LIN:FRAMe<fr>:STOP? 1735
SBUS<sb>:LIN:FRAMe<fr>:SYMBol? 1735
SBUS<sb>:LIN:FRAMe<fr>:PIDentifier? 1735

**SBUS<sb>:LIN:FCOunt?**

Returns the number of decoded frames.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count> Total number of decoded frames.

**Usage:** Query only
Asynchronous command

**SBUS<sb>:LIN:FRAMe<fr>:FLDCount?**

Returns the number of fields in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count> Total number of fields.

**Usage:** Query only
Asynchronous command

**SBUS<sb>:LIN:FRAMe<fr>:BITRate?**

Returns the primary bit rate.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<PrimaryBitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

**SBUS<sb>:LIN:FRAMe<fr>:FLD<fl>:STATe?**

Returns the state of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | WAKeup | CHCKsum | PRERror | STERror | SYERror | LNERror | UNK | INComplete

**OK**
The field has no error, it is valid.

**WAKeup**
Wakeup bit missing

**CHCKsum**
Checksum error

**PRERror**
Parity error, incorrect parity bit

**STERror**
Start error, incorrect start bit

**SYERror**
Synchronization error, incorrect synchronization bit

**LNERror**
Length error

**UNKNown**
State unknown

**INComplete**
Field not completely contained in the acquisition.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:LIN:FRAMe<fr>:FLD<fl>:VALue?**

Returns the value of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:LIN:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:LIN:FRAMe<fr>:FLD<fl>:LABel?**

Returns the label of the fields in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:LIN:FRAMe<fr>:CSValue?**

Returns the checksum value of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<ChecksumValue> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:LIN:FRAMe<fr>:DATA?**

Returns the data bytes of the specified frame in comma-separated values.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

<Values>

**Example:**
```
BUS:LIN:FRAMe4:DATA?
<-- 4,118,39,71,123
```

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:LIN:FRAMe<fr>:IDPValue?**

Returns the value of the identifier parity bits of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<IdtfParVal> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:LIN:FRAMe<fr>:IDValue?**

Returns the identifier value of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<IdtfVal> Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:LIN:FRAMe<fr>:STARt?**

Returns the start time of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:LIN:FRAMe<fr>:STATus?**

Returns the overall state of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameState> OK | WAKeup | CHCKsum | PRERror | STERror | SYERror | LNERror | UNK | INComplete

`WAKeup`: wake-up frame.
`CHCKsum`: checksum error
`PRERror`: parity error in identifier
`STERror`: stop error
`SYERror`: synchronization error
`LNERror`: unexpected length
`UNK`: unknown error

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:LIN:FRAMe<fr>:STOP?**

Returns the stop time of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:LIN:FRAMe<fr>:SYMBol?**

Returns the symbolic label of the specified frame if the label list is enabled.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<Symbol>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:LIN:FRAMe<fr>:PIDentifier?**

Returns the protected identifier of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Protected identifer> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

## 18.18.13 SENT (option R&S MXO3-K520)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.


- Configuration....1736
- Filter....1742
- Format....1748
- Software trigger....1757
- Decode results....1763


### 18.18.13.1 Configuration


SBUS<sb>:SENT:CLKPeriod....1736
SBUS<sb>:SENT:CLKTolerance....1737
SBUS<sb>:SENT:CRCMethod....1737
SBUS<sb>:SENT:CRCVersion....1737
SBUS<sb>:SENT:DATA:HYSTeresis....1738
SBUS<sb>:SENT:DATA:SOURce....1738
SBUS<sb>:SENT:DATA:THReshold....1738
SBUS<sb>:SENT:DNIBbles....1739
SBUS<sb>:SENT:MODE....1739
SBUS<sb>:SENT:NEWList....1739
SBUS<sb>:SENT:POSition....1740
SBUS<sb>:SENT:PPFLength....1740
SBUS<sb>:SENT:PPULse....1740
SBUS<sb>:SENT:RDSL....1741
SBUS<sb>:SENT:SCALe....1741
SBUS<sb>:SENT:SFORmat....1741
SBUS<sb>:SENT:SWTindex?....1741
SBUS<sb>:SENT:SWTTime?....1742
SBUS<sb>:SENT:SYMBols....1742


**SBUS<sb>:SENT:CLKPeriod** <ClockPeriod>

Sets the transmitter-specific nominal clock period (clock tick).

The clock period and signal length determine the speed of transmission.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ClockPeriod> Range: 1E-06 to 0.0001
Increment: 1E-06
*RST: 6E-06
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Clock period" on page 729

---

**SBUS<sb>:SENT:CLKTolerance** <ClockTolerance>

Specifies a tolerated deviation of the clock.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ClockTolerance> Range: 1 to 25
Increment: 1
*RST: 20
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Clock tolerance" on page 729

---

**SBUS<sb>:SENT:CRCMethod** <CRCCalculation>

Selects the method for CRC calculation.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CRCCalculation> SAEJ | TLE
`SAEJ`: SAE_J2716
`TLE`: TLE_4998X
*RST: SAEJ

**Usage:** Asynchronous command

**Manual operation:** See "CRC calculation" on page 730

---

**SBUS<sb>:SENT:CRCVersion** <CRCVersion>

Selects the version the CRC check is based on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<CRCVersion> LEGA | V2010
LEGA: Legacy
V2010: v2010/v2016
*RST: V2010

**Usage:** Asynchronous command

**Manual operation:** See "CRC version" on page 730

---

**SBUS<sb>:SENT:DATA:HYSTeresis** <SDATA Hyst>

Sets the hysteresis for the data channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SDATA Hyst>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 730

---

**SBUS<sb>:SENT:DATA:SOURce** <DataSource>

Sets the source of the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DataSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 729

---

**SBUS<sb>:SENT:DATA:THReshold** <SDATA Thres>

Sets the threshold for the data channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SDATA Thres>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 730

**SBUS<sb>:SENT:DNIBbles** <DataNibbles>

Sets the number of data units in a single transmission sequence.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DataNibbles> Range: 1 to 6
Increment: 1
*RST: 6

**Usage:** Asynchronous command

**Manual operation:** See "Data nibbles" on page 729

**SBUS<sb>:SENT:MODE** <Mode>

Selects the operating mode of the SENT protocol.

If `Legacy` is selected, the SPC mode is not supported.

If `SPC` mode is selected, the SENT protocol allows for the calibration of a sensor's output by sending a single calibrated value.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Mode> LEGacy | SPC
*RST: LEGacy

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 729

**SBUS<sb>:SENT:NEWList** <FileName>

Loads a symbol list file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName> String parameter with path and file name.

**Example:**
```
SBUS1:SENT:NEWList
'/home/storage/userData/Protocol/SENT.csv'
SBUS1:SENT:SYMBols ON
```

**Usage:** Setting only
Asynchronous command

**SBUS<sb>:SENT:POSition** <SENT position>

Sets the vertical position of the SENT signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SENT position>

**Usage:** Asynchronous command

---

**SBUS<sb>:SENT:PPFLength** <FrameLength>

Specifies the fixed frame length in terms of ticks, which requires setting the pause pulse (`SBUS<sb>:SENT:PPULse`) to `PPFL`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<FrameLength> Range: 104 to 922
Increment: 1
*RST: 256

**Usage:** Asynchronous command

**Manual operation:** See "Frame length" on page 730

---

**SBUS<sb>:SENT:PPULse** <PausePulse>

Determines whether a pause pulse is transmitted after the checksum nibble.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<PausePulse> NPP | PP | PPFL

**NPP**
No pause pulse is transmitted.

**PP**
Enables transmitting a pause pulse.

**PPFL**
A pause pulse is transmitted to achieve a fixed frame length, which is specified by `SBUS<sb>:SENT:PPFLength`.

*RST: NPP

**Usage:** Asynchronous command

**Manual operation:** See "Pause pulse" on page 730

**SBUS<sb>:SENT:RDSL** <RessDispSel>

Selects the results to be displayed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<RessDispSel> TRSQ | SMSG | ALL
`TRSQ`: transmission sequence
`SMSG`: serial messages
*RST: ALL

**Usage:** Asynchronous command

**SBUS<sb>:SENT:SCALe** <SENT scale>

Set the vertical scale of the indicated SENT signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SENT scale>

**Usage:** Asynchronous command

**SBUS<sb>:SENT:SFORmat** <SerialMessages>

Selects if serial messages are enabled or disabled.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SerialMessages> ENABled | DISabled
*RST: DISabled

**Usage:** Asynchronous command

**Manual operation:** See "Serial message" on page 729

**SBUS<sb>:SENT:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**

<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Return values:**

<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:SYMBols** <ShowSymbols>

Activates the symbol list to be used for decoding.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<ShowSymbols> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

### 18.18.13.2 Filter

There are two commands for each parameter, that you can use for defining the SENT settings.

For example, to set the *Frame type =TransmissionSequence > Field =Data >01100* value you can use one of the following commands:

- `SBUS:SENT:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:SENT:FILTer:DMIN "TransmissionSequence", "Data", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:SENT:FILTer:CHKall ... 1743
SBUS<sb>:SENT:FILTer:CLR ... 1743
SBUS<sb>:SENT:FILTer:INVert ... 1744
SBUS<sb>:SENT:FILTer:RST ... 1744
SBUS<sb>:SENT:FILTer:FRENable ... 1744
SBUS<sb>:SENT:FILTer:FRAMe<fr>:ENABle ... 1744
SBUS<sb>:SENT:FILTer:BIT ... 1744
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1744
SBUS<sb>:SENT:FILTer:DMAX ... 1745
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1745
SBUS<sb>:SENT:FILTer:DMIN ... 1745
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1745
SBUS<sb>:SENT:FILTer:DOPerator ... 1746
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1746
SBUS<sb>:SENT:FILTer:FIENable ... 1746
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1746
SBUS<sb>:SENT:FILTer:IMAX ... 1746
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1746
SBUS<sb>:SENT:FILTer:IMIN ... 1747
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1747
SBUS<sb>:SENT:FILTer:IOPerator ... 1747
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1747
SBUS<sb>:SENT:FILTer:ERENable ... 1748
SBUS<sb>:SENT:FILTer:ERRor<n>:ENABle ... 1748

### SBUS<sb>:SENT:FILTer:CHKall

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 733

### SBUS<sb>:SENT:FILTer:CLR

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 733

**SBUS<sb>:SENT:FILTer:INVert**

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 733

**SBUS<sb>:SENT:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 733

**SBUS<sb>:SENT:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:SENT:FILTer:FRENable?** <Enabler>
**SBUS<sb>:SENT:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables the filtering on SENT frames. Only the frames that match the selected filter conditions are displayed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 734

**SBUS<sb>:SENT:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:SENT:FILTer:BIT?** <Bit>
**SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 734

---

**SBUS<sb>:SENT:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:SENT:FILTer:DMAX?** <Data>
**SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 734

---

**SBUS<sb>:SENT:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:SENT:FILTer:DMIN?** <Data>
**SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 734

**SBUS<sb>:SENT:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:SENT:FILTer:DOPerator?** <Operator>
**SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 734

**SBUS<sb>:SENT:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:SENT:FILTer:FIENable?** <Enabler>
**SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> 1 to 4, index of the serial bus

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 734

**SBUS<sb>:SENT:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:SENT:FILTer:IMAX?** <Data>
**SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

You can set the operator with `SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 734

---

**SBUS<sb>:SENT:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:SENT:FILTer:IMIN?** <Data>
**SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 734

---

**SBUS<sb>:SENT:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:SENT:FILTer:IOPerator?** <Operator>
**SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 734

**SBUS<sb>:SENT:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:SENT:FILTer:ERENable?** <Enabler>
**SBUS<sb>:SENT:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 735

### 18.18.13.3 Format

This section describes the commands to specify and query frame formats for the "D-PHY" protocol.

SBUS<sb>:SENT:FORMat:SELect....1749
SBUS<sb>:SENT:FORMat:ADDFrame....1749
SBUS<sb>:SENT:FORMat:LOAD....1749
SBUS<sb>:SENT:FORMat:SAVE....1749
SBUS<sb>:SENT:FORMat:CLR....1750
SBUS<sb>:SENT:FORMat:FCOunt?....1750
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ADDField....1750
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLDCount?....1750
SBUS<sb>:SENT:FORMat:FRAMe<fr>:NAME....1751
SBUS<sb>:SENT:FORMat:FRAMe<fr>:COLor....1751
SBUS<sb>:SENT:FORMat:FRAMe<fr>:TYPE....1751
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID1Field....1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID2Field....1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID3Field....1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID1Value....1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID2Value....1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID3Value....1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:NAME....1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:COLor....1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1753
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1753
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1754
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1754
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1754
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1755
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1756
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1756
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1756

SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:PARA ..... 1757
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:PARB ..... 1757

**SBUS<sb>:SENT:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 736

**SBUS<sb>:SENT:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 736

**SBUS<sb>:SENT:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 737

**SBUS<sb>:SENT:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 737

---

**SBUS<sb>:SENT:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 737

---

**SBUS<sb>:SENT:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 737

---

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 738

---

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 737

---

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 737

---

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 737

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 737

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 737

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 738

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 738

---

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<FirstBit>

Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 739

---

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitCount>

Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 739

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 739

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 739

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 739

---

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 740

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value>
Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 740

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 740

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 740

---

**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

Available if `SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<ParameterB>
Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 740

### 18.18.13.4 Software trigger

There are two commands for each parameter, that you can use for defining the SENT settings.

For example, to set the *Frame type =TransmissionSequence > Field =Data >01100* value you can use one of the following commands:

- `TRIGger:SBSW:SENT:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:SBSW:SENT:DMIN "TransmissionSequence", "Data", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:SENT:CHKall ..... 1758
TRIGger:SBSW:SENT:CLR ..... 1758
TRIGger:SBSW:SENT:INVert ..... 1758
TRIGger:SBSW:SENT:RST ..... 1758
TRIGger:SBSW:SENT:FRAMe<fr>:ENABle ..... 1759
TRIGger:SBSW:SENT:FRENable ..... 1759
TRIGger:SBSW:SENT:BIT ..... 1759
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:BIT ..... 1759

TRIGger:SBSW:SENT:FIENable....1759
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:ENABle....1759
TRIGger:SBSW:SENT:DMAX....1760
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DMAX....1760
TRIGger:SBSW:SENT:DMIN....1760
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DMIN....1760
TRIGger:SBSW:SENT:DOPerator....1760
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DOPerator....1760
TRIGger:SBSW:SENT:IMAX....1761
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IMAX....1761
TRIGger:SBSW:SENT:IMIN....1761
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IMIN....1761
TRIGger:SBSW:SENT:IOPerator....1762
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IOPerator....1762
TRIGger:SBSW:SENT:ERENable....1762
TRIGger:SBSW:SENT:ERRor<m>:ENABle....1762
SBUS<sb>:SENT:SWTindex?....1762
SBUS<sb>:SENT:SWTTime?....1763

### TRIGger:SBSW:SENT:CHKall

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 742

### TRIGger:SBSW:SENT:CLR

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 742

### TRIGger:SBSW:SENT:INVert

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 742

### TRIGger:SBSW:SENT:RST

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 742

---

**TRIGger:SBSW:SENT:FRAMe<fr>:ENABle** <Enable>
**TRIGger:SBSW:SENT:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:SENT:FRENable?** <Enabler>

rcset

**Setting parameters:**
<Frame>

**Parameters for setting and query:**
<Enabler> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 742

---

**TRIGger:SBSW:SENT:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:SENT:BIT?** <Bit>
**TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 742

---

**TRIGger:SBSW:SENT:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:SENT:FIENable?** <Enabler>
**TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 742

---

**TRIGger:SBSW:SENT:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SENT:DMAX?** <Data>
**TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 742

---

**TRIGger:SBSW:SENT:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SENT:DMIN?** <Data>
**TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 742

---

**TRIGger:SBSW:SENT:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:SENT:DOPerator?** <Operator>
**TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 742

---

**TRIGger:SBSW:SENT:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SENT:IMAX?** <Data>
**TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 742

---

**TRIGger:SBSW:SENT:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SENT:IMIN?** <Data>
**TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 742

**TRIGger:SBSW:SENT:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:SENT:IOPerator?** <Operator>
**TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 742

**TRIGger:SBSW:SENT:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:SENT:ERENable?** <Enabler>
**TRIGger:SBSW:SENT:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 744

**SBUS<sb>:SENT:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SENT:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.13.5 Decode results

SBUS<sb>:SENT:FCOunt? 1763
SBUS<sb>:SENT:FRAMe<fr>:BITRate? 1764
SBUS<sb>:SENT:FRAMe<fr>:CSValue? 1764
SBUS<sb>:SENT:FRAMe<fr>:DATA? 1764
SBUS<sb>:SENT:FRAMe<fr>:FDValues? 1765
SBUS<sb>:SENT:FRAMe<fr>:FSCom? 1765
SBUS<sb>:SENT:FRAMe<fr>:FSDTa? 1765
SBUS<sb>:SENT:FRAMe<fr>:IDTYpe? 1765
SBUS<sb>:SENT:FRAMe<fr>:IDValue? 1766
SBUS<sb>:SENT:FRAMe<fr>:MTPDuration? 1766
SBUS<sb>:SENT:FRAMe<fr>:FLDCount? 1766
SBUS<sb>:SENT:FRAMe<fr>:FLD<fl>:STATe? 1767
SBUS<sb>:SENT:FRAMe<fr>:FLD<fl>:VALue? 1767
SBUS<sb>:SENT:FRAMe<fr>:FLD<fl>:FVALue? 1767
SBUS<sb>:SENT:FRAMe<fr>:FLD<fl>:LABel? 1768
SBUS<sb>:SENT:FRAMe<fr>:PAPTicks? 1768
SBUS<sb>:SENT:FRAMe<fr>:SCOM? 1768
SBUS<sb>:SENT:FRAMe<fr>:SDATa? 1769
SBUS<sb>:SENT:FRAMe<fr>:SENSor? 1769
SBUS<sb>:SENT:FRAMe<fr>:STARt? 1769
SBUS<sb>:SENT:FRAMe<fr>:STATus? 1770
SBUS<sb>:SENT:FRAMe<fr>:STOP? 1770
SBUS<sb>:SENT:FRAMe<fr>:SYMBol? 1771
SBUS<sb>:SENT:FRAMe<fr>:SYNCduration? 1771
SBUS<sb>:SENT:FRAMe<fr>:TYPE? 1771

**SBUS<sb>:SENT:FCOunt?**

Returns the number of decoded frames.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count> Total number of decoded frames.

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:BITRate?**

Returns the bit rate of the frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<BitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:CSValue?**

Returns the CRC sequence value of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<ChecksumValue> Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:DATA?**

Returns the data of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<data> Comma-separated sequence of integer values (N, D1, D2,…, DN). N is the number of nibbles in the frame, and D1…DN are the values of the nibbles.

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:FDValues?**

Returns the formatted data values.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedDatVals>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:FSCom?**

Retruns the formatted status communication.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedStatusComm>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:FSDTa?**

Returns the formatted symbolic data values.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedSymbDat>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:IDTYpe?**

Returns the identifier type of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<IdentifierType> B4 | B8

B4: standard format, 4 bit
B8: extended format, 8 bit

*RST: B4

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:IDValue?**

Returns the identifier value of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<IdtfVal> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:MTPDuration?**

Returns the duration of the main trigger pulse (MTP), which is an additional pulse added to the frame when running in "Short PWM Code" (SPC) mode. Withpulse width modulationn (PWM), you can connect multiple sensors to the bus. The duration of the MTP defines, which sensor starts the transmission.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<MTPDuration> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:FLDCount?**

Returns the number of fields in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:FLD<fl>:STATe?**

Returns the state of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | SYNC | PULSe | CRC | LENGth | FORM | PAUSe | INComplete
*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:FLD<fl>:VALue?**

Returns the value of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> Range: 0 to 18446744073709551615
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:FLD<fl>:LABel?**

Returns the label of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:PAPTicks?**

Returns the number of the pulse pause clock ticks.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<PausePlsTicks> Range: 12 to 768
Increment: 1
*RST: 12

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:SCOM?**

Returns the value of the status/communication pulse.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrmStatusComm> Range: 0 to 15
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:SDATa?**

Returns the symbolic data of the frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<SymbolicData>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:SENSor?**

Returns the sensor resposne time.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<SensRespTime> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:STARt?**

Returns the start time of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:STATus?**

Returns the overall state of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameState> OK | SYNC | PULSe | CRC | LENGth | FORM | PAUSe | INComplete

OK: the frame is valid.
SYNC: Synchronization error occurred.
PULse: Pulse error occurred.
CRC: Cyclic redundancy check failed.
LENGth: Irregular frame length error occurred.
FORM: Format error occurred.
INComplete: The frame is not completely contained in the acquisition. The acquired part of the frame is valid.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SENT:FRAMe<fr>:STOP?**

Returns the stop time of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SENT:FRAMe<fr>:SYMBol?**

Returns the symbolic label of the specified frame if the label list is enabled.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<SymbolicName>

**Example:** `BUS:SENT:FRAMe:SYMBol?`
Response: `Air Temperature`

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SENT:FRAMe<fr>:SYNCduration?**

Returns the time of the synchronization pulse.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameSyncTime> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SENT:FRAMe<fr>:TYPE?**

Returns the type of SENT message.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameType> TRSQ | SMSG | ESSM | ELSM | PAUSe | UNKNown

`TRSQ`: transmission sequence
`SMSG`: short serial message
`ESSM`: enhanced 12 bit message
`ELSM`: enhanced 16 bit message

*RST: TRSQ

**Usage:** Query only
Asynchronous command

## 18.18.14 ARINC 429 (option R&S MXO3-K530)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.


- Configuration....1772
- Filter....1777
- Format....1781
- Software trigger....1790
- Decode results....1795


### 18.18.14.1 Configuration


SBUS<sb>:ARINc:BRMode....1772
SBUS<sb>:ARINc:BRValue....1772
SBUS<sb>:ARINc:POLarity....1773
SBUS<sb>:ARINc:SOURce....1773
SBUS<sb>:ARINc:THReshold:HIGH....1773
SBUS<sb>:ARINc:THReshold:HYSTeresis....1774
SBUS<sb>:ARINc:THReshold:LOW....1774
SBUS<sb>:ARINc:NEWList....1774
SBUS<sb>:ARINc:SYMBols....1774
SBUS<sb>:ARINc:MAXGap:BITS....1775
SBUS<sb>:ARINc:MAXGap:SELect....1775
SBUS<sb>:ARINc:MINGap:BITS....1775
SBUS<sb>:ARINc:MINGap:SELect....1776
SBUS<sb>:ARINc:POSition....1776
SBUS<sb>:ARINc:SCALe....1776


**SBUS<sb>:ARINc:BRMode** <BitRateMode>

Sets the bit rate mode to high or low speed. You can set an exact bitrate value with `SBUS<sb>:ARINc:BRValue`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<BitRateMode> HIGH | LOW
*RST: HIGH

**Usage:** Asynchronous command

**Manual operation:** See "Bit rate mode, Bit rate" on page 749

**SBUS<sb>:ARINc:BRValue** <BitRateValue>

Sets the number of transmitted bits per second.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<BitRateValue> Range: 10000 to 110000
Increment: 100
*RST: 100000
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "Bit rate mode, Bit rate" on page 749

---

**SBUS<sb>:ARINc:POLarity** <Polarity>

Selects the wire on which the bus signal is measured : A Leg or B Leg. The setting affects the digitization of the signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Polarity> ALEG | BLEG
*RST: ALEG

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 749

---

**SBUS<sb>:ARINc:SOURce** Source

Sets the channel for the signal source.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
Source C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 749

---

**SBUS<sb>:ARINc:THReshold:HIGH** <Upper threshold>

Sets the upper threshold level of the signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Upper threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 750

---

**SBUS<sb>:ARINc:THReshold:HYSTeresis** <Hysteresis>

Sets a hysteresis value.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 750

---

**SBUS<sb>:ARINc:THReshold:LOW** <Lower threshold>

Sets the lower threshold level of the signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Lower threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 750

---

**SBUS<sb>:ARINc:NEWList** <FileName>

Loads a symbol list file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName> String parameter with path and file name.

**Example:**
```
SBUS1:ARIN:NEWList
'/home/storage/userData/Protocol/ARINC.csv'
SBUS1:ARIN:SYMBols ON
```

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Show symbols" on page 751

---

**SBUS<sb>:ARINc:SYMBols** <ShowSymbols>

Activates the symbol list to be used for decoding.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ShowSymbols> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show symbols" on page 751

---

**SBUS<sb>:ARINc:MAXGap:BITS** <MaxGapBits>

Sets the value for the maximum gap between two words.

See also: `SBUS<sb>:ARINc:MAXGap:SELect` on page 1775.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MaxGapBits> Range: 0 to 1000
Increment: 1
*RST: 100
Default unit: bit

**Usage:** Asynchronous command

**Manual operation:** See "Min gap time,Max gap time" on page 749

---

**SBUS<sb>:ARINc:MAXGap:SELect** <MaxGapSelect>

Enables the detection of the maximum gap time during decoding.

You can specify the maximum gap time with `SBUS<sb>:ARINc:MAXGap:BITS`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MaxGapSelect> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Min gap time,Max gap time" on page 749

---

**SBUS<sb>:ARINc:MINGap:BITS** <MinGapBits>

Sets a value for the minimum timing gap between two words.

See also: `SBUS<sb>:ARINc:MINGap:SELect`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**

<MinGapBits> Range: 0 to 100
Increment: 1
*RST: 4
Default unit: bit

**Usage:** Asynchronous command

**Manual operation:** See "Min gap time,Max gap time" on page 749

---

**SBUS<sb>:ARINc:MINGap:SELect** <MinGapSelect>

Enables the detection of the minimum idle time between two words during decoding.

To specify the minimum gap, use `SBUS<sb>:ARINc:MINGap:BITS`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<MinGapSelect> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Min gap time,Max gap time" on page 749

---

**SBUS<sb>:ARINc:POSition** <Position>

Sets the vertical position of the ARINC signal.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<Position>

**Usage:** Asynchronous command

---

**SBUS<sb>:ARINc:SCALe** <Scale>

Set the vertical scale of the indicated ARINC signal.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<Scale>

**Usage:** Asynchronous command

### 18.18.14.2 Filter

There are two commands for each parameter, that you can use for defining the ARINC 429 settings.

For example, to set the *Frame type =ARINC429-Word* > *Field =Data* >*Data* value you can use one of the following commands:

- `SBUS:ARINC:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:ARINC:FILTer:DMIN "ARINC429-Word", "Data", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:ARINc:FILTer:CHKall ..... 1777
SBUS<sb>:ARINc:FILTer:CLR ..... 1777
SBUS<sb>:ARINc:FILTer:INVert ..... 1778
SBUS<sb>:ARINc:FILTer:RST ..... 1778
SBUS<sb>:ARINc:FILTer:BIT ..... 1778
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:BIT ..... 1778
SBUS<sb>:ARINc:FILTer:ERENable ..... 1779
SBUS<sb>:ARINc:FILTer:ERRor<n>:ENABle ..... 1779
SBUS<sb>:ARINc:FILTer:DMAX ..... 1779
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DMAX ..... 1779
SBUS<sb>:ARINc:FILTer:DMIN ..... 1779
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DMIN ..... 1779
SBUS<sb>:ARINc:FILTer:DOPerator ..... 1780
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ..... 1780
SBUS<sb>:ARINc:FILTer:FIENable ..... 1780
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:ENABle ..... 1780
SBUS<sb>:ARINc:FILTer:FRENable ..... 1780
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:ENABle ..... 1780

---

**SBUS<sb>:ARINc:FILTer:CHKall**

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 752

---

**SBUS<sb>:ARINc:FILTer:CLR**

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 752

---

**SBUS<sb>:ARINc:FILTer:INVert**

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 752

---

**SBUS<sb>:ARINc:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 752

---

**SBUS<sb>:ARINc:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:ARINc:FILTer:BIT?** <Bit>
**SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 753

**SBUS<sb>:ARINc:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:ARINc:FILTer:ERENable?** <Enabler>
**SBUS<sb>:ARINc:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 754

**SBUS<sb>:ARINc:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:ARINc:FILTer:DMAX?** <Data>
**SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 753

**SBUS<sb>:ARINc:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:ARINc:FILTer:DMIN?** <Data>
**SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 753

---

**SBUS<sb>:ARINc:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:ARINc:FILTer:DOPerator?** <Operator>
**SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 753

---

**SBUS<sb>:ARINc:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:ARINc:FILTer:FIENable?** <Enabler>
**SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 753

---

**SBUS<sb>:ARINc:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:ARINc:FILTer:FRENable?** <Enabler>
**SBUS<sb>:ARINc:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables the filtering on ARINC 429 frames. Only the frames that match the selected filter conditions are displayed.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 753

### 18.18.14.3 Format

This section describes the commands to specify and query frame formats for the "ARINC 429" protocol.

SBUS<sb>:ARINc:FORMat:SELect ... 1781
SBUS<sb>:ARINc:FORMat:ADDFrame ... 1782
SBUS<sb>:ARINc:FORMat:LOAD ... 1782
SBUS<sb>:ARINc:FORMat:SAVE ... 1782
SBUS<sb>:ARINc:FORMat:CLR ... 1783
SBUS<sb>:ARINc:FORMat:FCOunt? ... 1783
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ADDField ... 1783
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLDCount? ... 1783
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:NAME ... 1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:COLor ... 1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:TYPE ... 1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID1Field ... 1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID2Field ... 1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID3Field ... 1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID1Value ... 1785
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID2Value ... 1785
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID3Value ... 1785
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:NAME ... 1785
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:COLor ... 1785
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ... 1786
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:BITCount ... 1786
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:REPetition ... 1786
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:BITorder ... 1787
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FORMat ... 1787
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ... 1788
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ... 1788
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ... 1789
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:UNIT ... 1789
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:PARA ... 1790
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:PARB ... 1790

---

**SBUS<sb>:ARINc:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 754

---

**SBUS<sb>:ARINc:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 754

---

**SBUS<sb>:ARINc:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 755

---

**SBUS<sb>:ARINc:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 755

**SBUS<sb>:ARINc:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 755

**SBUS<sb>:ARINc:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 755

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 756

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 756

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 756

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 756

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 756

---

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 756

---

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 757

---

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 757

---

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<FirstBit>

Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 757

---

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<BitCount>

Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 757

---

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 758

---

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 758

---

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 758

---

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 759

---

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field
<p> Index of the value

**Parameters:**
<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 759

---

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field
<p> Index of the value

**Parameters:**
<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 759

---

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 759

**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

Available if `SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

| | |
|---|---|
| <sb> | 1 to 4, index of the serial bus |
| <fr> | Index of the frame |
| <fl> | Index of the field |

**Parameters:**

| | | |
|---|---|---|
| <ParameterB> | Range: | -1E+20 to 1E+20 |
| | Increment: | 1E-12 |
| | *RST: | 0 |

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 759

### 18.18.14.4 Software trigger

There are two commands for each parameter, that you can use for defining the ARINC 429 settings.

For example, to set the *Frame type =ARINC429-Word* > *Field =Data* >*Data* value you can use one of the following commands:

- `TRIGger:SBSW:ARINc:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:SBSW:ARINc:DMIN "ARINC429-Word", "Data", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:ARINc:CHKall ..... 1791
TRIGger:SBSW:ARINc:CLR ..... 1791
TRIGger:SBSW:ARINc:INVert ..... 1791
TRIGger:SBSW:ARINc:RST ..... 1791
TRIGger:SBSW:ARINc:FRENable ..... 1791
TRIGger:SBSW:ARINc:FRAMe<fr>:ENABle ..... 1791
TRIGger:SBSW:ARINc:BIT ..... 1792
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:BIT ..... 1792
TRIGger:SBSW:ARINc:DMAX ..... 1792
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DMAX ..... 1792
TRIGger:SBSW:ARINc:DMIN ..... 1792
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DMIN ..... 1792
TRIGger:SBSW:ARINc:DOPerator ..... 1793
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DOPerator ..... 1793
TRIGger:SBSW:ARINc:FIENable ..... 1793

TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:ENABle....1793
TRIGger:SBSW:ARINc:ERENable....1794
TRIGger:SBSW:ARINc:ERRor<m>:ENABle....1794
SBUS<sb>:ARINc:SWTindex?....1794
SBUS<sb>:ARINc:SWTTime?....1794

**TRIGger:SBSW:ARINc:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 760

**TRIGger:SBSW:ARINc:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 760

**TRIGger:SBSW:ARINc:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 760

**TRIGger:SBSW:ARINc:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 760

**TRIGger:SBSW:ARINc:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:ARINc:FRENable?** <Enabler>
**TRIGger:SBSW:ARINc:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 761

---

**TRIGger:SBSW:ARINc:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:ARINc:BIT?** <Bit>
**TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 761

---

**TRIGger:SBSW:ARINc:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:ARINc:DMAX?** <Data>
**TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 761

---

**TRIGger:SBSW:ARINc:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:ARINc:DMIN?** <Data>
**TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 761

---

**TRIGger:SBSW:ARINc:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:ARINc:DOPerator?** <Operator>
**TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 761

---

**TRIGger:SBSW:ARINc:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:ARINc:FIENable?** <Enabler>
**TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 761

**TRIGger:SBSW:ARINc:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:ARINc:ERENable?** <Enabler>
**TRIGger:SBSW:ARINc:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 762

**SBUS<sb>:ARINc:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:ARINc:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.14.5 Decode results

This section describes the commands to query the results of the decoding.

SBUS<sb>:ARINc:WCOunt? 1795
SBUS<sb>:ARINc:WORD<w>:FLDCount? 1795
SBUS<sb>:ARINc:WORD<w>:BITRate? 1795
SBUS<sb>:ARINc:WORD<w>:FLD<fl>:STATe? 1796
SBUS<sb>:ARINc:WORD<w>:FLD<fl>:LABel? 1796
SBUS<sb>:ARINc:WORD<w>:FLD<fl>:VALue? 1797
SBUS<sb>:ARINc:WORD<w>:FLD<fl>:FVALue? 1797
SBUS<sb>:ARINc:WORD<w>:DATA? 1797
SBUS<sb>:ARINc:WORD<w>:LABel? 1798
SBUS<sb>:ARINc:WORD<w>:PATTern? 1798
SBUS<sb>:ARINc:WORD<w>:SDI? 1798
SBUS<sb>:ARINc:WORD<w>:SSM? 1799
SBUS<sb>:ARINc:WORD<w>:STARt? 1799
SBUS<sb>:ARINc:WORD<w>:STATe? 1799
SBUS<sb>:ARINc:WORD<w>:STOP? 1800
SBUS<sb>:ARINc:WORD<w>:SYMBol? 1800

---

**SBUS<sb>:ARINc:WCOunt?**

Returns the number of decoded words.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:ARINc:WORD<w>:FLDCount?**

Returns the number of fields available in the seleced word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:ARINc:WORD<w>:BITRate?**

Returns the bit rate of the word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<PrimaryBitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:ARINc:WORD<w>:FLD<fl>:STATe?**

Returns the overall state of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | CODE | GAP | PAR | INC | UNKN
CODE: coding error occured.
GAP: timing gap error occured.
PAR: parity error occured.
INComplete: the sequence is not completely contained in the acquisition
UNKNown: unknown frame type
*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:ARINc:WORD<w>:FLD<fl>:LABel?**

Returns the label of the specified field in the specified word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:ARINc:WORD<w>:FLD<fl>:VALue?**

Returns the value of the specified field in the specified word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

<fl> Index of the field

**Return values:**
<Value> Range: 0 to 18446744073709551615
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:ARINc:WORD<w>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field in the specified word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:ARINc:WORD<w>:DATA?**

Returns the data of the specified word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<FrameData> Range: 0 to 524287
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:ARINc:WORD<w>:LABel?**

Returns the label of the specified word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**

<FrameLabel> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:ARINc:WORD<w>:PATTern?**

Returns all 32 bits of the specified word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**

<FrmDatPatt> List of comma separated values.

**Example:**
```
SBUS1:ARINc:WORD1:PATTern?
#H04,#H10,#H04
```

**Usage:** Query only
Asynchronous command

**SBUS<sb>:ARINc:WORD<w>:SDI?**

Returns the source/destination identifier (SDI) bits of the specified word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**

<SDI> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:ARINc:WORD<w>:SSM?**

Returns the sign/status matrix (SSM) bits of the specified word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**

<SSM> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:ARINc:WORD<w>:STARt?**

Returns the start time of the specified word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**

<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

**SBUS<sb>:ARINc:WORD<w>:STATe?**

Returns the overall state of the specified word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**

<FrameState> OK | CODE | GAP | PAR | INC | UNKN

CODE: coding error occured.
GAP: timing gap error occured.
PAR: parity error occured.
INComplete: the sequence is not completely contained in the acquisition

UNKNown: unknown frame type

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:ARINc:WORD<w>:STOP?**

Returns the end time of the specified word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:ARINc:WORD<w>:SYMBol?**

Returns the label name of the word ID.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<Translation>

**Usage:** Query only
Asynchronous command

## 18.18.15 MIL-1553 (option R&S MXO3-K530)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.

- Configuration....1801
- Filter....1803
- Format....1809
- Software trigger....1817
- Decode results....1823

### 18.18.15.1 Configuration

SBUS<sb>:MILStd:POLarity....1801
SBUS<sb>:MILStd:SOURce....1801
SBUS<sb>:MILStd:THReshold:HIGH....1801
SBUS<sb>:MILStd:THReshold:HYSTeresis....1802
SBUS<sb>:MILStd:THReshold:LOW....1802
SBUS<sb>:MILStd:NEWList....1802
SBUS<sb>:MILStd:POSition....1802
SBUS<sb>:MILStd:SCALe....1803
SBUS<sb>:MILStd:SYMBols....1803

---

**SBUS<sb>:MILStd:POLarity** <Polarity>

Sets the wire on which the bus signal is measured : normal or inverted. The setting affects the digitization of the signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Polarity> NORMal | INVerted
*RST: NORMal

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 767

---

**SBUS<sb>:MILStd:SOURce** Source

Sets the channel for the signal source.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
Source C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 767

---

**SBUS<sb>:MILStd:THReshold:HIGH** <Threshold high>

Sets the lower threshold level of the signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold high>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 768

---

**SBUS<sb>:MILStd:THReshold:HYSTeresis** <Hysteresis>

Sets a hysteresis value.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 768

---

**SBUS<sb>:MILStd:THReshold:LOW** <Threshold low>

Sets the lower threshold level of the signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold low>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 768

---

**SBUS<sb>:MILStd:NEWList** <FileName>

Loads a symbol list file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName> String parameter with path and file name.

**Example:**
```
SBUS1:MILStd:NEWList
'/home/storage/userData/Protocol/MILStd.csv'
SBUS1:MILStd:SYMBols ON
```

**Usage:** Setting only
Asynchronous command

---

**SBUS<sb>:MILStd:POSition** <Position>

Sets the vertical position of the MIL-1553 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Position>

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:SCALe** <Scale>

Set the vertical scale of the indicated MIL-1553 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Scale>

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:SYMBols** <ShowSymbols>

Activates the symbol list to be used for decoding.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ShowSymbols> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

### 18.18.15.2 Filter

There are two commands for each parameter that you can use for defining the MIL-1553 settings.

For example, to set the *Frame type =Command > Field =Info >01100* value you can use one of the following commands:

- `SBUS:MIL:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:MIL:FILTer:DMIN "Command", "Info", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:MILStd:FILTer:CHKall ..... 1804
SBUS<sb>:MILStd:FILTer:CLR ..... 1804
SBUS<sb>:MILStd:FILTer:INVert ..... 1804
SBUS<sb>:MILStd:FILTer:RST ..... 1805
SBUS<sb>:MILStd:FILTer:BIT ..... 1805
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:BIT ..... 1805
SBUS<sb>:MILStd:FILTer:ERENable ..... 1805
SBUS<sb>:MILStd:FILTer:ERRor<n>:ENABle ..... 1805
SBUS<sb>:MILStd:FILTer:DMAX ..... 1806

SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DMAX ..... 1806
SBUS<sb>:MILStd:FILTer:DMIN ..... 1806
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DMIN ..... 1806
SBUS<sb>:MILStd:FILTer:DOPerator ..... 1806
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ..... 1806
SBUS<sb>:MILStd:FILTer:IMAX ..... 1807
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IMAX ..... 1807
SBUS<sb>:MILStd:FILTer:IMIN ..... 1807
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IMIN ..... 1807
SBUS<sb>:MILStd:FILTer:IOPerator ..... 1808
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ..... 1808
SBUS<sb>:MILStd:FILTer:FRENable ..... 1808
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:ENABle ..... 1808
SBUS<sb>:MILStd:FILTer:FIENable ..... 1808
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:ENABle ..... 1808

### SBUS<sb>:MILStd:FILTer:CHKall

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 770

### SBUS<sb>:MILStd:FILTer:CLR

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 770

### SBUS<sb>:MILStd:FILTer:INVert

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 770

**SBUS<sb>:MILStd:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 770

**SBUS<sb>:MILStd:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:MILStd:FILTer:BIT?** <Bit>
**SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 771

**SBUS<sb>:MILStd:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:MILStd:FILTer:ERENable?** <Enabler>
**SBUS<sb>:MILStd:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 772

**SBUS<sb>:MILStd:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:MILStd:FILTer:DMAX?** <Data>
**SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 771

**SBUS<sb>:MILStd:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:MILStd:FILTer:DMIN?** <Data>
**SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 771

**SBUS<sb>:MILStd:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:MILStd:FILTer:DOPerator?** <Operator>
**SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**

<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 771

---

**SBUS<sb>:MILStd:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:MILStd:FILTer:IMAX?** <Data>
**SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

You can set the operator with `SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max>
Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 771

---

**SBUS<sb>:MILStd:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:MILStd:FILTer:IMIN?** <Data>
**SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min>
Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 771

**SBUS<sb>:MILStd:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:MILStd:FILTer:IOPerator?** <Operator>
**SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 771

**SBUS<sb>:MILStd:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:MILStd:FILTer:FRENable?** <Enabler>
**SBUS<sb>:MILStd:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables the filtering on MIL-1553 frames. Only the frames that match the selected filter conditions are displayed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 770

**SBUS<sb>:MILStd:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:MILStd:FILTer:FIENable?** <Enabler>
**SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 771

### 18.18.15.3 Format

This section describes the commands to specify and query frame formats for the "MIL-153" protocol.

SBUS<sb>:MILStd:FORMat:SELect....1809
SBUS<sb>:MILStd:FORMat:ADDFrame....1810
SBUS<sb>:MILStd:FORMat:LOAD....1810
SBUS<sb>:MILStd:FORMat:SAVE....1810
SBUS<sb>:MILStd:FORMat:CLR....1810
SBUS<sb>:MILStd:FORMat:FCOunt?....1811
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ADDField....1811
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLDCount?....1811
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:NAME....1811
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:COLor....1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:TYPE....1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID1Field....1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID2Field....1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID3Field....1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID1Value....1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID2Value....1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID3Value....1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:NAME....1813
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:COLor....1813
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1813
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1814
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1814
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1814
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1815
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1816
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1816
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1816
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1817
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:PARA....1817
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:PARB....1817

---

**SBUS<sb>:MILStd:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**SBUS<sb>:MILStd:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<FirstBit> Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value>
Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**
<Label>

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Unit>

**Usage:** Asynchronous command

---

**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

Available if `SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<ParameterB>
Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

### 18.18.15.4 Software trigger

There are two commands for each parameter that you can use for defining the MIL-1553 settings.

For example, to set the *Frame type =Command > Field =Info >01100* value you can use one of the following commands:

- `TRIGger:SBSW:MIL:FRAMe1:FLD1:DMIN 01100`

Defines the parameter by using the index <m> for the frame number and <n> for the field number.

- `TRIGger:SBSW:MIL:DMIN "Command", "Info", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:MILStd:CHKall ..... 1818
TRIGger:SBSW:MILStd:CLR ..... 1818
TRIGger:SBSW:MILStd:INVert ..... 1819
TRIGger:SBSW:MILStd:RST ..... 1819
TRIGger:SBSW:MILStd:FRENable ..... 1819
TRIGger:SBSW:MILStd:FRAMe<fr>:ENABle ..... 1819
TRIGger:SBSW:MILStd:BIT ..... 1819
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:BIT ..... 1819
TRIGger:SBSW:MILStd:DMAX ..... 1820
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DMAX ..... 1820
TRIGger:SBSW:MILStd:DMIN ..... 1820
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DMIN ..... 1820
TRIGger:SBSW:MILStd:DOPerator ..... 1820
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DOPerator ..... 1820
TRIGger:SBSW:MILStd:FIENable ..... 1821
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:ENABle ..... 1821
TRIGger:SBSW:MILStd:IMAX ..... 1821
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IMAX ..... 1821
TRIGger:SBSW:MILStd:IMIN ..... 1822
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IMIN ..... 1822
TRIGger:SBSW:MILStd:IOPerator ..... 1822
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IOPerator ..... 1822
TRIGger:SBSW:MILStd:ERENable ..... 1822
TRIGger:SBSW:MILStd:ERRor<m>:ENABle ..... 1822
SBUS<sb>:MILStd:SWTindex? ..... 1823
SBUS<sb>:MILStd:SWTTime? ..... 1823

### TRIGger:SBSW:MILStd:CHKall

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 773

### TRIGger:SBSW:MILStd:CLR

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 773

**TRIGger:SBSW:MILStd:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 773

**TRIGger:SBSW:MILStd:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 773

**TRIGger:SBSW:MILStd:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:MILStd:FRENable?** <Enabler>
**TRIGger:SBSW:MILStd:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 773

**TRIGger:SBSW:MILStd:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:MILStd:BIT?** <Bit>
**TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 774

**TRIGger:SBSW:MILStd:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:MILStd:DMAX?** <Data>
**TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 774

**TRIGger:SBSW:MILStd:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:MILStd:DMIN?** <Data>
**TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 774

**TRIGger:SBSW:MILStd:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:MILStd:DOPerator?** <Operator>
**TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 774

---

**TRIGger:SBSW:MILStd:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:MILStd:FIENable?** <Enabler>
**TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 774

---

**TRIGger:SBSW:MILStd:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:MILStd:IMAX?** <Data>
**TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 774

**TRIGger:SBSW:MILStd:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:MILStd:IMIN?** <Data>
**TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> *

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 774

**TRIGger:SBSW:MILStd:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:MILStd:IOPerator?** <Operator>
**TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 774

**TRIGger:SBSW:MILStd:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:MILStd:ERENable?** <Enabler>
**TRIGger:SBSW:MILStd:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 775

**SBUS<sb>:MILStd:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:MILStd:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.15.5 Decode results

SBUS<sb>:MILStd:WCOunt? 1824
SBUS<sb>:MILStd:WORD<w>:FLDCount? 1824
SBUS<sb>:MILStd:WORD<w>:BITRate? 1824
SBUS<sb>:MILStd:WORD<w>:FLD<fl>:STATe? 1824
SBUS<sb>:MILStd:WORD<w>:FLD<fl>:LABel? 1825
SBUS<sb>:MILStd:WORD<w>:FLD<fl>:VALue? 1825
SBUS<sb>:MILStd:WORD<w>:FLD<fl>:FVALue? 1826
SBUS<sb>:MILStd:WORD<w>:DATA? 1826
SBUS<sb>:MILStd:WORD<w>:INFO? 1826
SBUS<sb>:MILStd:WORD<w>:RTADdress? 1826
SBUS<sb>:MILStd:WORD<w>:STARt? 1827
SBUS<sb>:MILStd:WORD<w>:STATus? 1827
SBUS<sb>:MILStd:WORD<w>:STOP? 1828
SBUS<sb>:MILStd:WORD<w>:SYMBol? 1828
SBUS<sb>:MILStd:WORD<w>:TYPE? 1828

**SBUS<sb>:MILStd:WCOunt?**

Returns the number of decoded words.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count> Total number of words.

**Usage:** Query only
Asynchronous command

**SBUS<sb>:MILStd:WORD<w>:FLDCount?**

Returns the number of fields available in the seleced word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:MILStd:WORD<w>:BITRate?**

Returns the bit rate of the word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<PrimaryBitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

**SBUS<sb>:MILStd:WORD<w>:FLD<fl>:STATe?**

Returns the overall state of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> Index of the frame

<fl> Index of the field

**Return values:**

<State> OK | SYNC | MANC | PAR | GAP | RT | INComplete | UNKNown

`OK`: the word is valid.
`SYNC`: synchronization error occured.
`MANC`: manchester coding error occured.
`PAR`: parity error occured.
`GAP`: timing gap error occured.
`RT`: remote terminal error occured.
`INComplete`: the sequence is not completely contained in the acquisition
`UNKNown`: unknown frame type

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MILStd:WORD<w>:FLD<fl>:LABel?**

Returns the label of the specified field in the specified word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> Index of the frame

<fl> Index of the field

**Return values:**

<Label>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MILStd:WORD<w>:FLD<fl>:VALue?**

Returns the value of the specified field in the specified word.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

<fl> Index of the field

**Return values:**

<Value> Range: 0 to 18446744073709551615
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:MILStd:WORD<w>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field in the specified word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:MILStd:WORD<w>:DATA?**

Return the data pattern of the specified word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<FrmDatPatt> List of comma separated values.

**Usage:** Query only
Asynchronous command

**SBUS<sb>:MILStd:WORD<w>:INFO?**

Returns the info value for the specified word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<FrameInfo> Range: 0 to 2047
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:MILStd:WORD<w>:RTADdress?**

Returns the RT address for the selected word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<FrameRta> Range: 0 to 31
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MILStd:WORD<w>:STARt?**

Return the start time of the selected word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MILStd:WORD<w>:STATus?**

Returns the overall state of the selected word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<FrameState> OK | SYNC | MANC | PAR | GAP | RT | INComplete | UNKNown
`OK`: the word is valid.
`SYNC`: synchronization error occured.
`MANC`: manchester coding error occured.
`PAR`: parity error occured.
`GAP`: timing gap error occured.
`RT`: remote terminal error occured.
`INComplete`: the sequence is not completely contained in the acquisition

UNKNown: unknown frame type

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MILStd:WORD<w>:STOP?**

Returns the stop time of the selected word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MILStd:WORD<w>:SYMBol?**

Returns the symbolic label of the specified word if the label list is enabled.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**
<SymbolicName>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:MILStd:WORD<w>:TYPE?**

Returns the type of the specified word.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<w> *
Index of the word

**Return values:**

<FrameType> UNKNown | DATA | CMD | STAT | CMST | IM

`CMD`: command word
`CMST`: command/status word
`IM`: inter message. Shows if there are gap errors or response timeout.

*RST: DATA

**Usage:** Query only
Asynchronous command

## 18.18.16 SpaceWire (option R&S MXO3-K530)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.


- Configuration....1829
- Filter....1832
- Format....1836
- Software trigger....1845
- Decode results....1850


### 18.18.16.1 Configuration


SBUS<sb>:SWIRe:DATA:SOURce....1829
SBUS<sb>:SWIRe:STRBe:SOURce....1830
SBUS<sb>:SWIRe:DATA:THReshold....1830
SBUS<sb>:SWIRe:DATA:HYSTeresis....1830
SBUS<sb>:SWIRe:STRBe:THReshold....1830
SBUS<sb>:SWIRe:STRBe:HYSTeresis....1831
SBUS<sb>:SWIRe:MINGap....1831
SBUS<sb>:SWIRe:MGAP....1831
SBUS<sb>:SWIRe:POSition....1831
SBUS<sb>:SWIRe:SCALe....1831


---

**SBUS<sb>:SWIRe:DATA:SOURce** Source

Selects the channel for the data signal source.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

Source C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Data Source" on page 779

---

**SBUS<sb>:SWIRe:STRBe:SOURce** <StrobeSource>

Selects the source for the strobe signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<StrobeSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Strobe Source" on page 779

---

**SBUS<sb>:SWIRe:DATA:THReshold** <DataThreshold>

Sets the threshold for the data channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DataThreshold>

**Usage:** Asynchronous command

**Manual operation:** See "Thresholds" on page 780

---

**SBUS<sb>:SWIRe:DATA:HYSTeresis** <DataHysteresis>

Sets the hysteresis for the data channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DataHysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Thresholds" on page 780

---

**SBUS<sb>:SWIRe:STRBe:THReshold** <StrobeThres>

Sets the threshold value for the digitization of the strobe signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<StrobeThres>

**Usage:** Asynchronous command

**Manual operation:** See "Thresholds" on page 780

---

**SBUS<sb>:SWIRe:STRBe:HYSTeresis** <StrobeHyst>

Sets a value for the hysteresis of the strobe signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<StrobeHyst>

**Usage:** Asynchronous command

**Manual operation:** See "Thresholds" on page 780

---

**SBUS<sb>:SWIRe:MINGap** <MinGapTime>
**SBUS<sb>:SWIRe:MGAP** <MinGapTime>

Sets the minimum duration of a gap. Any inactivity greater than this time is interpreted as a gap and leads to a resynchronization to the signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<MinGapTime>
Range: 1E-09 to 1
Increment: 1E-09
*RST: 2E-07
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Min Gap" on page 779

---

**SBUS<sb>:SWIRe:POSition** <Position>

Sets the vertical position of the SpaceWire signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Position>

**Usage:** Asynchronous command

---

**SBUS<sb>:SWIRe:SCALe** <Scale>

Sets the vertical scale of the SpaceWire signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Scale>

**Usage:** Asynchronous command

### 18.18.16.2 Filter

There are two commands for each parameter, that you can use for defining the SpaceWire settings.

For example, to set the *Frame type =SpaceWire-Frame* > *Field =Data* >*Data* value you can use one of the following commands:

- `SBUS:SWIRe:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:SWIRe:FILTer:DMIN "SpaceWire-Frame", "Data", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:SWIRe:FILTer:CHKall ... 1832
SBUS<sb>:SWIRe:FILTer:CLR ... 1833
SBUS<sb>:SWIRe:FILTer:INVert ... 1833
SBUS<sb>:SWIRe:FILTer:RST ... 1833
SBUS<sb>:SWIRe:FILTer:BIT ... 1833
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1833
SBUS<sb>:SWIRe:FILTer:ERENable ... 1834
SBUS<sb>:SWIRe:FILTer:ERRor<n>:ENABle ... 1834
SBUS<sb>:SWIRe:FILTer:DMAX ... 1834
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1834
SBUS<sb>:SWIRe:FILTer:DMIN ... 1834
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1834
SBUS<sb>:SWIRe:FILTer:DOPerator ... 1835
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1835
SBUS<sb>:SWIRe:FILTer:FIENable ... 1835
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1835
SBUS<sb>:SWIRe:FILTer:FRENable ... 1836
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:ENABle ... 1836

---

**SBUS<sb>:SWIRe:FILTer:CHKall**

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 782

**SBUS<sb>:SWIRe:FILTer:CLR**

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 782

**SBUS<sb>:SWIRe:FILTer:INVert**

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 782

**SBUS<sb>:SWIRe:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 782

**SBUS<sb>:SWIRe:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:SWIRe:FILTer:BIT?** <Bit>
**SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 782

---

**SBUS<sb>:SWIRe:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:SWIRe:FILTer:ERENable?** <Enabler>
**SBUS<sb>:SWIRe:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> index of the error

**Parameters:**
<Enable> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 783

---

**SBUS<sb>:SWIRe:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:SWIRe:FILTer:DMAX?** <Data>
**SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 782

---

**SBUS<sb>:SWIRe:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:SWIRe:FILTer:DMIN?** <Data>
**SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 782

---

**SBUS<sb>:SWIRe:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:SWIRe:FILTer:DOPerator?** <Operator>
**SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 782

---

**SBUS<sb>:SWIRe:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:SWIRe:FILTer:FIENable?** <Enabler>
**SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 782

**SBUS<sb>:SWIRe:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:SWIRe:FILTer:FRENable?** <Enabler>
**SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables the filtering on SpaceWire frames. Only the frames that match the selected filter conditions are displayed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 782

### 18.18.16.3 Format

This section describes the commands to specify and query frame formats for the "SpaceWire" protocol.

SBUS<sb>:SWIRe:FORMat:SELect....1837
SBUS<sb>:SWIRe:FORMat:ADDFrame....1837
SBUS<sb>:SWIRe:FORMat:LOAD....1837
SBUS<sb>:SWIRe:FORMat:SAVE....1837
SBUS<sb>:SWIRe:FORMat:CLR....1838
SBUS<sb>:SWIRe:FORMat:FCOunt?....1838
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ADDField....1838
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLDCount?....1838
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:NAME....1839
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:COLor....1839
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:TYPE....1839
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID1Field....1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID2Field....1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID3Field....1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID1Value....1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID2Value....1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID3Value....1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:NAME....1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:COLor....1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1841
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1841
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1842
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1842
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1842
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1843
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1844
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1844

SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:UNIT ..... 1844
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:PARA ..... 1845
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:PARB ..... 1845

**SBUS<sb>:SWIRe:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 784

**SBUS<sb>:SWIRe:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 784

**SBUS<sb>:SWIRe:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 785

**SBUS<sb>:SWIRe:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 785

---

**SBUS<sb>:SWIRe:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 785

---

**SBUS<sb>:SWIRe:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 785

---

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 786

---

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 786

---

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 786

---

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 786

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

---

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 786

---

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 787

---

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 787

---

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<FirstBit>

Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 787

---

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitCount>

Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 787

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 788

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 788

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 788

---

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 789

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value>
Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 789

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 789

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 789

---

**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

Available if `SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<ParameterB>
Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 789

### 18.18.16.4 Software trigger

There are two commands for each parameter, that you can use for defining the SpaceWire settings.

For example, to set the *Frame type =SpaceWire-Frame > Field =Data >Data* value you can use one of the following commands:

- `TRIGger:SBSW:SWIRe:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <fr> for the frame number and <fl> for the field number.
- `TRIGger:SBSW:SWIRe:DMIN "SpaceWire-Frame", "Data", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:SWIRe:CHKall ... 1846
TRIGger:SBSW:SWIRe:CLR ... 1846
TRIGger:SBSW:SWIRe:INVert ... 1846
TRIGger:SBSW:SWIRe:RST ... 1846
TRIGger:SBSW:SWIRe:FRENable ... 1847
TRIGger:SBSW:SWIRe:FRAMe<fr>:ENABle ... 1847
TRIGger:SBSW:SWIRe:BIT ... 1847
TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:BIT ... 1847

TRIGger:SBSW:SWIRe:DMAX ..... 1847
TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DMAX ..... 1847
TRIGger:SBSW:SWIRe:DMIN ..... 1848
TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DMIN ..... 1848
TRIGger:SBSW:SWIRe:DOPerator ..... 1848
TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DOPerator ..... 1848
TRIGger:SBSW:SWIRe:FIENable ..... 1848
TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:ENABle ..... 1848
TRIGger:SBSW:SWIRe:ERENable ..... 1849
TRIGger:SBSW:SWIRe:ERRor<m>:ENABle ..... 1849
SBUS<sb>:SWIRe:SWTindex? ..... 1849
SBUS<sb>:SWIRe:SWTTime? ..... 1849

### TRIGger:SBSW:SWIRe:CHKall

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 790

### TRIGger:SBSW:SWIRe:CLR

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 790

### TRIGger:SBSW:SWIRe:INVert

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 790

### TRIGger:SBSW:SWIRe:RST

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 790

**TRIGger:SBSW:SWIRe:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:SWIRe:FRENable?** <Enabler>
**TRIGger:SBSW:SWIRe:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 791

**TRIGger:SBSW:SWIRe:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:SWIRe:BIT?** <Bit>
**TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 791

**TRIGger:SBSW:SWIRe:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SWIRe:DMAX?** <Data>
**TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 791

---

**TRIGger:SBSW:SWIRe:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SWIRe:DMIN?** <Data>
**TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 791

---

**TRIGger:SBSW:SWIRe:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:SWIRe:DOPerator?** <Operator>
**TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 791

---

**TRIGger:SBSW:SWIRe:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:SWIRe:FIENable?** <Enabler>
**TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 791

---

**TRIGger:SBSW:SWIRe:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:SWIRe:ERENable?** <Enabler>
**TRIGger:SBSW:SWIRe:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 792

---

**SBUS<sb>:SWIRe:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**

<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.16.5 Decode results

This section describes the commands to query the results of the decoding.

SBUS<sb>:SWIRe:FCOunt? 1850
SBUS<sb>:SWIRe:FRAMe<fr>:BITRate? 1850
SBUS<sb>:SWIRe:FRAMe<fr>:DATA? 1851
SBUS<sb>:SWIRe:FRAMe<fr>:CODParity? 1851
SBUS<sb>:SWIRe:FRAMe<fr>:DATParity? 1852
SBUS<sb>:SWIRe:FRAMe<fr>:ESCParity? 1852
SBUS<sb>:SWIRe:FRAMe<fr>:FLD<fl>:STATus? 1853
SBUS<sb>:SWIRe:FRAMe<fr>:FLD<fl>:TYPE? 1853
SBUS<sb>:SWIRe:FRAMe<fr>:FLD<fl>:VALue? 1854
SBUS<sb>:SWIRe:FRAMe<fr>:FLD<fl>:FVALue? 1854
SBUS<sb>:SWIRe:FRAMe<fr>:STATe? 1854
SBUS<sb>:SWIRe:FRAMe<fr>:STARt? 1855
SBUS<sb>:SWIRe:FRAMe<fr>:STOP? 1855
SBUS<sb>:SWIRe:FRAMe<fr>:TYPE? 1856

---

**SBUS<sb>:SWIRe:FCOunt?**

Returns the number of decoded frames.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Return values:**

<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:BITRate?**

Returns the bit rate of the frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<PrimaryBitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:DATA?**

Returns the data of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameData>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:CODParity?**

If the frame has the `FCT`, `EOP`, `EEP` or `Code` field, the command returns the parity state of this field, which can take values of `OK` or `PARity`. For a description of these fields, see Section 14.15.1, "SpaceWire basics", on page 777.

If the frame has none of these fields, the command can return the states `LENGth`, `AMBiguous` or `INComplete` of the frame, as described below.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<ParityCode> OK | PARity | LENGth | AMBiguous | INComplete

**OK**
The parity check for the field covered by this parity bit passes.

**PARity**
The parity of the field is not correct.

**LENGth**
Regardless of the field, the length of the frame is not as expected, indicating an error.

**AMBiguous**
Regardless of the field, the frame is ambiguous.

**INComplete**
Regardless of the field, the frame is incomplete.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:DATParity?**

If the frame has the `Data` field, the command returns the parity state of this field, which can take values of `OK` or `PARity`. The `Data` field only appears in `Data` and `Time` frames, see Figure 14-44 and Figure 14-46.

If the frame has no `Data` field, the command can return the states `LENGth`, `AMBiguous` or `INComplete` of the frame, as described below.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<ParityData> OK | PARity | LENGth | AMBiguous | INComplete

**OK**
The parity check for the `Data` field passes.

**PARity**
The parity of the `Data` field is not correct.

**LENGth**
Regardless of the `Data` field, the length of the frame is not as expected, indicating an error

**AMBiguous**
Regardless of the `Data` field, the frame is ambiguous.

**INComplete**
Regardless of the `Data` field, the frame is incomplete.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:ESCParity?**

If the frame has the `ESC` field, the command returns the parity state of this field, which can take values of `OK` or `PARity`. See Figure 14-45.

If the frame has no `ESC` field, the command can return the states `LENGth`, `AMBiguous` or `INComplete` of the frame, as described below.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<ParityEscape> OK | PARity | LENGth | AMBiguous | INComplete

**OK**
The parity check for the ESC field passes.

**PARity**
The parity of the ESC field is not correct.

**LENGth**
Regardless of the ESC field, the length of the frame is not as expected, indicating an error.

**AMBiguous**
Regardless of the ESC field, the frame is ambiguous.

**INComplete**
Regardless of the ESC field, the frame is incomplete.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:FLD<fl>:STATus?**

Returns the overall state of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | PARity | LENGth | AMBiguous | INComplete

**OK**
The parity check for the selected field passes.

**PARity**
Parity is not as expected, indicating an error.

**LENGth**
The length of the field is not as expected, indicating an error.

**AMBiguous**
The frame is ambiguous.

**INComplete**
The frame is incomplete.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:FLD<fl>:TYPE?**

Returns the label of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:FLD<fl>:VALue?**

Returns the value of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> *RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:STATe?**

Returns the overall state of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameState> OK | PARity | LENGth | AMBiguous | INComplete

**OK**
The frame is valid.

**PARity**
Parity is not as expected, indicating an error

**LENGth**
Length of the frame is not as expected, indicating an error

**AMBiguous**
The frame is ambiguous.

**INComplete**
The frame is incomplete.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:STARt?**

Returns the start time of the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:STOP?**

Returns the end time of the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SWIRe:FRAMe<fr>:TYPE?**

Returns the type of the specified decoded frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameType> DATA | TCOD | FCT | NULL | EOP | EEP | BADX

**DATA**
Data frame

**TCOD**
Time-code frame

**FCT**
Flow control token

**NULL**
Null symbol

**EOP**
End of packet

**EEP**
Error end of packet

**BADX**
Illegal escape code

*RST: DATA

**Usage:** Query only
Asynchronous command

## 18.18.17 SPMI (option R&S MXO3-K550)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.

- Configuration....1857
- Filter....1861
- Format....1866
- Software trigger....1876
- Decode results....1881

### 18.18.17.1 Configuration

SBUS<sb>:SPMI:GSIDenable....1857
SBUS<sb>:SPMI:GIDValue....1857
SBUS<sb>:SPMI:GTCHenable....1858
SBUS<sb>:SPMI:GTWDith....1858
SBUS<sb>:SPMI:SCLK:HYSTeresis....1858
SBUS<sb>:SPMI:SCLK:SOURce....1858
SBUS<sb>:SPMI:SCLK:THReshold....1859
SBUS<sb>:SPMI:SDATa:HYSTeresis....1859
SBUS<sb>:SPMI:SDATa:SOURce....1859
SBUS<sb>:SPMI:SDATa:THReshold....1860
SBUS<sb>:SPMI:POSition....1860
SBUS<sb>:SPMI:SCALe....1860
SBUS<sb>:SPMI:SYMBols....1860
SBUS<sb>:SPMI:NEWList....1861

**SBUS<sb>:SPMI:GSIDenable** <UseGSID>

Enables the use of the group sub ID (GSID). You can set the GSID with `SBUS<sb>:SPMI:GIDValue`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<UseGSID> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Use GSID, GSID (hex)" on page 795

**SBUS<sb>:SPMI:GIDValue** <GSID>

Sets a value for the group sub index. Available, if `SBUS<sb>:SPMI:GSIDenable` is set to `ON`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<GSID> Range: 0 to 15
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Use GSID, GSID (hex)" on page 795

**SBUS<sb>:SPMI:GTCHenable** <UseGlitchFilter>

Enables the glitch filter. A glitch filter can help to filter out short-duration voltage spikes/ glitches that can occur on the communication line.

You can set the glitch filter width with `SBUS<sb>:SPMI:GTWDith`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<UseGlitchFilter> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Glitch filter, Glitch width" on page 796

**SBUS<sb>:SPMI:GTWDith** <GlitchFilterWdt>

Sets the glitch width. Any signal transitions with a duration smaller than this value will be considered a glitch and filtered out.

This is available, if `SBUS<sb>:SPMI:GTCHenable` is set to `ON`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<GlitchFilterWdt> Range: 1E-15 to 1000000000
Increment: 1E-10
*RST: 1E-09
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Glitch filter, Glitch width" on page 796

**SBUS<sb>:SPMI:SCLK:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis for the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 796

**SBUS<sb>:SPMI:SCLK:SOURce** <ClockSource>

Sets the source of the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ClockSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "SCLK" on page 795

---

**SBUS<sb>:SPMI:SCLK:THReshold** <Threshold>

Sets a user-defined threshold value for the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 796

---

**SBUS<sb>:SPMI:SDATa:HYSTeresis** <SDATA Hyst>

Sets a value for the hysteresis for the source data channel.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SDATA Hyst>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 796

---

**SBUS<sb>:SPMI:SDATa:SOURce** <DataSource>

Sets the source of the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DataSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "SDATA" on page 795

---

**SBUS<sb>:SPMI:SDATa:THReshold** <SDATA Thres>

Sets a user-defined threshold value for the source data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SDATA Thres>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 796

---

**SBUS<sb>:SPMI:POSition** <SPMI position>

Sets the vertical position of the SPMI signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SPMI position>

**Usage:** Asynchronous command

---

**SBUS<sb>:SPMI:SCALe** <SPMI scale>

Sets the vertical position of the SPMI signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SPMI scale>

**Usage:** Asynchronous command

---

**SBUS<sb>:SPMI:SYMBols** <ShowSymbols>

Activates the symbol list to be used for decoding.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ShowSymbols> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**SBUS<sb>:SPMI:NEWList** <FileName>

Loads a symbol list file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName> String parameter with path and filename.

**Usage:** Setting only
Asynchronous command

### 18.18.17.2 Filter

There are two commands for each parameter, that you can use for defining the SPMI settings.

For example, to set the *Frame type =Register 0 Write* > *Field =Data >00* value you can use one of the following commands:

- `SBUS:SPMI:FILTer:FRAMe1:FLD1:DMIN 00`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:SPMI:FILTer:DMIN "Register 0 Write", "Data", 00`
  Defines the parameter by using the frame and field name.

SBUS<sb>:SPMI:FILTer:CHKall....1862
SBUS<sb>:SPMI:FILTer:CLR....1862
SBUS<sb>:SPMI:FILTer:INVert....1862
SBUS<sb>:SPMI:FILTer:RST....1862
SBUS<sb>:SPMI:FILTer:BIT....1862
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:BIT....1862
SBUS<sb>:SPMI:FILTer:DMAX....1863
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1863
SBUS<sb>:SPMI:FILTer:DMIN....1863
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1863
SBUS<sb>:SPMI:FILTer:DOPerator....1864
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1864
SBUS<sb>:SPMI:FILTer:ERENable....1864
SBUS<sb>:SPMI:FILTer:ERRor<n>:ENABle....1864
SBUS<sb>:SPMI:FILTer:IOPerator....1864
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1864
SBUS<sb>:SPMI:FILTer:IMIN....1865
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1865
SBUS<sb>:SPMI:FILTer:IMAX....1865
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1865
SBUS<sb>:SPMI:FILTer:FRENable....1866
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:ENABle....1866
SBUS<sb>:SPMI:FILTer:FIENable....1866
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1866

**SBUS<sb>:SPMI:FILTer:CHKall**

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 798

**SBUS<sb>:SPMI:FILTer:CLR**

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 798

**SBUS<sb>:SPMI:FILTer:INVert**

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 798

**SBUS<sb>:SPMI:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 798

**SBUS<sb>:SPMI:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:SPMI:FILTer:BIT?** <Bit>
**SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 799

---

**SBUS<sb>:SPMI:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:SPMI:FILTer:DMAX?** <Data>
**SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 799

---

**SBUS<sb>:SPMI:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:SPMI:FILTer:DMIN?** <Data>
**SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 799

**SBUS<sb>:SPMI:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:SPMI:FILTer:DOPerator?** <Operator>
**SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 799

**SBUS<sb>:SPMI:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:SPMI:FILTer:ERENable?** <Enabler>
**SBUS<sb>:SPMI:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the error

**Parameters:**
<Enable> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 800

**SBUS<sb>:SPMI:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:SPMI:FILTer:IOPerator?** <Operator>
**SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 799

---

**SBUS<sb>:SPMI:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:SPMI:FILTer:IMIN?** <Data>
**SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 799

---

**SBUS<sb>:SPMI:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:SPMI:FILTer:IMAX?** <Data>
**SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

You can set the operator with `SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 799

**SBUS<sb>:SPMI:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:SPMI:FILTer:FRENable?** <Enabler>
**SBUS<sb>:SPMI:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables the filtering on SPMI frames. Only the frames that match the selected filter conditions are displayed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 799

**SBUS<sb>:SPMI:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:SPMI:FILTer:FIENable?** <Enabler>
**SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 799

### 18.18.17.3 Format

This section describes the commands to specify and query frame formats for the "SPMI" protocol.

SBUS<sb>:SPMI:FORMat:SELect ... 1867
SBUS<sb>:SPMI:FORMat:ADDFrame ... 1867
SBUS<sb>:SPMI:FORMat:LOAD ... 1867
SBUS<sb>:SPMI:FORMat:SAVE ... 1868
SBUS<sb>:SPMI:FORMat:CLR ... 1868
SBUS<sb>:SPMI:FORMat:FCOunt? ... 1868
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ADDField ... 1869
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLDCount? ... 1869
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:NAME ... 1869

SBUS<sb>:SPMI:FORMat:FRAMe<fr>:COLor ... 1869
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:TYPE ... 1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID1Field ... 1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID2Field ... 1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID3Field ... 1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID1Value ... 1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID2Value ... 1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID3Value ... 1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:NAME ... 1871
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:COLor ... 1871
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ... 1871
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:BITCount ... 1872
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:REPetition ... 1872
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:BITorder ... 1872
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FORMat ... 1873
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ... 1874
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ... 1874
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ... 1874
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:UNIT ... 1875
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:PARA ... 1875
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:PARB ... 1875

---

**SBUS<sb>:SPMI:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 800

---

**SBUS<sb>:SPMI:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 800

---

**SBUS<sb>:SPMI:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 801

---

**SBUS<sb>:SPMI:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 801

---

**SBUS<sb>:SPMI:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 801

---

**SBUS<sb>:SPMI:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 801

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 802

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 802

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.
Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 802

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 802

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 802

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 802

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 803

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 803

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<FirstBit> Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 803

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 803

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 804

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 804

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 804

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 805

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value>
Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 805

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field
<p> Index of the value

**Parameters:**
<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 805

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 805

---

**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

Available if `SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<ParameterB> Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 805

### 18.18.17.4 Software trigger

There are two commands for each parameter, that you can use for defining the SPMI settings.

For example, to set the *Frame type =Register 0 Write* > *Field =Data >00* value you can use one of the following commands:

- `TRIGger:SBSW:SPMI:FRAMe1:FLD1:DMIN 00`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:SBSW:SPMI:DMIN "Register 0 Write", "Data", 00`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:SPMI:CHKall....1876
TRIGger:SBSW:SPMI:CLR....1877
TRIGger:SBSW:SPMI:INVert....1877
TRIGger:SBSW:SPMI:RST....1877
TRIGger:SBSW:SPMI:FRENable....1877
TRIGger:SBSW:SPMI:FRAMe<fr>:ENABle....1877
TRIGger:SBSW:SPMI:BIT....1877
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:BIT....1877
TRIGger:SBSW:SPMI:DMAX....1878
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DMAX....1878
TRIGger:SBSW:SPMI:DMIN....1878
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DMIN....1878
TRIGger:SBSW:SPMI:DOPerator....1879
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DOPerator....1879
TRIGger:SBSW:SPMI:FIENable....1879
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:ENABle....1879
TRIGger:SBSW:SPMI:IMAX....1879
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IMAX....1879
TRIGger:SBSW:SPMI:IMIN....1880
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IMIN....1880
TRIGger:SBSW:SPMI:IOPerator....1880
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IOPerator....1880
TRIGger:SBSW:SPMI:ERENable....1880
TRIGger:SBSW:SPMI:ERRor<m>:ENABle....1880
SBUS<sb>:SPMI:SWTindex?....1881
SBUS<sb>:SPMI:SWTTime?....1881

---

**TRIGger:SBSW:SPMI:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 807

**TRIGger:SBSW:SPMI:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 807

**TRIGger:SBSW:SPMI:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 807

**TRIGger:SBSW:SPMI:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 807

**TRIGger:SBSW:SPMI:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:SPMI:FRENable?** <Enabler>
**TRIGger:SBSW:SPMI:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 807

**TRIGger:SBSW:SPMI:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:SPMI:BIT?** <Bit>
**TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 807

---

**TRIGger:SBSW:SPMI:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SPMI:DMAX?** <Data>
**TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 807

---

**TRIGger:SBSW:SPMI:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SPMI:DMIN?** <Data>
**TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 807

**TRIGger:SBSW:SPMI:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:SPMI:DOPerator?** <Operator>
**TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 807

**TRIGger:SBSW:SPMI:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:SPMI:FIENable?** <Enabler>
**TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 807

**TRIGger:SBSW:SPMI:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SPMI:IMAX?** <Data>
**TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 807

---

**TRIGger:SBSW:SPMI:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:SPMI:IMIN?** <Data>
**TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 807

---

**TRIGger:SBSW:SPMI:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:SPMI:IOPerator?** <Operator>
**TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 807

---

**TRIGger:SBSW:SPMI:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:SPMI:ERENable?** <Enabler>
**TRIGger:SBSW:SPMI:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 809

---

**SBUS<sb>:SPMI:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPMI:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.17.5 Decode results

SBUS<sb>:SPMI:FCOunt? ..... 1882
SBUS<sb>:SPMI:FRAMe<fr>:ADDRess? ..... 1882
SBUS<sb>:SPMI:FRAMe<fr>:BC? ..... 1882
SBUS<sb>:SPMI:FRAMe<fr>:DATA? ..... 1883
SBUS<sb>:SPMI:FRAMe<fr>:FLDCount? ..... 1883
SBUS<sb>:SPMI:FRAMe<fr>:FLD<fl>:FVALue? ..... 1883

SBUS<sb>:SPMI:FRAMe<fr>:FLD<fl>:LABel? 1884
SBUS<sb>:SPMI:FRAMe<fr>:FLD<fl>:STATe? 1884
SBUS<sb>:SPMI:FRAMe<fr>:FLD<fl>:VALue? 1884
SBUS<sb>:SPMI:FRAMe<fr>:RBRate? 1885
SBUS<sb>:SPMI:FRAMe<fr>:SA? 1885
SBUS<sb>:SPMI:FRAMe<fr>:STARt? 1885
SBUS<sb>:SPMI:FRAMe<fr>:STATus? 1885
SBUS<sb>:SPMI:FRAMe<fr>:STOP? 1886
SBUS<sb>:SPMI:FRAMe<fr>:TYPE? 1886
SBUS<sb>:SPMI:FRAMe<fr>:WBRate? 1887
SBUS<sb>:SPMI:FRAMe<fr>:SYMBol? 1887

**SBUS<sb>:SPMI:FCOunt?**

Returns the number of decoded frames in the acquisition.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SPMI:FRAMe<fr>:ADDRess?**

Returns the address of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameADDR> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SPMI:FRAMe<fr>:BC?**

Returns the BC of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameBC> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPMI:FRAMe<fr>:DATA?**

Returns the data value of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameData>

**Usage:** Query only

---

**SBUS<sb>:SPMI:FRAMe<fr>:FLDCount?**

Returns the field count of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPMI:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified byte.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SPMI:FRAMe<fr>:FLD<fl>:LABel?**

Returns the label of the fields in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SPMI:FRAMe<fr>:FLD<fl>:STATe?**

Returns the overall state of the frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | NOReponse | ACKerror | BPERror | PARerror | LENerror | ARBerror | SSCerror | CMDerror | CODerror | INComplete

*RST: OK

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SPMI:FRAMe<fr>:FLD<fl>:VALue?**

Returns the data value of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SPMI:FRAMe<fr>:RBRate?**

Returns the read bit rate of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameRBR> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

**SBUS<sb>:SPMI:FRAMe<fr>:SA?**

Returns the address of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameSA> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SPMI:FRAMe<fr>:STARt?**

Returns the start time of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

**SBUS<sb>:SPMI:FRAMe<fr>:STATus?**

Returns the overall state of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameState> OK | NOReponse | ACKerror | BPERror | PARerror | LENerror | ARBerror | SSCerror | CMDerror | CODerror | INComplete

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPMI:FRAMe<fr>:STOP?**

Returns the stop time of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPMI:FRAMe<fr>:TYPE?**

Returns the frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameType> RZWR | RWR | RRD | ERWR | ERRD | ERWL | ERRL | MAWR | MARD | BMRD | BSRD | TBOW | REST | SLEP | SHUT | WAK | AUTH | INV | UNKN | ARB

`RZWR`: register 0 write
`RWR`: register write
`RRD`: register read
`ERWR`: extended register write
`ERRD`: extended register read
`ERWL`: extended register write long
`ERRL`: extended register read long
`MAWR`: main write
`MARD`: main read

BMRD: device descriptor block main read
BSRD: device descriptor block sub read
TBOW: transfer bus ownership
REST: reset
SLEP: sleep
SHUT: shutdown
WAK: wakeup
AUTH: authenticate
INV: invalid
UNKN: unknown
ARB: arbitration frame

*RST: RZWR

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:SPMI:FRAMe<fr>:WBRate?**

Returns the write bit rate of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameWBR> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**SBUS<sb>:SPMI:FRAMe<fr>:SYMBol?**

Returns the symbolic label of the specified frame if the label list is enabled.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<SymbolicName> String with symbolic name of the address
String with the symbolic name of the frame

**Usage:** Query only
Asynchronous command

### 18.18.18 RFFE (option R&S MXO3-K550)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.

### 18.18.18.1 Configuration

SBUS<sb>:RFFE:CLOCk:HYSTeresis....1888
SBUS<sb>:RFFE:CLOCk:SOURce....1888
SBUS<sb>:RFFE:CLOCk:THReshold....1888
SBUS<sb>:RFFE:DATA:HYSTeresis....1889
SBUS<sb>:RFFE:DATA:SOURce....1889
SBUS<sb>:RFFE:DATA:THReshold....1889
SBUS<sb>:RFFE:GFILter....1890
SBUS<sb>:RFFE:GFWidth....1890
SBUS<sb>:RFFE:POSition....1890
SBUS<sb>:RFFE:RDMD....1890
SBUS<sb>:RFFE:SCALe....1891

**SBUS<sb>:RFFE:CLOCk:HYSTeresis** <Hysteresis>

Sets a hysteresis for the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 815

**SBUS<sb>:RFFE:CLOCk:SOURce** <SourceClock>

Sets the source of the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SourceClock> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "SCLK" on page 815

**SBUS<sb>:RFFE:CLOCk:THReshold** <Threshold>

Sets a threshold value for the clock line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 815

---

**SBUS<sb>:RFFE:DATA:HYSTeresis** <DATA Hyst>

Sets a hysteresis for the data.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DATA Hyst>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 815

---

**SBUS<sb>:RFFE:DATA:SOURce** <SourceData>

Sets the source of the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<SourceData> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "SDATA" on page 815

---

**SBUS<sb>:RFFE:DATA:THReshold** <DATA Threshold>

Sets a threshold value for the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DATA Threshold>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 815

**SBUS<sb>:RFFE:GFILter** <GlitchFilter>

Enables the glitch filter on the SCLK and SDATA lines to improve decode accuracy.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<GlitchFilter> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Use glitch filter, Glitch filter width" on page 815

**SBUS<sb>:RFFE:GFWidth** <GlitchFilterWdt>

Sets the maximum glitch width to be ignored.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<GlitchFilterWdt> Range: 1E-11 to 1E-08
Increment: 1E-11
*RST: 1E-09
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Use glitch filter, Glitch filter width" on page 815

**SBUS<sb>:RFFE:POSition** <RFFE position>

Sets the vertical position of the RFFE signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<RFFE position>

**Usage:** Asynchronous command

**SBUS<sb>:RFFE:RDMD** <ReadMode>

Selects, if the standard or synchronous read mode is used.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**

<ReadMode> STRD | SREAD

`STRD`: standard

`SREAD`: synchronous read

*RST: STRD

**Usage:** Asynchronous command

**Manual operation:** See "Read mode" on page 815

---

**SBUS<sb>:RFFE:SCALe** <RFFE scale>

Set the vertical scale of the indicated RFFE signal.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<RFFE scale>

**Usage:** Asynchronous command

### 18.18.18.2 Filter

There are two commands for each parameter, that you can use for defining the RFFE settings.

For example, to set the *Frame type =Register 0 Write > Field =Data >01100* value you can use one of the following commands:

- `SBUS:RFFE:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:RFFE:FILTer:DMIN "Register 0 Write", "Data", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:RFFE:FILTer:CHKall ... 1892
SBUS<sb>:RFFE:FILTer:CLR ... 1892
SBUS<sb>:RFFE:FILTer:INVert ... 1892
SBUS<sb>:RFFE:FILTer:RST ... 1892
SBUS<sb>:RFFE:FILTer:FRENable ... 1893
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:ENABle ... 1893
SBUS<sb>:RFFE:FILTer:BIT ... 1893
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1893
SBUS<sb>:RFFE:FILTer:DMAX ... 1893
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1893
SBUS<sb>:RFFE:FILTer:DMIN ... 1894
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1894
SBUS<sb>:RFFE:FILTer:DOPerator ... 1894
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1894
SBUS<sb>:RFFE:FILTer:FIENable ... 1895
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1895

SBUS<sb>:RFFE:FILTer:IMAX ... 1895
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1895
SBUS<sb>:RFFE:FILTer:IMIN ... 1895
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1895
SBUS<sb>:RFFE:FILTer:IOPerator ... 1896
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1896
SBUS<sb>:RFFE:FILTer:ERENable ... 1896
SBUS<sb>:RFFE:FILTer:ERRor<n>:ENABle ... 1896

### SBUS<sb>:RFFE:FILTer:CHKall

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 817

### SBUS<sb>:RFFE:FILTer:CLR

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 817

### SBUS<sb>:RFFE:FILTer:INVert

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 817

### SBUS<sb>:RFFE:FILTer:RST

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 817

---

**SBUS<sb>:RFFE:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:RFFE:FILTer:FRENable?** <Enabler>
**SBUS<sb>:RFFE:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables the filtering on RFFE frames. Only the frames that match the selected filter conditions are displayed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

---

**SBUS<sb>:RFFE:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:RFFE:FILTer:BIT?** <Bit>
**SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 818

---

**SBUS<sb>:RFFE:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:RFFE:FILTer:DMAX?** <Data>
**SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 818

---

**SBUS<sb>:RFFE:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:RFFE:FILTer:DMIN?** <Data>
**SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 818

---

**SBUS<sb>:RFFE:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:RFFE:FILTer:DOPerator?** <Operator>
**SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 818

**SBUS<sb>:RFFE:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:RFFE:FILTer:FIENable?** <Enabler>
**SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 818

**SBUS<sb>:RFFE:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:RFFE:FILTer:IMAX?** <Data>
**SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

You can set the operator with `SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 818

**SBUS<sb>:RFFE:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:RFFE:FILTer:IMIN?** <Data>
**SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 818

---

**SBUS<sb>:RFFE:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:RFFE:FILTer:IOPerator?** <Operator>
**SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 818

---

**SBUS<sb>:RFFE:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:RFFE:FILTer:ERENable?** <Enabler>
**SBUS<sb>:RFFE:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 819

### 18.18.18.3 Format

This section describes the commands to specify and query frame formats for the "RFFE" protocol.

SBUS<sb>:RFFE:FORMat:SELect ... 1897
SBUS<sb>:RFFE:FORMat:ADDFrame ... 1897
SBUS<sb>:RFFE:FORMat:LOAD ... 1898
SBUS<sb>:RFFE:FORMat:SAVE ... 1898
SBUS<sb>:RFFE:FORMat:CLR ... 1898
SBUS<sb>:RFFE:FORMat:FCOunt? ... 1898
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ADDField ... 1899
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLDCount? ... 1899
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:NAME ... 1899
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:COLor ... 1900
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:TYPE ... 1900
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID1Field ... 1900
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID2Field ... 1900
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID3Field ... 1900
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID1Value ... 1901
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID2Value ... 1901
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID3Value ... 1901
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:NAME ... 1901
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:COLor ... 1901
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ... 1902
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:BITCount ... 1902
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:REPetition ... 1902
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:BITorder ... 1903
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FORMat ... 1903
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ... 1904
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ... 1904
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ... 1905
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:UNIT ... 1905
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:PARA ... 1905
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:PARB ... 1905

---

**SBUS<sb>:RFFE:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 820

---

**SBUS<sb>:RFFE:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 820

---

**SBUS<sb>:RFFE:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 821

---

**SBUS<sb>:RFFE:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 821

---

**SBUS<sb>:RFFE:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 821

---

**SBUS<sb>:RFFE:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 821

---

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 822

---

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 822

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 822

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 822

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 822

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 822

---

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 823

---

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 823

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<FirstBit> Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 823

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 823

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 824

---

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 824

---

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 824

---

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 825

---

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 825

---

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 825

---

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 825

---

**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)

- Parameter B: Offset

Available if `SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<ParameterB> Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 825

### 18.18.18.4 Software trigger

There are two commands for each parameter, that you can use for defining the RFFE settings.

For example, to set the *Frame type =Register 0 Write > Field =Data >01100* value you can use one of the following commands:

- `TRIGger:SBSW:RFFE:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:SBSW:RFFE:DMIN "Register 0 Write", "Data", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:RFFE:CHKall ..... 1907
TRIGger:SBSW:RFFE:CLR ..... 1907
TRIGger:SBSW:RFFE:INVert ..... 1907
TRIGger:SBSW:RFFE:RST ..... 1907
TRIGger:SBSW:RFFE:FRENable ..... 1907
TRIGger:SBSW:RFFE:FRAMe<fr>:ENABle ..... 1907
TRIGger:SBSW:RFFE:BIT ..... 1908
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:BIT ..... 1908
TRIGger:SBSW:RFFE:DMAX ..... 1908
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DMAX ..... 1908
TRIGger:SBSW:RFFE:DMIN ..... 1908
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DMIN ..... 1908
TRIGger:SBSW:RFFE:DOPerator ..... 1909
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DOPerator ..... 1909
TRIGger:SBSW:RFFE:FIENable ..... 1909
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:ENABle ..... 1909
TRIGger:SBSW:RFFE:IMAX ..... 1910
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IMAX ..... 1910
TRIGger:SBSW:RFFE:IMIN ..... 1910
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IMIN ..... 1910

TRIGger:SBSW:RFFE:IOPerator ..... 1910
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IOPerator ..... 1910
TRIGger:SBSW:RFFE:ERENable ..... 1911
TRIGger:SBSW:RFFE:ERRor<m>:ENABle ..... 1911
SBUS<sb>:RFFE:SWTindex? ..... 1911
SBUS<sb>:RFFE:SWTTime? ..... 1911

**TRIGger:SBSW:RFFE:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 827

**TRIGger:SBSW:RFFE:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 827

**TRIGger:SBSW:RFFE:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 827

**TRIGger:SBSW:RFFE:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 827

**TRIGger:SBSW:RFFE:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:RFFE:FRENable?** <Enabler>
**TRIGger:SBSW:RFFE:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 827

---

**TRIGger:SBSW:RFFE:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:RFFE:BIT?** <Bit>
**TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 827

---

**TRIGger:SBSW:RFFE:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:RFFE:DMAX?** <Data>
**TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 827

---

**TRIGger:SBSW:RFFE:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:RFFE:DMIN?** <Data>
**TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 827

---

**TRIGger:SBSW:RFFE:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:RFFE:DOPerator?** <Operator>
**TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 827

---

**TRIGger:SBSW:RFFE:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:RFFE:FIENable?** <Enabler>
**TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 827

**TRIGger:SBSW:RFFE:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:RFFE:IMAX?** <Data>
**TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max>
Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 827

**TRIGger:SBSW:RFFE:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:RFFE:IMIN?** <Data>
**TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min>
Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 827

**TRIGger:SBSW:RFFE:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:RFFE:IOPerator?** <Operator>
**TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 827

---

**TRIGger:SBSW:RFFE:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:RFFE:ERENable?** <Enabler>
**TRIGger:SBSW:RFFE:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 829

---

**SBUS<sb>:RFFE:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:RFFE:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**

<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.18.5 Decode results

SBUS<sb>:RFFE:FRAMe<fr>:PADone? ..... 1912
SBUS<sb>:RFFE:FRAMe<fr>:PADZero? ..... 1913
SBUS<sb>:RFFE:FRAMe<fr>:PCTRl? ..... 1913
SBUS<sb>:RFFE:FCOunt? ..... 1913
SBUS<sb>:RFFE:FRAMe<fr>:ADDRess? ..... 1913
SBUS<sb>:RFFE:FRAMe<fr>:BCOunt? ..... 1914
SBUS<sb>:RFFE:FRAMe<fr>:DATA? ..... 1914
SBUS<sb>:RFFE:FRAMe<fr>:FLDCount? ..... 1914
SBUS<sb>:RFFE:FRAMe<fr>:FLD<fl>:FVALue? ..... 1915
SBUS<sb>:RFFE:FRAMe<fr>:FLD<fl>:NAME? ..... 1915
SBUS<sb>:RFFE:FRAMe<fr>:FLD<fl>:STATe? ..... 1915
SBUS<sb>:RFFE:FRAMe<fr>:FLD<fl>:VALue? ..... 1915
SBUS<sb>:RFFE:FRAMe<fr>:RBTRate? ..... 1916
SBUS<sb>:RFFE:FRAMe<fr>:SADD? ..... 1916
SBUS<sb>:RFFE:FRAMe<fr>:STARt? ..... 1916
SBUS<sb>:RFFE:FRAMe<fr>:STATe? ..... 1917
SBUS<sb>:RFFE:FRAMe<fr>:STOP? ..... 1917
SBUS<sb>:RFFE:FRAMe<fr>:TYPE? ..... 1917
SBUS<sb>:RFFE:FRAMe<fr>:WBTRate? ..... 1919
SBUS<sb>:RFFE:NEWList ..... 1919
SBUS<sb>:RFFE:FRAMe<fr>:SYMBol? ..... 1919
SBUS<sb>:RFFE:SYMBols ..... 1919

---

**SBUS<sb>:RFFE:FRAMe<fr>:PADone?**

Returns the parity address one of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<ParityAddr1> OK | NORESPONSE | BPERR | PARity | LENGth | VERSion | SSC | GAP | INComplete
*RST: OK

**Usage:** Query only
Asynchronous command

**SBUS<sb>:RFFE:FRAMe<fr>:PADZero?**

Returns the parity address zero of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<ParityAddr0> OK | NORESPONSE | BPERR | PARity | LENGth | VERSion | SSC | GAP | INComplete

*RST: OK

**Usage:** Query only
Asynchronous command

**SBUS<sb>:RFFE:FRAMe<fr>:PCTRl?**

Returns the parity control of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<ParityCtrl> OK | NORESPONSE | BPERR | PARity | LENGth | VERSion | SSC | GAP | INComplete

*RST: OK

**Usage:** Query only
Asynchronous command

**SBUS<sb>:RFFE:FCOunt?**

Returns the number of decoded frames in the current acquisition.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:RFFE:FRAMe<fr>:ADDRess?**

Returns the register address of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Address> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:RFFE:FRAMe<fr>:BCOunt?**

Returns the byte count of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<BC> Range: 0 to 256
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:RFFE:FRAMe<fr>:DATA?**

Returns the data value of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Data>

**Usage:** Query only

---

**SBUS<sb>:RFFE:FRAMe<fr>:FLDCount?**

Returns the field count of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:RFFE:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:RFFE:FRAMe<fr>:FLD<fl>:NAME?**

Returns the name (label) of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:RFFE:FRAMe<fr>:FLD<fl>:STATe?**

Returns the overall state of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | NORESPONSE | BPERR | PARity | LENGth | VERSion | SSC | GAP | INComplete

*RST: OK

**Usage:** Query only
Asynchronous command

**SBUS<sb>:RFFE:FRAMe<fr>:FLD<fl>:VALue?**

Returns the data value of the specified field in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:RFFE:FRAMe<fr>:RBTRate?**

Returns the read bit rate of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<ReadBitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**SBUS<sb>:RFFE:FRAMe<fr>:SADD?**

Returns the address of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<SA> Range: 0 to 15
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:RFFE:FRAMe<fr>:STARt?**

Returns the start time of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<Start> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:RFFE:FRAMe<fr>:STATe?**

Returns the overall state of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<State> OK | NORESPONSE | BPERR | PARity | LENGth | VERSion | SSC | GAP | INComplete
*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:RFFE:FRAMe<fr>:STOP?**

Returns the stop time of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<Stop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:RFFE:FRAMe<fr>:TYPE?**

Returns the type of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<Type> RZWR | RWR | ERWR | ERWL | MWR | MSKW | MCTW | RRD | ERRD | ERRL | MRD | MCTR | MOHO | IRSUM | UNDEF | UNKN | ERRor

**RZWR**
Register 0 write

**RWR**
Register write

**RRD**
Register read

**ERWR**
Extended register write

**ERRD**
Extended register read

**ERWL**
Extended register write long

**ERRL**
Extended register read long

**MRD**
Main device read

**MWR**
Main device write

**MOHO**
Main device Ownership Handove

**IRSUM**
Interrupt summary and notification

**MSKW**
Masked write

**MCTR**
Main device context transfer read

**MCTW**
Main device context transfer write

**UNDEF**
Undefined frame type

**UNKN**
Unknown frame type

**ERRor**
The bits defining the command sequence are not valid, no supported command sequence

*RST: RZWR

**Usage:** Query only
Asynchronous command

**SBUS<sb>:RFFE:FRAMe<fr>:WBTRate?**

Returns the write bit rate of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<PrimaryBitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**SBUS<sb>:RFFE:NEWList** <FileName>

Loads a symbol list file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

---

**SBUS<sb>:RFFE:FRAMe<fr>:SYMBol?**

Returns the symbolic label of the specified sequence if the label list is enabled.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<SymbolicName> String with symbolic name of the frame

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:RFFE:SYMBols** <ShowSymbols>

Activates the symbol list to be used for decoding.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ShowSymbols> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

## 18.18.19 10BASE-T1S (option R&S MXO3-K560)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.


- Configuration ..... 1920
- Filter ..... 1922
- Format ..... 1928
- Software trigger ..... 1937
- Decode results ..... 1943


### 18.18.19.1 Configuration


SBUS<sb>:TNOS:POSition ..... 1920
SBUS<sb>:TNOS:SCALe ..... 1920
SBUS<sb>:TNOS:SOURce ..... 1921
SBUS<sb>:TNOS:SYMBols ..... 1921
SBUS<sb>:TNOS:THReshold:LOWer ..... 1921
SBUS<sb>:TNOS:THReshold:HYSTeresis ..... 1921
SBUS<sb>:TNOS:THReshold:UPPer ..... 1922
SBUS<sb>:TNOS:NEWList ..... 1922


---

**SBUS<sb>:TNOS:POSition** <TENBTO position>

Sets the vertical position of the 10BASE-T1S signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<TENBTO position>

**Usage:** Asynchronous command

---

**SBUS<sb>:TNOS:SCALe** <TENBTO scale>

Set the vertical scale of the indicated 10BASE-T1S signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<TENBTO scale>

**Usage:** Asynchronous command

**SBUS<sb>:TNOS:SOURce** <Source>

Selects the source channel for the 10BASE-T1S protocol. Only analog channels can be used.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Source> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 834

**SBUS<sb>:TNOS:SYMBols** <ShowSymbols>

Activates the symbol list to be used for decoding.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<ShowSymbols> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**SBUS<sb>:TNOS:THReshold:LOWer** <Lower Thres>

Sets a lower threshold.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Lower Thres>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 835

**SBUS<sb>:TNOS:THReshold:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 835

**SBUS<sb>:TNOS:THReshold:UPPer** <Upper Thres>

Sets an upper threshold.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Upper Thres>

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 835

**SBUS<sb>:TNOS:NEWList** <FileName>

Loads a symbol list file.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName> String parameter with path and file name.

**Example:**
```
SBUS1:TNOS:NEWList
'/home/storage/userData/Protocol/TNOS.csv'
SBUS1:TNOS:SYMBols ON
```

**Usage:** Setting only
Asynchronous command

### 18.18.19.2 Filter

There are two commands for each parameter, that you can use for defining the 10BASE-T1S settings.

For example, to set the *Frame type =MAC > Field =Data >00* value you can use one of the following commands:

- `SBUS:TNOS:FILTer:FRAMe1:FLD1:DMIN 00`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:TNOS:FILTer:DMIN "MAC", "Data", 00`
  Defines the parameter by using the frame and field name.

SBUS<sb>:TNOS:FILTer:CHKall....1923
SBUS<sb>:TNOS:FILTer:CLR....1923
SBUS<sb>:TNOS:FILTer:RST....1923
SBUS<sb>:TNOS:FILTer:INVert....1924
SBUS<sb>:TNOS:FILTer:BIT....1924
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:BIT....1924
SBUS<sb>:TNOS:FILTer:DMAX....1924
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1924
SBUS<sb>:TNOS:FILTer:DMIN....1925

SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DMIN ..... 1925
SBUS<sb>:TNOS:FILTer:DOPerator ..... 1925
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ..... 1925
SBUS<sb>:TNOS:FILTer:ERENable ..... 1925
SBUS<sb>:TNOS:FILTer:ERRor<n>:ENABle ..... 1925
SBUS<sb>:TNOS:FILTer:FIENable ..... 1926
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:ENABle ..... 1926
SBUS<sb>:TNOS:FILTer:IMAX ..... 1926
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IMAX ..... 1926
SBUS<sb>:TNOS:FILTer:IMIN ..... 1926
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IMIN ..... 1926
SBUS<sb>:TNOS:FILTer:IOPerator ..... 1927
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ..... 1927
SBUS<sb>:TNOS:FILTer:FRENable ..... 1927
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:ENABle ..... 1927

### SBUS<sb>:TNOS:FILTer:CHKall

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 838

### SBUS<sb>:TNOS:FILTer:CLR

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 838

### SBUS<sb>:TNOS:FILTer:RST

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 838

**SBUS<sb>:TNOS:FILTer:INVert**

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 838

**SBUS<sb>:TNOS:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:TNOS:FILTer:BIT?** <Bit>
**SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 839

**SBUS<sb>:TNOS:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:TNOS:FILTer:DMAX?** <Data>
**SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 839

**SBUS<sb>:TNOS:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:TNOS:FILTer:DMIN?** <Data>
**SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 839

**SBUS<sb>:TNOS:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:TNOS:FILTer:DOPerator?** <Operator>
**SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 839

**SBUS<sb>:TNOS:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:TNOS:FILTer:ERENable?** <Enabler>
**SBUS<sb>:TNOS:FILTer:ERRor<n>:ENABle** <Enable>

rcset

**Suffix:**
<sb> 1..4

<n> *

**Parameters:**
<Enable> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 840

---

**SBUS<sb>:TNOS:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:TNOS:FILTer:FIENable?** <Enabler>
**SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 839

---

**SBUS<sb>:TNOS:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:TNOS:FILTer:IMAX?** <Data>
**SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

You can set the operator with `SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 839

---

**SBUS<sb>:TNOS:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:TNOS:FILTer:IMIN?** <Data>
**SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 839

---

**SBUS<sb>:TNOS:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:TNOS:FILTer:IOPerator?** <Operator>
**SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 839

---

**SBUS<sb>:TNOS:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:TNOS:FILTer:FRENable?** <Enabler>
**SBUS<sb>:TNOS:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables the filtering on 10Base-T1S frames. Only the frames that match the selected filter conditions are displayed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 839

### 18.18.19.3 Format

This section describes the commands to specify and query frame formats for the "10BASE-T1S" protocol.

SBUS<sb>:TNOS:FORMat:SELect....1928
SBUS<sb>:TNOS:FORMat:ADDFrame....1929
SBUS<sb>:TNOS:FORMat:LOAD....1929
SBUS<sb>:TNOS:FORMat:SAVE....1929
SBUS<sb>:TNOS:FORMat:CLR....1929
SBUS<sb>:TNOS:FORMat:FCOunt?....1930
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ADDField....1930
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLDCount?....1930
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:NAME....1930
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:COLor....1931
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:TYPE....1931
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID1Field....1931
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID2Field....1931
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID3Field....1931
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID1Value....1932
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID2Value....1932
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID3Value....1932
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:NAME....1932
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:COLor....1932
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1933
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1933
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1933
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1934
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1934
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1935
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1935
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1936
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1936
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:PARA....1936
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:PARB....1936

---

**SBUS<sb>:TNOS:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 840

**SBUS<sb>:TNOS:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 841

**SBUS<sb>:TNOS:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 841

**SBUS<sb>:TNOS:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 842

**SBUS<sb>:TNOS:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 841

**SBUS<sb>:TNOS:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 841

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 843

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 842

---

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 842

---

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 842

---

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 842

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 842

---

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 843

---

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 843

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<FirstBit> Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 843

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 844

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 844

---

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 844

---

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 844

---

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 845

---

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 845

---

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 845

---

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 845

---

**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)

- Parameter B: Offset

Available if `SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**

| | |
|---|---|
| <sb> | 1 to 4, index of the serial bus |
| <fr> | Index of the frame |
| <fl> | Index of the field |

**Parameters:**

| | | |
|---|---|---|
| <ParameterB> | Range: | -1E+20 to 1E+20 |
| | Increment: | 1E-12 |
| | *RST: | 0 |

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 845

### 18.18.19.4 Software trigger

There are two commands for each parameter, that you can use for defining the 10BASE-T1S settings.

For example, to set the *Frame type =MAC > Field =Data >00* value you can use one of the following commands:

- `TRIGger:SBSW:TNOS:FRAMe1:FLD1:DMIN 00`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:SBSW:TNOS:DMIN "MAC", "Data", 00`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:TNOS:CHKall....1938
TRIGger:SBSW:TNOS:CLR....1938
TRIGger:SBSW:TNOS:INVert....1938
TRIGger:SBSW:TNOS:RST....1938
TRIGger:SBSW:TNOS:FRENable....1938
TRIGger:SBSW:TNOS:FRAMe<fr>:ENABle....1938
TRIGger:SBSW:TNOS:BIT....1939
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:BIT....1939
TRIGger:SBSW:TNOS:DMAX....1939
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DMAX....1939
TRIGger:SBSW:TNOS:DMIN....1939
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DMIN....1939
TRIGger:SBSW:TNOS:DOPerator....1940
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DOPerator....1940
TRIGger:SBSW:TNOS:FIENable....1940
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:ENABle....1940
TRIGger:SBSW:TNOS:IMAX....1941
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IMAX....1941
TRIGger:SBSW:TNOS:IMIN....1941
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IMIN....1941

TRIGger:SBSW:TNOS:IOPerator....1941
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IOPerator....1941
TRIGger:SBSW:TNOS:ERENable....1942
TRIGger:SBSW:TNOS:ERRor<m>:ENABle....1942
SBUS<sb>:TNOS:SWTindex?....1942
SBUS<sb>:TNOS:SWTTime?....1942

**TRIGger:SBSW:TNOS:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 846

**TRIGger:SBSW:TNOS:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 846

**TRIGger:SBSW:TNOS:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 846

**TRIGger:SBSW:TNOS:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 846

**TRIGger:SBSW:TNOS:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:TNOS:FRENable?** <Enabler>
**TRIGger:SBSW:TNOS:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 847

---

**TRIGger:SBSW:TNOS:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:TNOS:BIT?** <Bit>
**TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 847

---

**TRIGger:SBSW:TNOS:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:TNOS:DMAX?** <Data>
**TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DOPerator` on page 1940.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 847

---

**TRIGger:SBSW:TNOS:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:TNOS:DMIN?** <Data>
**TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min>

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 847

---

**TRIGger:SBSW:TNOS:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:TNOS:DOPerator?** <Operator>
**TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 847

---

**TRIGger:SBSW:TNOS:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:TNOS:FIENable?** <Enabler>
**TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 847

**TRIGger:SBSW:TNOS:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:TNOS:IMAX?** <Data>
**TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max>
Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 847

**TRIGger:SBSW:TNOS:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:TNOS:IMIN?** <Data>
**TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min>
Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 847

**TRIGger:SBSW:TNOS:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:TNOS:IOPerator?** <Operator>
**TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 847

---

**TRIGger:SBSW:TNOS:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:TNOS:ERENable?** <Enabler>
**TRIGger:SBSW:TNOS:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 848

---

**SBUS<sb>:TNOS:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**

<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.19.5 Decode results

SBUS<sb>:TNOS:FCOunt? 1943
SBUS<sb>:TNOS:FRAMe<fr>:FLDCount? 1943
SBUS<sb>:TNOS:FRAMe<fr>:BITRate? 1944
SBUS<sb>:TNOS:FRAMe<fr>:FLD<fl>:FVALue? 1944
SBUS<sb>:TNOS:FRAMe<fr>:FLD<fl>:LABel? 1944
SBUS<sb>:TNOS:FRAMe<fr>:FLD<fl>:STATe? 1945
SBUS<sb>:TNOS:FRAMe<fr>:FLD<fl>:VALue? 1945
SBUS<sb>:TNOS:FRAMe<fr>:CRC? 1946
SBUS<sb>:TNOS:FRAMe<fr>:DATA? 1946
SBUS<sb>:TNOS:FRAMe<fr>:DTADdress? 1946
SBUS<sb>:TNOS:FRAMe<fr>:DTSYmbol? 1947
SBUS<sb>:TNOS:FRAMe<fr>:FCRC? 1947
SBUS<sb>:TNOS:FRAMe<fr>:FDATa? 1947
SBUS<sb>:TNOS:FRAMe<fr>:FDTaddress? 1947
SBUS<sb>:TNOS:FRAMe<fr>:FSRaddress? 1948
SBUS<sb>:TNOS:FRAMe<fr>:FTPLenth? 1948
SBUS<sb>:TNOS:FRAMe<fr>:SRADdress? 1948
SBUS<sb>:TNOS:FRAMe<fr>:SRSYmbol? 1949
SBUS<sb>:TNOS:FRAMe<fr>:STARt? 1949
SBUS<sb>:TNOS:FRAMe<fr>:STATus? 1949
SBUS<sb>:TNOS:FRAMe<fr>:STOP? 1950
SBUS<sb>:TNOS:FRAMe<fr>:TPLenth? 1950
SBUS<sb>:TNOS:FRAMe<fr>:TYPE? 1951

---

**SBUS<sb>:TNOS:FCOunt?**

Returns the number of decoded frames.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Return values:**

<Count> Total number of decoded frames.

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:FLDCount?**

Returns the number of fields in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count> Total number of fields.

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:BITRate?**

Returns the primary bit rate.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<PrimaryBitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:FLD<fl>:LABel?**

Returns the label of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:FLD<fl>:STATe?**

Returns the state of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | EPRMble | ESFD | EESD | ECRC | ELEN | INComplete

**OK**
The field has no error, it is valid.

**EPRMble**
Preamble error, the hex value of the preamble field is different from 0x55555555555555 (56 alternating 0's and 1's).

**ESFD**
SFD error, the hex value of the SFD field is different from 0xD5

**EESD**
ESD error, the value of the ESD field does not correspond to the symbol pair "ESD, ESDOK"

**ECRC**
CRC error, the value of the FCS field does not match the calculated CRC.

**ELEN**
Length error, the number of bits in the specified field is higher or lower than expected.

**INComplete**
The frame is incomplete

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:FLD<fl>:VALue?**

Returns the value of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:CRC?**

Returns the CRC for the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<CRC> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:DATA?**

Returns the data for the specified frame in comma-separated values.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

<Values>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:DTADdress?**

Returns the destination address for the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<DestAddress> Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:DTSYmbol?**

Returns the destination symbol for the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<DestSymbNme>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:FCRC?**

Returns the formatted CRC for the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedCRC>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:FDATa?**

Returns the formatted data for the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedData>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:FDTaddress?**

Returns the formatted destination address of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedDestAddr>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:FSRaddress?**

Returns the formatted SRC address for the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedSrcAddr>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:FTPLenth?**

Returns the formatted type length for the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedTypLenth>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:SRADdress?**

Returns the source address for the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<SourceAddress> Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:TNOS:FRAMe<fr>:SRSYmbol?**

Returns the frame of the source symbol.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<SrcSymbNme>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:TNOS:FRAMe<fr>:STARt?**

Returns the start time of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

**SBUS<sb>:TNOS:FRAMe<fr>:STATus?**

Returns the state of the frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameState> OK | EPRMble | ESFD | EESD | ECRC | ELEN | INComplete

**OK**
The frame has no error, it is valid.

**EPRMble**
Preamble error, the hex value of the preamble field is different from 0x55555555555555 (56 alternating 0's and 1's).

**ESFD**
SFD error, the hex value of the SFD field is different from 0xD5

**EESD**
ESD error, the value of the ESD field does not correspond to the symbol pair "ESD, ESDOK"

**ECRC**
CRC error, the value of the FCS field does not match the calculated CRC.

**ELEN**
Length error, the number of bits in the specified frame is higher or lower than expected.

**INComplete**
The frame is incomplete

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:STOP?**

Returns the stop time of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:TNOS:FRAMe<fr>:TPLenth?**

Returns the type length for the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<TypeLenth> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:TNOS:FRAMe<fr>:TYPE?**

Returns the type of the frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameType> MAC | COMMit | BEACon | UNKN

*RST: MAC

**Usage:** Query only
Asynchronous command

## 18.18.20 USB 1.1 (option R&S MXO3-K570)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.

- Configuration....1951
- Filter....1954
- Format....1959
- Software trigger....1969
- Decode results....1974

### 18.18.20.1 Configuration

SBUS<sb>:USBone:DPLus:SOURce....1951
SBUS<sb>:USBone:DMINus:SOURce....1952
SBUS<sb>:USBone:MODE....1952
SBUS<sb>:USBone:DPLus:THReshold....1952
SBUS<sb>:USBone:DPLus:HYSTeresis....1953
SBUS<sb>:USBone:DMINus:THReshold....1953
SBUS<sb>:USBone:DMINus:HYSTeresis....1953
SBUS<sb>:USBone:POSition....1953
SBUS<sb>:USBone:SCALe....1954

**SBUS<sb>:USBone:DPLus:SOURce** <DPlusSource>

Sets the source channel to which the D+ data line is connected.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**

<DPlusSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "D+ source, D- source" on page 857

---

**SBUS<sb>:USBone:DMINus:SOURce** <DMinusSource>

Sets the source channel to which the D- data line is connected.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<DMinusSource> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "D+ source, D- source" on page 857

---

**SBUS<sb>:USBone:MODE** <Mode>

Selects the low speed or full speed variant of the USB 1.1 protocol.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<Mode> USBLow | USBFull

*RST: USBLow

**Usage:** Asynchronous command

**Manual operation:** See "Speed variant" on page 857

---

**SBUS<sb>:USBone:DPLus:THReshold** <DPlusThreshold>

Sets a threshold value for the D+ line.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<DPlusThreshold>

**Usage:** Asynchronous command

**Manual operation:** See "D+ threshold" on page 859

**SBUS<sb>:USBone:DPLus:HYSTeresis** <DPlusHyst>

Sets a hysteresis value for the D+ line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DPlusHyst>

**Usage:** Asynchronous command

**Manual operation:** See "D+ hysteresis" on page 859

**SBUS<sb>:USBone:DMINus:THReshold** <DMinusThres>

Sets a threshold value for the D- line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DMinusThres>

**Usage:** Asynchronous command

**Manual operation:** See "D- threshold" on page 859

**SBUS<sb>:USBone:DMINus:HYSTeresis** <DMinusHyst>

Sets a hysteresis value for the D- line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<DMinusHyst>

**Usage:** Asynchronous command

**Manual operation:** See "D- hysteresis" on page 859

**SBUS<sb>:USBone:POSition** <Position>

Sets the vertical position of the USB 1.1 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Position> Range: 1 to 100

**Usage:** Asynchronous command

**SBUS<sb>:USBone:SCALe** <Scale>

Sets the vertical scale of the USB 1.1 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Scale> Range: 1 to 5

**Usage:** Asynchronous command

### 18.18.20.2 Filter

There are two commands for each parameter, that you can use for defining the USB 1.1 settings.

For example, to set the *Frame type =WRITe > Field =Address >Data* value you can use one of the following commands:

- `SBUS:USBone:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:USBone:FILTer:DMIN "WRITe", "Address", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:USBone:FILTer:CHKall ... 1955
SBUS<sb>:USBone:FILTer:CLR ... 1955
SBUS<sb>:USBone:FILTer:INVert ... 1955
SBUS<sb>:USBone:FILTer:RST ... 1955
SBUS<sb>:USBone:FILTer:FRENable ... 1955
SBUS<sb>:USBone:FILTer:FRAMe<fr>:ENABle ... 1955
SBUS<sb>:USBone:FILTer:DMAX ... 1956
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1956
SBUS<sb>:USBone:FILTer:DMIN ... 1956
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1956
SBUS<sb>:USBone:FILTer:DOPerator ... 1957
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1957
SBUS<sb>:USBone:FILTer:ERENable ... 1957
SBUS<sb>:USBone:FILTer:ERRor<n>:ENABle ... 1957
SBUS<sb>:USBone:FILTer:IMAX ... 1957
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1957
SBUS<sb>:USBone:FILTer:IMIN ... 1958
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1958
SBUS<sb>:USBone:FILTer:IOPerator ... 1958
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1958
SBUS<sb>:USBone:FILTer:BIT ... 1959
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1959
SBUS<sb>:USBone:FILTer:FIENable ... 1959
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1959

**SBUS<sb>:USBone:FILTer:CHKall**

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 861

**SBUS<sb>:USBone:FILTer:CLR**

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 861

**SBUS<sb>:USBone:FILTer:INVert**

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 861

**SBUS<sb>:USBone:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 861

**SBUS<sb>:USBone:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:USBone:FILTer:FRENable?** <Enabler>
**SBUS<sb>:USBone:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables or disables the specific frame to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 861

---

**SBUS<sb>:USBone:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:USBone:FILTer:DMAX?** <Data>
**SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max> List of comma-separated values

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 862

---

**SBUS<sb>:USBone:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:USBone:FILTer:DMIN?** <Data>
**SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min> List of comma-separated values

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 862

**SBUS<sb>:USBone:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:USBone:FILTer:DOPerator?** <Operator>
**SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 862

**SBUS<sb>:USBone:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:USBone:FILTer:ERENable?** <Enabler>
**SBUS<sb>:USBone:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the error

**Parameters:**
<Enable> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 863

**SBUS<sb>:USBone:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:USBone:FILTer:IMAX?** <Data>
**SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

You can set the operator with `SBUS<sb>:USBone:FILTer:IOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 862

---

**SBUS<sb>:USBone:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:USBone:FILTer:IMIN?** <Data>
**SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 862

---

**SBUS<sb>:USBone:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:USBone:FILTer:IOPerator?** <Operator>
**SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 862

**SBUS<sb>:USBone:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:USBone:FILTer:BIT?** <Bit>
**SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 862

**SBUS<sb>:USBone:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:USBone:FILTer:FIENable?** <Enabler>
**SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 862

### 18.18.20.3 Format

This section describes the commands to specify and query frame formats for the "USB1.1" protocol.

SBUS<sb>:USBone:FORMat:SELect ..... 1960
SBUS<sb>:USBone:FORMat:ADDFrame ..... 1960
SBUS<sb>:USBone:FORMat:LOAD ..... 1961
SBUS<sb>:USBone:FORMat:SAVE ..... 1961
SBUS<sb>:USBone:FORMat:CLR ..... 1961
SBUS<sb>:USBone:FORMat:FCOunt? ..... 1961
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ADDField ..... 1962
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLDCount? ..... 1962

SBUS<sb>:USBone:FORMat:FRAMe<fr>:NAME ..... 1962
SBUS<sb>:USBone:FORMat:FRAMe<fr>:COLor ..... 1962
SBUS<sb>:USBone:FORMat:FRAMe<fr>:TYPE ..... 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID1Field ..... 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID2Field ..... 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID3Field ..... 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID1Value ..... 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID2Value ..... 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID3Value ..... 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:NAME ..... 1964
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:COLor ..... 1964
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ..... 1964
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:BITCount ..... 1965
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:REPetition ..... 1965
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:BITorder ..... 1965
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FORMat ..... 1966
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ..... 1967
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ..... 1967
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ..... 1967
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:UNIT ..... 1968
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:PARA ..... 1968
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:PARB ..... 1968

**SBUS<sb>:USBone:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 863

**SBUS<sb>:USBone:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 863

**SBUS<sb>:USBone:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 864

**SBUS<sb>:USBone:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 864

**SBUS<sb>:USBone:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 864

**SBUS<sb>:USBone:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 864

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 865

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 865

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**

<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 865

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**

<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 865

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**

<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 865

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 865

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 866

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 866

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<FirstBit> Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 866

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 866

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 867

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 867

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 867

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 868

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 868

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if `SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field
<p> Index of the value

**Parameters:**
<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 868

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if `SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 868

---

**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

Available if `SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `FLIN`.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<ParameterB> Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 868

### 18.18.20.4 Software trigger

There are two commands for each parameter, that you can use for defining the USB 1.1 settings.

For example, to set the *Frame type =WRITe > Field =Address >Data* value you can use one of the following commands:

- `TRIGger:SBSW:USBone:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `STRIGger:SBSW:USBone:DMIN "WRITe", "Address", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:USBone:CHKall....1969
TRIGger:SBSW:USBone:CLR....1970
TRIGger:SBSW:USBone:INVert....1970
TRIGger:SBSW:USBone:RST....1970
TRIGger:SBSW:USBone:FRENable....1970
TRIGger:SBSW:USBone:FRAMe<fr>:ENABle....1970
TRIGger:SBSW:USBone:BIT....1970
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:BIT....1970
TRIGger:SBSW:USBone:DMAX....1971
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DMAX....1971
TRIGger:SBSW:USBone:DMIN....1971
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DMIN....1971
TRIGger:SBSW:USBone:DOPerator....1972
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DOPerator....1972
TRIGger:SBSW:USBone:FIENable....1972
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:ENABle....1972
TRIGger:SBSW:USBone:IMAX....1972
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IMAX....1972
TRIGger:SBSW:USBone:IMIN....1973
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IMIN....1973
TRIGger:SBSW:USBone:IOPerator....1973
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IOPerator....1973
TRIGger:SBSW:USBone:ERENable....1973
TRIGger:SBSW:USBone:ERRor<m>:ENABle....1973
SBUS<sb>:USBone:SWTindex?....1974
SBUS<sb>:USBone:SWTTime?....1974

---

**TRIGger:SBSW:USBone:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 870

**TRIGger:SBSW:USBone:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 870

**TRIGger:SBSW:USBone:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 870

**TRIGger:SBSW:USBone:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 870

**TRIGger:SBSW:USBone:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:USBone:FRENable?** <Enabler>
**TRIGger:SBSW:USBone:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 870

**TRIGger:SBSW:USBone:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:USBone:BIT?** <Bit>
**TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 871

---

**TRIGger:SBSW:USBone:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:USBone:DMAX?** <Data>
**TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max> List of comma-separated values

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 871

---

**TRIGger:SBSW:USBone:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:USBone:DMIN?** <Data>
**TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min> List of comma-separated values

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 871

**TRIGger:SBSW:USBone:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:USBone:DOPerator?** <Operator>
**TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange
*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 871

**TRIGger:SBSW:USBone:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:USBone:FIENable?** <Enabler>
**TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 871

**TRIGger:SBSW:USBone:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:USBone:IMAX?** <Data>
**TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 871

---

**TRIGger:SBSW:USBone:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:USBone:IMIN?** <Data>
**TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 871

---

**TRIGger:SBSW:USBone:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:USBone:IOPerator?** <Operator>
**TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 871

---

**TRIGger:SBSW:USBone:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:USBone:ERENable?** <Enabler>
**TRIGger:SBSW:USBone:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 872

---

**SBUS<sb>:USBone:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.20.5 Decode results

SBUS<sb>:USBone:FCOunt? 1975
SBUS<sb>:USBone:FRAMe<fr>:ADDRess? 1975
SBUS<sb>:USBone:FRAMe<fr>:FLDCount? 1975
SBUS<sb>:USBone:FRAMe<fr>:BITRate? 1976
SBUS<sb>:USBone:FRAMe<fr>:DATA? 1976
SBUS<sb>:USBone:FRAMe<fr>:STARt? 1976

SBUS<sb>:USBone:FRAMe<fr>:STATus? 1977
SBUS<sb>:USBone:FRAMe<fr>:STOP? 1977
SBUS<sb>:USBone:FRAMe<fr>:FLD<fl>:FVALue? 1978
SBUS<sb>:USBone:FRAMe<fr>:FLD<fl>:LABel? 1978
SBUS<sb>:USBone:FRAMe<fr>:FLD<fl>:STATe? 1978
SBUS<sb>:USBone:FRAMe<fr>:FLD<fl>:VALue? 1979
SBUS<sb>:USBone:FRAMe<fr>:CRC? 1979
SBUS<sb>:USBone:FRAMe<fr>:FADDress? 1980
SBUS<sb>:USBone:FRAMe<fr>:FCRC? 1980
SBUS<sb>:USBone:FRAMe<fr>:ENDPoint? 1980
SBUS<sb>:USBone:FRAMe<fr>:FRNumber? 1981
SBUS<sb>:USBone:FRAMe<fr>:PID? 1981
SBUS<sb>:USBone:FRAMe<fr>:PIDName? 1981
SBUS<sb>:USBone:FRAMe<fr>:TYPE? 1981

### SBUS<sb>:USBone:FCOunt?

Returns the number of decoded frames.

**Suffix:**
<sb> 1 to 4, index of the serial bus
Selects the serial bus.

**Return values:**
<Count> Total number of decoded frames.

**Usage:** Query only
Asynchronous command

### SBUS<sb>:USBone:FRAMe<fr>:ADDRess?

Returns the device address value of the indicated frame. That is, the address value that is shown in the decoded cells and in the decode results table.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<AddressValue> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

### SBUS<sb>:USBone:FRAMe<fr>:FLDCount?

Returns the number of fields in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:BITRate?**

Returns the number of fields in the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<BitRate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:DATA?**

Returns a comma-separated list of values, where the first value is the number of data bytes in the frame, and the remaining values are this number of data bytes.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

<Values>

**Example:**
```
SBUS:USB:FRAMe2:DATA?
13,7,7,8,9,16,17,18,19,20,21,21,22,255
```
The first value, here 13, is the number of data bytes to follow.

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:STARt?**

Returns the start time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:STATus?**

Returns the overall state of the frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameState> OK | PID | CRC | LENGth | BTST | INComplete

**OK**
The frame is valid.

**PID**
A packet identifier error signifies that the integrity check has failed, see Packet structure > "PID" in Section 14.19.1, "USB protocol basics", on page 851. The 8-bit PID is invalid, if its 4 least significant bits (LSB) are not the complement of its 4 most significant bits (MSB).

**CRC**
Cyclic redundancy check failed.

**LENGth**
Wrong length of frame.

**BTST**
Bit stuffing error occurred.

**INComplete**
The frame is not completely contained in the acquisition. The acquired part of the frame is usually valid, but cannot always be trusted.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:STOP?**

Returns the end time of the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:FLD<fl>:LABel?**

Returns the name (label) of the specified field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:FLD<fl>:STATe?**

Returns the overall state of the specified field for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | PID | CRC | LENGth | BTST | INComplete

**OK**
The field has no error, it is valid.

**PID**
A packet identifier error signifies that the integrity check has failed, see Packet structure > "PID" in Section 14.19.1, "USB protocol basics", on page 851. The 8-bit PID is invalid, if its 4 least significant bits (LSB) are not the complement of its 4 most significant bits (MSB).

**CRC**
Cyclic redundancy check failed.

**LENGth**
Wrong length of the field.

**BTST**
Bit stuffing error occurred.

**INComplete**
The field is not completely contained in the acquisition. The acquired part of the field is usually valid, but cannot always be trusted.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:FLD<fl>:VALue?**

Returns the data value of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:CRC?**

Returns the cyclic redundancy check (CRC) checksum of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<CRCValue> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:FADDress?**

Returns the formatted address for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedAddrVal>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:FCRC?**

Returns the formatted CRC value for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedCRCVal>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBone:FRAMe<fr>:ENDPoint?**

Returns the endpoint for the selected serial bus and frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Endpoint> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBone:FRAMe<fr>:FRNumber?**

Returns the frame number for the selected serial bus and frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameNumber> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBone:FRAMe<fr>:PID?**

Returns the packet identifier for the selected serial bus and frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Command> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBone:FRAMe<fr>:PIDName?**

Returns the name of the packet identifier for the selected serial bus and frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedCmd>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBone:FRAMe<fr>:TYPE?**

Returns the frame type of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<FrameType> UNKNown | DATA | TOKen | SOF | HSHK | PERR

Unknown, data, token, start of frame, handshake, PRE/ERR frame (with PID=0x3C meaning either preamble or error)

*RST: UNKNown

**Usage:** Query only
Asynchronous command

## 18.18.21 USB 2.0 (option R&S MXO3-K570)

Before MXO 3 can accept the specific bus commands, make sure that:

- The decoding of the specified bus is enabled with `SBUS<sb>[:STATe]`.
- The bus is set to the correct type with `SBUS<sb>:TYPE`.


- Configuration....1982
- Filter....1984
- Format....1989
- Software trigger....1999
- Decode results....2005


### 18.18.21.1 Configuration


SBUS<sb>:USBHspeed:SOURce....1982
SBUS<sb>:USBHspeed:THReshold:UPPer....1983
SBUS<sb>:USBHspeed:THReshold:LOWer....1983
SBUS<sb>:USBHspeed:THReshold:HYSTeresis....1983
SBUS<sb>:USBHspeed:POSition....1983
SBUS<sb>:USBHspeed:SCALe....1984


---

**SBUS<sb>:USBHspeed:SOURce** <Source>

Selects the differential signal source to which the data line is connected.

**Suffix:**

<sb> 1 to 4, index of the serial bus

**Parameters:**

<Source> C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8

**Usage:** Asynchronous command

**Manual operation:** See "Differential source" on page 875

**SBUS<sb>:USBHspeed:THReshold:UPPer** <Upper Thres>

Sets the upper threshold value for the differential signal on the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Upper Thres>

**Usage:** Asynchronous command

**Manual operation:** See "Upper threshold" on page 876

**SBUS<sb>:USBHspeed:THReshold:LOWer** <Lower Thres>

Sets the lower threshold value for the differential signal on the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Lower Thres>

**Usage:** Asynchronous command

**Manual operation:** See "Lower threshold" on page 877

**SBUS<sb>:USBHspeed:THReshold:HYSTeresis** <Hysteresis>

Sets a hysteresis value for the differential signal on the data line.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Hysteresis>

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis" on page 877

**SBUS<sb>:USBHspeed:POSition** <Position>

Sets the vertical position of the USB 2.0 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Position> Range: 1 to 100

**Usage:** Asynchronous command

**SBUS<sb>:USBHspeed:SCALe** <Scale>

Sets the vertical scale of the USB 2.0 signal.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Scale> Range: 1 to 5

**Usage:** Asynchronous command

### 18.18.21.2 Filter

There are two commands for each parameter, that you can use for defining the USB 2.0 settings.

For example, to set the *Frame type =WRITe > Field =Address >Data* value you can use one of the following commands:

- `SBUS:USBHspeed:FILTer:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SBUS:USBHspeed:FILTer:DMIN "WRITe", "Address", 01100`
  Defines the parameter by using the frame and field name.

SBUS<sb>:USBHspeed:FILTer:CHKall....1985
SBUS<sb>:USBHspeed:FILTer:CLR....1985
SBUS<sb>:USBHspeed:FILTer:INVert....1985
SBUS<sb>:USBHspeed:FILTer:RST....1985
SBUS<sb>:USBHspeed:FILTer:FRENable....1985
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:ENABle....1985
SBUS<sb>:USBHspeed:FILTer:DMAX....1986
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1986
SBUS<sb>:USBHspeed:FILTer:DMIN....1986
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1986
SBUS<sb>:USBHspeed:FILTer:DOPerator....1987
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1987
SBUS<sb>:USBHspeed:FILTer:ERENable....1987
SBUS<sb>:USBHspeed:FILTer:ERRor<n>:ENABle....1987
SBUS<sb>:USBHspeed:FILTer:IMAX....1987
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1987
SBUS<sb>:USBHspeed:FILTer:IMIN....1988
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1988
SBUS<sb>:USBHspeed:FILTer:IOPerator....1988
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1988
SBUS<sb>:USBHspeed:FILTer:BIT....1989
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:BIT....1989
SBUS<sb>:USBHspeed:FILTer:FIENable....1989
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1989

**SBUS<sb>:USBHspeed:FILTer:CHKall**

Enables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 878

**SBUS<sb>:USBHspeed:FILTer:CLR**

Disables the filter for all available frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 878

**SBUS<sb>:USBHspeed:FILTer:INVert**

Inverts the current state of the frame and error types: all frames and error types that were enabled are disabled and vice versa.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 878

**SBUS<sb>:USBHspeed:FILTer:RST**

Presets the state of the selected frames and error types.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 878

**SBUS<sb>:USBHspeed:FILTer:FRENable** <Frame>,<Enabler>
**SBUS<sb>:USBHspeed:FILTer:FRENable?** <Enabler>
**SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:ENABle** <Enable>

Enables or disables the specific frame to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 879

---

**SBUS<sb>:USBHspeed:FILTer:DMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:USBHspeed:FILTer:DMAX?** <Data>
**SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max> List of comma-separated values

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 879

---

**SBUS<sb>:USBHspeed:FILTer:DMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:USBHspeed:FILTer:DMIN?** <Data>
**SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min> List of comma-separated values

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 879

**SBUS<sb>:USBHspeed:FILTer:DOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:USBHspeed:FILTer:DOPerator?** <Operator>
**SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 879

**SBUS<sb>:USBHspeed:FILTer:ERENable** <ErrorName>,<Enabler>
**SBUS<sb>:USBHspeed:FILTer:ERENable?** <Enabler>
**SBUS<sb>:USBHspeed:FILTer:ERRor<n>:ENABle** <Enable>

Defines the error type to be filtered on.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<n> Index of the error

**Parameters:**
<Enable> OFF | ON

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 880

**SBUS<sb>:USBHspeed:FILTer:IMAX** <Frame>,<Field>,<Data>
**SBUS<sb>:USBHspeed:FILTer:IMAX?** <Data>
**SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range if the operator is set to `INRange`.

You can set the operator with `SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 879

---

**SBUS<sb>:USBHspeed:FILTer:IMIN** <Frame>,<Field>,<Data>
**SBUS<sb>:USBHspeed:FILTer:IMIN?** <Data>
**SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 879

---

**SBUS<sb>:USBHspeed:FILTer:IOPerator** <Frame>,<Field>,<Operator>
**SBUS<sb>:USBHspeed:FILTer:IOPerator?** <Operator>
**SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 879

**SBUS<sb>:USBHspeed:FILTer:BIT** <Frame>,<Field>,<Bit>
**SBUS<sb>:USBHspeed:FILTer:BIT?** <Bit>
**SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 879

**SBUS<sb>:USBHspeed:FILTer:FIENable** <Frame>,<Field>,<Enabler>
**SBUS<sb>:USBHspeed:FILTer:FIENable?** <Enabler>
**SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 879

### 18.18.21.3 Format

This section describes the commands to specify and query frame formats for the "USB2.0" protocol.

SBUS<sb>:USBHspeed:FORMat:SELect ..... 1990
SBUS<sb>:USBHspeed:FORMat:ADDFrame ..... 1990
SBUS<sb>:USBHspeed:FORMat:LOAD ..... 1991
SBUS<sb>:USBHspeed:FORMat:SAVE ..... 1991
SBUS<sb>:USBHspeed:FORMat:CLR ..... 1991
SBUS<sb>:USBHspeed:FORMat:FCOunt? ..... 1991
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ADDField ..... 1992
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLDCount? ..... 1992

SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:NAME ..... 1992
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:COLor ..... 1992
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:TYPE ..... 1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID1Field ..... 1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID2Field ..... 1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID3Field ..... 1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID1Value ..... 1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID2Value ..... 1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID3Value ..... 1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:NAME ..... 1994
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:COLor ..... 1994
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ..... 1994
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:BITCount ..... 1995
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:REPetition ..... 1995
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:BITorder ..... 1995
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FORMat ..... 1996
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ..... 1997
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ..... 1997
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ..... 1997
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:UNIT ..... 1998
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:PARA ..... 1998
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:PARB ..... 1998

---

**SBUS<sb>:USBHspeed:FORMat:SELect** <Select>

Enables the format configuration

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Parameters:**
<Select> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show format" on page 881

---

**SBUS<sb>:USBHspeed:FORMat:ADDFrame**

Appendes a new frame format to the frame list.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add frame" on page 881

**SBUS<sb>:USBHspeed:FORMat:LOAD** <FileName>

Loads a the specified XML file with a list of frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Open" on page 882

**SBUS<sb>:USBHspeed:FORMat:SAVE** <FileName>

Saves the current list of frame formats to an XML file with the specified name.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Setting parameters:**
<FileName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save as" on page 882

**SBUS<sb>:USBHspeed:FORMat:CLR**

Erases from the volatile memory all field descriptions and frame formats that you have created.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Edit list" on page 882

**SBUS<sb>:USBHspeed:FORMat:FCOunt?**

Returns the number of available frame formats.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Edit list" on page 882

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ADDField**

Appends a new field description to the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add field" on page 883

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLDCount?**

Returns the number of field descriptions available for the seleced frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:NAME** <Name>

Specifies the name for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Frame name" on page 882

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:COLor** <Color>

Specifies the color for the selected frame format.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**

<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4278255615

**Usage:** Asynchronous command

**Manual operation:** See "Frame color" on page 882

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:TYPE** <Type>

Sets an original frame type.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**

<Type>

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 882

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID1Field** <ID1Field>
**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID2Field** <ID2Field>
**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID3Field** <ID3Field>

Sets the identifying field type for the configured frame type.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**

<ID3Field>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> field" on page 882

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID1Value** <ID1Value>
**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID2Value** <ID2Value>
**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID3Value** <ID3Value>

Sets the value of the corresponding field type.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Parameters:**
<ID3Value>

**Usage:** Asynchronous command

**Manual operation:** See "ID<n> value" on page 882

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:NAME** <Name>

Specifies the name for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Name>

**Usage:** Asynchronous command

**Manual operation:** See "Field name" on page 883

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:COLor** <Color>

Specifies the color for the selected field of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Color> Use 32-bit RGB encoding in decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 4294967040

**Usage:** Asynchronous command

**Manual operation:** See "Field color" on page 883

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FSTBit** <FirstBit>

In order of transmission, sets the first bit of the specified field in the payload.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<FirstBit> Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "First bit" on page 884

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:BITCount** <BitCount>

Specifies the bit count, hence length, of the selected field in the selected frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<BitCount> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Bit count" on page 884

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:REPetition** <Repetition>

Sets the repetition count of the field. Setting the value to 0 indicates an unlimited repetition that continues until the end of the payload is reached.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**

<Repetition> Range: 0 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Repetition" on page 884

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:BITorder** <BitOrder>

Specifies, in which order the algorithm evaluates the bits of the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first

**MSBF**
Most significant bit first

*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 884

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FORMat** <NumericFormat>

Specifies the numerical format for the condition value of the selected field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<NumericFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG | DEC | LIST | FLIN

**HEX**
Hexadecimal format

**OCT**
Octal format

**BIN**
Binary format

**ASCII = ASCii**
ASCII format

**SIGN**
Signed, decimal format

**DEC=USIG**
Unsigned, decimal format

**LIST**
Value list

**FLIN**
Linear function

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Numeric format" on page 884

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ADDValue**

Creates a new hexagonal value for customization.

Available if `SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

Related commands:

`SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue`

`SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel`

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Add value" on page 885

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue** <Value>

Sets the hexagonal value to convert it into a labeled state.

Available if `SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FORMat` = `LIST`.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

<p> Index of the value

**Parameters:**

<Value> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Value" on page 885

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel** <Label>

Sets the label to convert the corresponding hexagonal value to a labeled state.

Available if SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FORMat = LIST.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field
<p> Index of the value

**Parameters:**
<Label>

**Usage:** Asynchronous command

**Manual operation:** See "Label" on page 885

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:UNIT** <Unit>

Sets the unit for the linear function.

Available if SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FORMat = FLIN.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**
<Unit>

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 885

---

**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:PARA** <ParameterA>
**SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:PARB** <ParameterB>

Sets the parameters of the linear function.

- Parameter A: Slope (rate of change)
- Parameter B: Offset

Available if SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FORMat = FLIN.

**Suffix:**
<sb> 1 to 4, index of the serial bus
<fr> Index of the frame
<fl> Index of the field

**Parameters:**

<ParameterB> Range: -1E+20 to 1E+20
Increment: 1E-12
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Parameter A,B" on page 886

### 18.18.21.4 Software trigger

There are two commands for each parameter, that you can use for defining the USB 2.0 settings.

For example, to set the *Frame type =WRITe > Field =Address >Data* value you can use one of the following commands:

- `TRIGger:SBSW:USBHspeed:FRAMe1:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `STRIGger:SBSW:USBHspeed:DMIN "WRITe", "Address", 01100`
  Defines the parameter by using the frame and field name.

TRIGger:SBSW:USBHspeed:CHKall....2000
TRIGger:SBSW:USBHspeed:CLR....2000
TRIGger:SBSW:USBHspeed:INVert....2000
TRIGger:SBSW:USBHspeed:RST....2000
TRIGger:SBSW:USBHspeed:FRENable....2000
TRIGger:SBSW:USBHspeed:FRAMe<fr>:ENABle....2000
TRIGger:SBSW:USBHspeed:BIT....2001
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:BIT....2001
TRIGger:SBSW:USBHspeed:DMAX....2001
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DMAX....2001
TRIGger:SBSW:USBHspeed:DMIN....2001
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DMIN....2001
TRIGger:SBSW:USBHspeed:DOPerator....2002
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DOPerator....2002
TRIGger:SBSW:USBHspeed:FIENable....2002
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:ENABle....2002
TRIGger:SBSW:USBHspeed:IMAX....2002
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:IMAX....2002
TRIGger:SBSW:USBHspeed:IMIN....2003
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:IMIN....2003
TRIGger:SBSW:USBHspeed:IOPerator....2003
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:IOPerator....2003
TRIGger:SBSW:USBHspeed:ERENable....2004
TRIGger:SBSW:USBHspeed:ERRor<m>:ENABle....2004
SBUS<sb>:USBHspeed:SWTindex?....2004
SBUS<sb>:USBHspeed:SWTTime?....2004

**TRIGger:SBSW:USBHspeed:CHKall**

Enables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 887

**TRIGger:SBSW:USBHspeed:CLR**

Disables the software trigger for all available frames and error types.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 887

**TRIGger:SBSW:USBHspeed:INVert**

Inverts the current state of the frame and error types for the software trigger: all frames and error types that were enabled are disabled and vice versa.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 887

**TRIGger:SBSW:USBHspeed:RST**

Presets the state of the selected frames and error types for the software trigger.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Check all, Clear, Invert, Preset" on page 887

**TRIGger:SBSW:USBHspeed:FRENable** <Frame>,<Enabler>
**TRIGger:SBSW:USBHspeed:FRENable?** <Enabler>
**TRIGger:SBSW:USBHspeed:FRAMe<fr>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Frame type" on page 888

**TRIGger:SBSW:USBHspeed:BIT** <Frame>,<Field>,<Bit>
**TRIGger:SBSW:USBHspeed:BIT?** <Bit>
**TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<BitState> ONE | ZERO
*RST: ZERO

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 888

---

**TRIGger:SBSW:USBHspeed:DMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:USBHspeed:DMAX?** <Data>
**TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DMAX** <Data_Max>

Sets the end value of a data pattern range for the software trigger, if the operator is set to `INRange` or `OORANGe`.

You can set the operator with `TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Max> List of comma-separated values

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 888

---

**TRIGger:SBSW:USBHspeed:DMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:USBHspeed:DMIN?** <Data>
**TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DMIN** <Data_Min>

Specifies the data pattern, or sets the start value of a data pattern range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Min> List of comma-separated values

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 888

---

**TRIGger:SBSW:USBHspeed:DOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:USBHspeed:DOPerator?** <Operator>
**TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DOPerator** <Data_Operator>

Sets the operator for the data pattern of the software trigger in the selected field of the selected frame.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Data_Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 888

---

**TRIGger:SBSW:USBHspeed:FIENable** <Frame>,<Field>,<Enabler>
**TRIGger:SBSW:USBHspeed:FIENable?** <Enabler>
**TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame of the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<CondEnabler> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 888

---

**TRIGger:SBSW:USBHspeed:IMAX** <Frame>,<Field>,<Data>
**TRIGger:SBSW:USBHspeed:IMAX?** <Data>
**TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:IMAX** <Index_Max>

Sets the end value of an index range for the software trigger if the operator is set to `INRange`.

You can set the operator with `TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IOPerator`.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Max> Range: 1 to 65535
Increment: 1
*RST: 65535

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 888

---

**TRIGger:SBSW:USBHspeed:IMIN** <Frame>,<Field>,<Data>
**TRIGger:SBSW:USBHspeed:IMIN?** <Data>
**TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:IMIN** <Index_Min>

Specifies the index, or sets the start value of an index range for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Min> Range: 1 to 65535
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 888

---

**TRIGger:SBSW:USBHspeed:IOPerator** <Frame>,<Field>,<Operator>
**TRIGger:SBSW:USBHspeed:IOPerator?** <Operator>
**TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:IOPerator** <Index_Operator>

Sets the operator for the index in the selected field of the selected frame for the software trigger.

**Suffix:**
<fr> Index of the frame

<fl> Index of the field

**Parameters:**
<Index_Operator> EQUal | INRange | RANGe
*RST: INRange

**Usage:** Asynchronous command

**Manual operation:** See "Edit" on page 888

**TRIGger:SBSW:USBHspeed:ERENable** <ErrorName>,<Enabler>
**TRIGger:SBSW:USBHspeed:ERENable?** <Enabler>
**TRIGger:SBSW:USBHspeed:ERRor<m>:ENABle** <Enable>

Defines the error type for the software trigger.

**Suffix:**
<m> Index of the error

**Parameters:**
<Enable> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Error type" on page 889

**SBUS<sb>:USBHspeed:SWTindex?**

Queries the software trigger index.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Index> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBHspeed:SWTTime?**

Queries the software trigger time in seconds.

If no software trigger event has occurred, the command returns `Invalid`.

**Suffix:**
<sb> 1 to 4, index of the serial bus

**Return values:**
<Time> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

### 18.18.21.5 Decode results

SBUS<sb>:USBHspeed:FCOunt? 2005
SBUS<sb>:USBHspeed:FRAMe<fr>:ADDRess? 2005
SBUS<sb>:USBHspeed:FRAMe<fr>:FLDCount? 2006
SBUS<sb>:USBHspeed:FRAMe<fr>:BITRate? 2006
SBUS<sb>:USBHspeed:FRAMe<fr>:DATA? 2006
SBUS<sb>:USBHspeed:FRAMe<fr>:STARt? 2007
SBUS<sb>:USBHspeed:FRAMe<fr>:STATus? 2007
SBUS<sb>:USBHspeed:FRAMe<fr>:STOP? 2008
SBUS<sb>:USBHspeed:FRAMe<fr>:FLD<fl>:FVALue? 2008
SBUS<sb>:USBHspeed:FRAMe<fr>:FLD<fl>:LABel? 2008
SBUS<sb>:USBHspeed:FRAMe<fr>:FLD<fl>:STATe? 2009
SBUS<sb>:USBHspeed:FRAMe<fr>:FLD<fl>:VALue? 2009
SBUS<sb>:USBHspeed:FRAMe<fr>:CRC? 2010
SBUS<sb>:USBHspeed:FRAMe<fr>:FADDress? 2010
SBUS<sb>:USBHspeed:FRAMe<fr>:FCRC? 2010
SBUS<sb>:USBHspeed:FRAMe<fr>:ENDPoint? 2011
SBUS<sb>:USBHspeed:FRAMe<fr>:FRNumber? 2011
SBUS<sb>:USBHspeed:FRAMe<fr>:PID? 2011
SBUS<sb>:USBHspeed:FRAMe<fr>:PIDName? 2011
SBUS<sb>:USBHspeed:FRAMe<fr>:TYPE? 2012
SBUS<sb>:USBHspeed:FRAMe<fr>:SPET? 2012
SBUS<sb>:USBHspeed:FRAMe<fr>:SPEU? 2013
SBUS<sb>:USBHspeed:FRAMe<fr>:SPPort? 2013
SBUS<sb>:USBHspeed:FRAMe<fr>:SPS? 2013
SBUS<sb>:USBHspeed:FRAMe<fr>:SPSC? 2014

**SBUS<sb>:USBHspeed:FCOunt?**

Returns the number of decoded frames.

**Suffix:**

<sb> 1 to 4, index of the serial bus
Selects the serial bus.

**Return values:**

<Count> Total number of decoded frames.

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBHspeed:FRAMe<fr>:ADDRess?**

Returns the device address value of the indicated frame. That is, the address value that is shown in the decoded cells and in the decode results table.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<AddressValue> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:FLDCount?**

Returns the number of fields in the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<Count>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:BITRate?**

Returns the number of fields in the specified frame.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<Frame calculated bit rate> Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:DATA?**

Returns a comma-separated list of values, where the first value is the number of data bytes in the frame, and the remaining values are this number of data bytes.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<Count>

<Values>

**Example:** `SBUS:USBH:FRAMe3:DATA?`
`5,24,156,75,62,230`
The first value, here 5, is the number of data bytes to follow.

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:STARt?**

Returns the start time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStart> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:STATus?**

Returns the overall state of the frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameState> OK | PID | CRC | LENGth | BTST | INComplete

**OK**
The frame is valid.

**PID**
A packet identifier error signifies that the integrity check has failed, see Packet structure > "PID" in Section 14.19.1, "USB protocol basics", on page 851. The 8-bit PID is invalid, if its 4 least significant bits (LSB) are not the complement of its 4 most significant bits (MSB).

**CRC**
Cyclic redundancy check failed.

**LENGth**
Wrong length of frame.

**BTST**
Bit stuffing error occurred.

**INComplete**
The frame is not completely contained in the acquisition. The acquired part of the frame is usually valid, but cannot always be trusted.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:STOP?**

Returns the end time of the specified frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameStop> Range: -1E+26 to 1E+26
Increment: 1E-10
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:FLD<fl>:FVALue?**

Returns the formatted value of the specified field for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<FormattedValue>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:FLD<fl>:LABel?**

Returns the name (label) of the specified field in the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Label>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:FLD<fl>:STATe?**

Returns the overall state of the specified field for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<State> OK | PID | CRC | LENGth | BTST | INComplete

**OK**
The field has no error, it is valid.

**PID**
A packet identifier error signifies that the integrity check has failed, see Packet structure > "PID" in Section 14.19.1, "USB protocol basics", on page 851. The 8-bit PID is invalid, if its 4 least significant bits (LSB) are not the complement of its 4 most significant bits (MSB).

**CRC**
Cyclic redundancy check failed.

**LENGth**
Wrong length of the field.

**BTST**
Bit stuffing error occurred.

**INComplete**
The field is not completely contained in the acquisition. The acquired part of the field is usually valid, but cannot always be trusted.

*RST: OK

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:FLD<fl>:VALue?**

Returns the data value of the specified field.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

<fl> Index of the field

**Return values:**
<Value> Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:CRC?**

Returns the cyclic redundancy check (CRC) checksum of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<CRCValue> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:FADDress?**

Returns the formatted address for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedAddrVal>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:FCRC?**

Returns the formatted CRC value for the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedCRCVal>

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBHspeed:FRAMe<fr>:ENDPoint?**

Returns the endpoint for the selected serial bus and frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Endpoint> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBHspeed:FRAMe<fr>:FRNumber?**

Returns the frame number for the selected serial bus and frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameNumber> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBHspeed:FRAMe<fr>:PID?**

Returns the packet identifier for the selected serial bus and frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Command> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBHspeed:FRAMe<fr>:PIDName?**

Returns the name of the packet identifier for the selected serial bus and frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FormattedCmd>

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:TYPE?**

Returns the frame type of the selected frame.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<FrameType> UNKNown | DATA | TOKen | SOF | HSHK | PERR | SPLit | PING | RSVD

Unknown, data, token, start of frame, handshake, PRE/ERR frame (with PID=0x3C meaning either preamble or error), split frame, ping response, reserved

*RST: UNKNown

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:SPET?**

Returns the content of the "SPLIT packet transaction Endpoint Type" field of the selected frame. This 2-bit field tells the USB hub, which split-transaction-state machine to use:

- 00 Control
- 01 Isochronous
- 10 Bulk
- 11 Interrupt

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Split ET> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBHspeed:FRAMe<fr>:SPEU?**

Returns the content of the "SPLIT packet transaction E/U" field (end or unused) of the selected frame. This single-bit field, combined with the S field (see SBUS<sb>:USBHspeed:FRAMe<fr>:SPS?), describes the type of "payload continuation encoding" used for start-split packets. For complete-split packets, it is unused.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Split E/U> Range: 0 to 65535
Increment: 1
*RST: 0

**Example:**

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBHspeed:FRAMe<fr>:SPPort?**

Returns the content of the "SPLIT packet transaction Port" field of the selected frame. This field is a 7-bit word that contains the port number of the target hub, for which this split transaction is destined.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**
<Split Port> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**SBUS<sb>:USBHspeed:FRAMe<fr>:SPS?**

Returns the content of the "SPLIT packet transaction S" field of the selected frame. This single-bit field indicates the speed for the interrupt or control transaction announced by this SPLIT packet: 0 = Full speed, 1 = Low speed.

**Suffix:**
<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<Split S> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**SBUS<sb>:USBHspeed:FRAMe<fr>:SPSC?**

Returns the content of the "SPLIT packet transaction Start / Complete" field of the selected frame. This single-bit field indicates the SPLIT token: 0 = start-split transaction, 1 = complete-split transaction.

**Suffix:**

<sb> 1 to 4, index of the serial bus

<fr> Index of the frame

**Return values:**

<Split SC> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

## 18.19 Logic analysis on digital channels

This section describes the remote commands of MSO option R&S MXO3-B1.

Some of the commands in the following section are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, section "Command sequence and synchronization"

### 18.19.1 Digital channels

Except for `DIGital<m>:PROBe<ch>[:ENABle]`, all other `DIGital:` commands affect only the settings of the first MSO bus (Logic1). The settings of all other logic groups (Logic 2, 3,4) remain unchanged.

DIGital<m>:PROBe<ch>[:ENABle]?....2015
DIGital<m>:HYSTeresis....2015
DIGital<m>:LABel....2016
DIGital<m>:SIZE....2016
DIGital<m>:SKEW....2016
DIGital<m>:STATe....2016
DIGital<m>:TECHnology....2017
DIGital<m>:THCoupling....2017
DIGital<m>:THReshold....2017

---

**DIGital<m>:PROBe<ch>[:ENABle]?**

Enables one digital probe.

**Suffix:**

<m> 0 to 15, index of the logic channel
Digital channels 0 to 7 belong to probe 1, and channels 8 to 15 belong to probe 2.

<ch> 1..2
Index of the logic probe (pod)

**Return values:**

<ProbeConnected> OFF | ON

**Usage:** Query only
Asynchronous command

---

**DIGital<m>:HYSTeresis** <Hysteresis>

Sets the hysteresis for the indicated digital channel.

The setting affects only the settings of the first MSO bus (Logic1).

You can set the hysteresis for all buses with `PBUS<pb>:HYSTeresis<n>`.

**Suffix:**

<m> 0 to 15, index of the logic channel

**Parameters:**

<Hysteresis> MAXimum | ROBust | NORMal

**MAXimum**
Maximum value that is possible and useful for the signal and its settings, to be used for noisy signals.

**ROBust**
Different hysteresis values for falling and rising edges to avoid an undefined state of the trigger system, to be used for very noisy signals.

**NORMal**
Small value suitable for the signal and its settings, to be used for clean signals.

**Usage:** Asynchronous command

**DIGital<m>:LABel** <Label>

Sets a name for the indicated digital channel. The name is displayed in the diagram.

The setting affects only the settings of the first MSO bus (Logic1).

You can set the label for all buses with `PBUS<pb>:BIT<n>:LABel`

**Suffix:**
<m> 0 to 15, index of the logic channel

**Parameters:**
<Label> String containing the channel name

**Usage:** Asynchronous command

**DIGital<m>:SIZE** <Size>

Sets the vertical size for the channel group to which the indicated digital channel belongs.

**Suffix:**
<m> 0..15
Number of the digital channel.
Channel groups: 0..3; 4..7; 8..11; 12..15

**Parameters:**
<Size> Number of vertical divisions per logic channel
Range: 0.2 to 10

**Usage:** Asynchronous command

**DIGital<m>:SKEW** <Skew>

Sets an individual delay for each digital channel to time-align it with other digital channels. The skew value compensates delays that are known from the circuit specifics or caused by the different length of cables.

The setting affects only the settings of the first MSO bus (Logic1).

You can set the skew for all buses with `PBUS<pb>:BIT<n>:SKEW`.

**Suffix:**
<m> 0 to 15, index of the logic channel

**Parameters:**
<Skew>

**Usage:** Asynchronous command

**DIGital<m>:STATe** <State>

Enables or disables the indicated digital channel, displays it, and enables the Logic 1 if the bus was disabled.

If another active bus already uses the selected digital channel, the instrument disables the other bus to avoid conflicts.

For Logic 1, the `DIGital<m>:STATe` command has the same effect as `PBUS<pb>:BIT<n>[:STATe]`. To enable digital channels for buses 2, 3 and 4, use the `PBUS<pb>:BIT<n>[:STATe]` command.

**Suffix:**
<m> 0 to 15, index of the logic channel

**Parameters:**
<State> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Active logic channel" on page 894

---

**DIGital<m>:TECHnology** <Technology>

Selects the threshold voltage for various types of integrated circuits and applies it to all digital channels.

The setting affects only the settings of the first MSO bus (Logic1).

You can set the technology value for all buses with `PBUS<pb>:TECHnology`.

**Suffix:**
<m> 0..15
The suffix is irrelevant.

**Parameters:**
<Technology> V15 | V25 | V165 | V125 | V09 | VM13 | V38 | V20 | V0 | MANual
See `PBUS<pb>:TECHnology`.

**Usage:** Asynchronous command

---

**DIGital<m>:THCoupling** <State>

Sets the threshold and the hysteresis for all digital channels of Logic1 to the same value.

**Suffix:**
<m> Irrelevant, omit the suffix.

**Parameters:**
<State> OFF | ON

**Usage:** Asynchronous command

---

**DIGital<m>:THReshold** <Value>

Sets the logical threshold for the channel group to which the indicated digital channel belongs.

The setting affects only the settings of the first MSO bus (Logic1).

You can set the threshold for all buses with `PBUS<pb>:TECHnology` or `PBUS<pb>:THReshold<n>`

See also: `DIGital<m>:THCoupling` on page 2017

**Suffix:**
<m> 0..15
Number of the digital channel.
Channel groups: 0..3; 4..7; 8..11; 12..15

**Parameters:**
<Value>

**Usage:** Asynchronous command

### 18.19.2 Logic configuration

The following commands configure the four logic groups (parallel buses).

PBUS<pb>:BIT<n>:LABel....2018
PBUS<pb>:BIT<n>:SKEW....2019
PBUS<pb>:BIT<n>[:STATe]....2019
PBUS<pb>:CLEar....2020
PBUS<pb>:CLOCk....2020
PBUS<pb>:CLON....2020
PBUS<pb>:CLSLope....2020
PBUS<pb>:DECTable:SHOW....2021
PBUS<pb>:DECTable:COUNt?....2021
PBUS<pb>:DECTable:DATA?....2021
PBUS<pb>:DIGSignals:POSition....2022
PBUS<pb>:DIGSignals:SCALe....2022
PBUS<pb>:DISPlay:SHBU....2022
PBUS<pb>:DISPlay:SHDI....2022
PBUS<pb>:HYSTeresis<n>....2023
PBUS<pb>:POSition....2023
PBUS<pb>:SCALe....2024
PBUS<pb>:SKEW....2024
PBUS<pb>:STATe....2024
PBUS<pb>:TECHnology....2025
PBUS<pb>:THCoupling....2025
PBUS<pb>:THReshold<n>....2025

---

**PBUS<pb>:BIT<n>:LABel** <Label>

Sets a name for the indicated digital channel. The name is displayed in the diagram.

**Suffix:**
<pb> 1...4, index of the logic group

<n> 0..15
Selects the bit of the bus word. Each bit corresponds to a digital channel.

**Parameters:**

<Label> String containing the channel name

**Usage:** Asynchronous command

---

**PBUS<pb>:BIT<n>:SKEW** <Skew>

Sets an individual delay for each digital channel to time-align it with other digital channels.

The skew value compensates delays that are known from the circuit specifics or caused by the different length of cables. The skew between the probe boxes of the digital channels and the probe connectors of the analog channels is automatically aligned by the instrument.

**Suffix:**

<pb> 1...4, index of the logic group

<n> 0..15
Selects the bit of the bus word. Each bit corresponds to a digital channel.

**Parameters:**

<Skew>
Range: -2E-07 to 2E-07
Increment: 2E-10
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "D0-D15" on page 898

---

**PBUS<pb>:BIT<n>[:STATe]** <Assigned>

Enables the selected logic group. The corresponding signal icon appears on the signal bar.

If another active bus already uses the selected digital channel, the instrument disables the other bus to avoid conflicts.

**Suffix:**

<pb> 1...4, index of the logic group

<n> 0..15
Selects the bit of the bus word. Each bit corresponds to a digital channel.

**Parameters:**

<Assigned> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Active logic channel" on page 894

**PBUS<pb>:CLEar**

Removes all assigned digital channels from the bus.

**Suffix:**
<pb> 1...4, index of the logic group

**Usage:** Setting only
Asynchronous command

**PBUS<pb>:CLOCk** <ClockSource>

Selects the digital channel used as clock.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<ClockSource> D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15
Clock channel
*RST: D0

**Usage:** Asynchronous command

**Manual operation:** See "Clock source" on page 896

**PBUS<pb>:CLON** <Clocked>

Defines if the bus is a clocked bus - one of the digital channels serves as the clock of the bus.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<Clocked> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Bus clocked" on page 896

**PBUS<pb>:CLSLope** <ClockSlope>

Selects the slope of the clock signal at which all digital channels of the bus are analyzed.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<ClockSlope> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Clock slope" on page 896

---

**PBUS<pb>:DECTable:SHOW** <DecodeTable>

If enabled, a result table is shown with decoded values and corresponding points in time of the bus signal. Each clock edge corresponds to one row in the table.

The decode table is only available for clocked buses to check the data words.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<DecodeTable> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show decode table" on page 896

---

**PBUS<pb>:DECTable:COUNt?**

Returns the number of rows in the decode table. Each clock edge corresponds to one row in the table. The decode table is only available for clocked buses.

**Suffix:**
<pb> 1...4, index of the logic group

**Return values:**
<Count> Number of rows

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Show decode table" on page 896

---

**PBUS<pb>:DECTable:DATA?**

Returns a list of decoded values and corresponding points in time from the decode table. Each data pair corresponds to one clock edge, which is one row in the table. The decode table is only available for clocked buses.

**Suffix:**
<pb> 1...4, index of the logic group

**Return values:**
<Data> Comma-separated list of values

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Show decode table" on page 896

**PBUS<pb>:DIGSignals:POSition** <DigChanPosition>

Sets the vertical position of all active digital channels.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<DigChanPosition> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**PBUS<pb>:DIGSignals:SCALe** <RelativeHeight>

Sets the size of the display that is used by each active digital signal.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<RelativeHeight> Range: 2 to 100
Increment: 0.5
*RST: 6.25
Default unit: %

**Usage:** Asynchronous command

**PBUS<pb>:DISPlay:SHBU** <ShowBus>

If enabled, the resulting bus signal and bus values are displayed in the diagram.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<ShowBus> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show bus" on page 895

**PBUS<pb>:DISPlay:SHDI** <ShwDigSigns>

If enabled, the selected digital channels are shown in the diagram. Each channel is displayed as a logic signal.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**

<ShwDigSigns> OFF | ON

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show dig. signals" on page 893

---

**PBUS<pb>:HYSTeresis<n>** <Hysteresis>

Defines the size of the hysteresis for the respective channels.

**Suffix:**

<pb> 1...4, index of the logic group

<n> 1..4
Selects the channel group:
1 = digital channels 0..3
2 = digital channels 4..7
3 = digital channels 8..11
4 = digital channels 12..15

**Parameters:**

<Hysteresis> MAXimum | ROBust | NORMal

**MAXIMUM = MAXimum**
Maximum value that is possible and useful for the signal and its settings (corresponds to "Medium" in the GUI).

**ROBUST = ROBust**
Different hysteresis values for falling and rising edges to avoid an undefined state of the trigger system (corresponds to "Large" in the GUI).

**NORMAL = NORMal**
The instrument sets a value suitable for the signal and its settings (corresponds to "Small" in the GUI).

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis" on page 895

---

**PBUS<pb>:POSition** <YPosition>

Sets the position of the indicated logic group waveform.

**Suffix:**

<pb> 1...4, index of the logic group

**Parameters:**

<YPosition> Range: 0 to 100
Increment: 0.1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**PBUS<pb>:SCALe** <RelativeHeight>

Sets the size of the display that is used by the indicated logic group waveform.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<RelativeHeight> Range: 3 to 100
Increment: 0.5
*RST: 10
Default unit: %

**Usage:** Asynchronous command

**PBUS<pb>:SKEW** <SkewOffset>

Sets a general delay for all digital channels.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<SkewOffset> Range: -2E-07 to 2E-07
Increment: 2E-10
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Skew" on page 898

**PBUS<pb>:STATe** <State>

Enables the selected logic group. The corresponding signal icon appears on the signal bar.

Dependencies: At least one digital channel must be enabled for the selected bus, otherwise the command does not work.

The bus is enabled automatically if the first digital channel is enabled with `PBUS<pb>:BIT<n>[:STATe]`.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<State> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 893

**PBUS<pb>:TECHnology** <Technology>

Selects the threshold voltage for various types of integrated circuits and applies it to all digital channels.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<Technology> V15 | V25 | V165 | V125 | V09 | VM13 | V38 | V20 | V0 | MANual
V15: TTL
V25: CMOS 5.0 V
V165: CMOS 3.3 V
V125: CMOS 2.5 V
V09: CMOS 1.85 V
VM13: ECL, -1.3 V
V38: PECL
V20: LVPECL
V0: Ground
MANual: Set a user-defined threshold value with `DIGital<m>:THReshold` on page 2017.
*RST: V165

**Usage:** Asynchronous command

**Manual operation:** See "Technology, Threshold" on page 894

**PBUS<pb>:THCoupling** <LevelCoupling>

Sets the threshold and the hysteresis for all digital channels and all buses to the same value.

For Logic 1, the command `DIGital<m>:THCoupling` has the same effect.

**Suffix:**
<pb> 1...4, index of the logic group

**Parameters:**
<LevelCoupling> OFF | ON
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Level coupling" on page 894

**PBUS<pb>:THReshold<n>** <Threshold>

Sets the logical threshold for the indicated channel group.

Alternatively you can use the following commands:

- To select from a list of predefined technologies: `PBUS<pb>:TECHnology`
- For logic 1: `DIGital<m>:THReshold`

See also DIGital<m>:THCoupling on page 2017.

**Suffix:**

<pb> 1...4, index of the logic group

<n> 1..4
Selects the channel group:
1 = digital channels 0..3;
2 = digital channels 4..7
3 = digital channels 8..11
4 = digital channels 12..15

**Parameters:**

<Threshold>
Range: -8 to 8
Increment: 0.025
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Technology, Threshold" on page 894

### 18.19.3 Digital resolution

**ACQuire:DRESolution?**

Returns the current digital resolution of the digital channels.

**Return values:**

<DigRes>
Range: 1E-15 to 100
Increment: 1E-11
*RST: 0.0005
Default unit: s

**Usage:** Query only
Asynchronous command

**ACQuire:POINts:DVALue?**

Returns the current digital record length used by each digital channel.

**Return values:**

<DigReclgth>
Range: 1000 to 200000000
Increment: 2
*RST: 1000
Default unit: pts

**Usage:** Query only
Asynchronous command

### 18.19.4 Export of digital data

To transfer the waveform data to a remote computer, use:

- `DIGital<m>:DATA:HEADer?` on page 1147
- `DIGital<m>:DATA[:VALues]?` on page 1147
- `PBUS<pb>:DATA:FORMat` on page 1148
- `PBUS<pb>:DATA:HEADer?` on page 1148
- `PBUS<pb>:DATA[:VALues]?` on page 1148

For fast export of several waveforms at once, use `EXPort:WAVeform:DATA[:VALues]?` on page 1150.

For export of waveform data to file, see Section 18.13.4, "Waveform data: export to file", on page 1127.

## 18.20 Waveform generator (option R&S MXO3-B6)

This section describes the remote commands of the waveform generator.

The instrument preset does not affect the generator settings. Each generator has its own preset: `WGENerator<wg>:PRESet`.

Some of the commands in the following section are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, section "Command sequence and synchronization"

### 18.20.1 Waveform generator setup

#### 18.20.1.1 General settings

WGENerator<wg>:FREQuency ..... 2028
WGENerator<wg>:FUNCtion:PULSe[:WIDTh] ..... 2028
WGENerator<wg>:FUNCtion:RAMP[:SYMMetry] ..... 2028
WGENerator<wg>:FUNCtion[:SELect] ..... 2029
WGENerator<wg>:FUNCtion[:SQUare]:DCYCle ..... 2029
WGENerator<wg>:OUTPut[:LOAD] ..... 2029
WGENerator<wg>:PERiod ..... 2030
WGENerator<wg>:PRESet ..... 2030

WGENerator<wg>:MODulation:AM:CAMPlitude....2030
WGENerator<wg>:SOURce....2031
WGENerator<wg>:VOLTage:DCLevel....2031
WGENerator<wg>:VOLTage:INVersion....2032
WGENerator<wg>:VOLTage:OFFSet....2032
WGENerator<wg>:VOLTage[:VPP]....2032
WGENerator<wg>[:ENABle]....2032

**WGENerator<wg>:FREQuency** <Frequency>

Sets the frequency of the waveform.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<Frequency>
Range: 0.001 to 50E+6 or lower, depending on the function type.
Increment: 0.001
*RST: 10E+6
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Frequency" on page 902

**WGENerator<wg>:FUNCtion:PULSe[:WIDTh]** <PulseWidth>

Sets the pulse width, the pulse duration of the generated pulse waveform.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<PulseWidth>
Range: 1.65E-08 to 90000
Increment: 1
*RST: 5E-07
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Pulse width" on page 904

**WGENerator<wg>:FUNCtion:RAMP[:SYMMetry]** <RampSymmetry>

Sets the symmetry of a ramp waveform, the percentage of time the waveform is rising. By changing the symmetry of the ramp, you can create, for example, triangular waveforms.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**

<RampSymmetry>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Symmetry" on page 904

---

**WGENerator<wg>:FUNCtion[:SELect]** <FunctionType>

Selects the type of waveform to be generated.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<FunctionType> SINusoid | SQUare | RAMP | DC | PULSe | SINC | CARDiac | GAUSs | LORNtz | HAVer | EXPRise | EXPFall | ARBitrary

`SINC`: cardinal sine
HAVer: haversine (great-circle distance between two points on a sphere)

*RST: SINusoid

**Usage:** Asynchronous command

**Manual operation:** See "Function type" on page 901

---

**WGENerator<wg>:FUNCtion[:SQUare]:DCYCle** <SquareDutyCycle>

Sets the duty cycle for the pulse function.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<SquareDutyCycle>
Range: 0.01 to 99.99
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Duty cycle" on page 904

---

**WGENerator<wg>:OUTPut[:LOAD]** <Load>

Select the user load, the load of the DUT at its connection.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<Load> FIFTy | HIZ

`FIFTy`: 50Ω
`HIZ`: High-Z (high input impedance)

*RST: HIZ

**Usage:** Asynchronous command

**Manual operation:** See "User load" on page 904

---

**WGENerator<wg>:PERiod** <Period>

Sets the period of the pulse waveform, if `WGENerator<wg>:FUNCtion[:SELect]` is set to `PULSe`.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<Period> Range: 1E-08 to 1000
Increment: 1
*RST: 1E-06
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Period" on page 905

---

**WGENerator<wg>:PRESet**

Presets the generator to a default setup. The default includes the following settings:

- "Function type" = "Sine"
- "Frequency" = "1 MHz"
- "Amplitude" = "1 Vpp"

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Default setup" on page 904

---

**WGENerator<wg>:MODulation:AM:CAMPlitude** <CarrierAmplitude>

Sets the mean peak-to-peak voltage of the modulated signal if AM modulation is active.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<CarrierAmplitude>
Range: 0.01 to 12
Increment: 0.01
*RST: 0.4
Default unit: Vpp

**Usage:** Asynchronous command

**Manual operation:** See "Carrier amplitude" on page 903

---

**WGENerator<wg>:SOURce** <OperationMode>

Selects the operation mode of the waveform generator-

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<OperationMode> FUNCgen | MODulation | SWEep | ARBGenerator

**FUNCgen**
Enables the function generator and disables modulation, sweep, and arbitrary waveforms

**MODulation**
Enables the modulation, disables sweep and selects the sine function.

**SWEep**
Enables the sweep, disables modulation, and selects the sine function.

**ARBGenerator**
Selects the arbitrary function and disables modulation and sweep.

*RST: FUNCgen

**Usage:** Asynchronous command

---

**WGENerator<wg>:VOLTage:DCLevel** <DCLevel>

Sets the level for the DC signal, if `WGENerator<wg>:FUNCtion[:SELect]` is set to `DC`.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<DCLevel>
Range: -5 to 5
Increment: 0.01
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "DC level" on page 904

**WGENerator<wg>:VOLTage:INVersion** <Inversion>

Inverts the waveform at the offset level.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<Inversion> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Inversion" on page 901

**WGENerator<wg>:VOLTage:OFFSet** <Offset>

Sets the vertical offset of the generated waveform.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<Offset> Range: -5.995 to 5.995
Increment: 0.01
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Offset" on page 903

**WGENerator<wg>:VOLTage[:VPP]** <Amplitude>

Sets the amplitude of the waveform. If AM modulation is active, this value defines the peak amplitude of the modulated signal.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<Amplitude> Range: 0.01 to 12
Increment: 0.01
*RST: 0.4
Default unit: Vpp

**Usage:** Asynchronous command

**Manual operation:** See "Amplitude, Peak amplitude" on page 903

**WGENerator<wg>[:ENABle]** <State>

Enables the function generator.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<State> OFF | ON
*RST: No effect

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 901

### 18.20.1.2 Modulation settings

WGENerator<wg>:MODulation[:STATe] ..... 2033
WGENerator<wg>:MODulation:AM:DCYCle ..... 2034
WGENerator<wg>:MODulation:AM:DEPTh ..... 2034
WGENerator<wg>:MODulation:AM:FREQuency ..... 2034
WGENerator<wg>:MODulation:AM:SYMMetry ..... 2035
WGENerator<wg>:MODulation:AM[:FUNCtion] ..... 2035
WGENerator<wg>:MODulation:CARRier:FREQuency ..... 2035
WGENerator<wg>:MODulation:CARRier:PERiod ..... 2035
WGENerator<wg>:MODulation:FM:DCYCle ..... 2036
WGENerator<wg>:MODulation:FM:DEViation ..... 2036
WGENerator<wg>:MODulation:FM:FREQuency ..... 2036
WGENerator<wg>:MODulation:FM:SYMMetry ..... 2037
WGENerator<wg>:MODulation:FM[:FUNCtion] ..... 2037
WGENerator<wg>:MODulation:FSK:FONE ..... 2037
WGENerator<wg>:MODulation:FSK:FTWO ..... 2037
WGENerator<wg>:MODulation:FSK[:RATE] ..... 2038
WGENerator<wg>:MODulation:NDCLevel ..... 2038
WGENerator<wg>:MODulation:NLPCent ..... 2038
WGENerator<wg>:MODulation:PWM:DCYCle ..... 2039
WGENerator<wg>:MODulation:PWM:DEPTh ..... 2039
WGENerator<wg>:MODulation:PWM:FREQuency ..... 2039
WGENerator<wg>:MODulation:PWM:SYMMetry ..... 2040
WGENerator<wg>:MODulation:PWM[:FUNCtion] ..... 2040
WGENerator<wg>:MODulation:TYPE ..... 2040

---

**WGENerator<wg>:MODulation[:STATe]** <Modulation>

Enables or disables modulation.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<Modulation> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Modulation state" on page 905

**WGENerator<wg>:MODulation:AM:DCYCle** <SquareDutyCycle>

Sets the duty cycle for a square waveform. The duty cycle expresses for what percentage fraction of the period, the waveform is active, i.e. the signal state is high.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<SquareDutyCycle>
Range: 10 to 90
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Duty cycle" on page 906

**WGENerator<wg>:MODulation:AM:DEPTh** <Depth>

Sets the modulation depth, the percentage of the amplitude range that is used for AM modulation.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<Depth>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Depth" on page 906

**WGENerator<wg>:MODulation:AM:FREQuency** <Frequency>

Sets the frequency of the modulation waveform for AM frequency modulation.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<Frequency>
Range: 0.001 to 1000000
Increment: 0.001
*RST: 1000
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Frequency" on page 906

**WGENerator<wg>:MODulation:AM:SYMMetry** <RampSymmetry>

Sets the symmetry for the AM ramp modulation waveform, the percentage of time that the waveform is rising.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<RampSymmetry> Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Symmetry" on page 906

**WGENerator<wg>:MODulation:AM[:FUNCtion]** <SignalType>

Selects the type of the modulating signal for AM modulation.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<SignalType> SINusoid | SQUare | RAMP
*RST: SINusoid

**Usage:** Asynchronous command

**Manual operation:** See "Signal type" on page 906

**WGENerator<wg>:MODulation:CARRier:FREQuency** <FreqCarrierAlias>

Sets the frequency of the carrier signal for a modulation waveform.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<FreqCarrierAlias> Range: 0.001 to 100000000
Increment: 0.001
*RST: 1000000
Default unit: Hz

**Usage:** Asynchronous command

**WGENerator<wg>:MODulation:CARRier:PERiod** <PeriodCarrierAlias>

Sets the period of the carrier signal for a modulation waveform.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**

<PeriodCarrierAlias> Range: 1E-08 to 1000
Increment: 1
*RST: 1E-06
Default unit: s

**Usage:** Asynchronous command

---

**WGENerator<wg>:MODulation:FM:DCYCle** <SquareDutyCycle>

Sets the duty cycle for a square waveform. The duty cycle expresses for what percentage fraction of the period, the waveform is active, i.e. the signal state is high.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<SquareDutyCycle> Range: 10 to 90
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Duty cycle" on page 906

---

**WGENerator<wg>:MODulation:FM:DEViation** <Deviation>

Sets the frequency deviation, the maximum difference between the FM modulated signal and the carrier signal.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<Deviation> Range: 0.001 to 10000000
Increment: 1
*RST: 1000
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Deviation" on page 907

---

**WGENerator<wg>:MODulation:FM:FREQuency** <Frequency>

Sets the frequency of the modulating waveform for FM modulation.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<Frequency>
Range: 0.001 to 1000000
Increment: 0.001
*RST: 1000
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Frequency" on page 906

---

**WGENerator<wg>:MODulation:FM:SYMMetry** <RampSymmetry>

Sets the symmetry for the FM ramp modulation waveform, the percentage of time that the waveform is rising.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<RampSymmetry>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Symmetry" on page 906

---

**WGENerator<wg>:MODulation:FM[:FUNCtion]** <SignalType>

Selects the type of the modulating signal for FM modulation.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<SignalType>
SINusoid | SQUare | RAMP
*RST: SINusoid

**Usage:** Asynchronous command

**Manual operation:** See "Signal type" on page 906

---

**WGENerator<wg>:MODulation:FSK:FONE** <Frequency1>
**WGENerator<wg>:MODulation:FSK:FTWO** <Frequency2>

Sets the frequency of the first /second signal in FSK modulated signal.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<Frequency2>
Range: 0.001 to 50E+6
Increment: 1
*RST: 1000
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Frequency 1/Frequency 2" on page 907

---

**WGENerator<wg>:MODulation:FSK[:RATE]** <Rate>

Sets the frequency at which the signal switches between `WGENerator<wg>:MODulation:FSK:FONE` and `WGENerator<wg>:MODulation:FSK:FTWO`.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<Rate>
Range: 0.001 to 1000000
Increment: 1
*RST: 1000
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "FSK rate" on page 907

---

**WGENerator<wg>:MODulation:NDCLevel** <LevelDC>

Sets the DC noise level, if `WGENerator<wg>:FUNCtion[:SELect]` is set to `DC`.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<LevelDC>
Range: 0 to 10
Increment: 0.01
*RST: 0
Default unit: Vpp

**Usage:** Asynchronous command

**Manual operation:** See "Noise level for DC" on page 904

---

**WGENerator<wg>:MODulation:NLPCent** <LevelPct>

Sets the level of the noise in percentage of the set "Amplitude" output of the signal.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<LevelPct>
Range: 0 to 100
Increment: 0.1
*RST: 0
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Noise level in %" on page 903

---

**WGENerator<wg>:MODulation:PWM:DCYCle** <SquareDutyCycle>

Sets the duty cycle for a square waveform. The duty cycle expresses for what percentage fraction of the period, the waveform is active, i.e. the signal state is high.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<SquareDutyCycle>
Range: 10 to 90
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Duty cycle" on page 906

---

**WGENerator<wg>:MODulation:PWM:DEPTh** <Depth>

Sets the modulation depth, the percentage of the amplitude range that is used for PWM modulation.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<Depth>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Modulation depth" on page 907

---

**WGENerator<wg>:MODulation:PWM:FREQuency** <Frequency>

Sets the frequency of the modulating waveform for PWM modulation.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<Frequency>
Range: 0.001 to 1000000
Increment: 0.001
*RST: 1000
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Frequency" on page 906

---

**WGENerator<wg>:MODulation:PWM:SYMMetry** <RampSymmetry>

Sets the symmetry for the PWM ramp modulation waveform, the percentage of time that the waveform is rising.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<RampSymmetry>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Symmetry" on page 906

---

**WGENerator<wg>:MODulation:PWM[:FUNCtion]** <SignalType>

Selects the type of the modulating signal for PWM modulation.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<SignalType>
SINusoid | SQUare | RAMP
*RST: SINusoid

**Usage:** Asynchronous command

**Manual operation:** See "Signal type" on page 906

---

**WGENerator<wg>:MODulation:TYPE** <ModulationType>

Selects the modulation type, which defines how the carrier signal is modified.

**Suffix:**

<wg> Irrelevant, omit the suffix.

**Parameters:**

<ModulationType>
NONE | AM | FM | PWM | ASK | FSK
*RST: AM

**Usage:** Asynchronous command

**Manual operation:** See "Modulation type" on page 905

### 18.20.1.3 ARB settings

WGENerator<wg>:ARBGen:NAME ..... 2041
WGENerator<wg>:ARBGen:OPEN ..... 2041
WGENerator<wg>:ARBGen:RUNMode ..... 2041
WGENerator<wg>:ARBGen:RUNSingle ..... 2042
WGENerator<wg>:ARBGen:SAMPles? ..... 2042
WGENerator<wg>:ARBGen:SRATe ..... 2042
WGENerator<wg>:ARBGen[:SOURce] ..... 2043

---

**WGENerator<wg>:ARBGen:NAME** <Folder>

Sets the file path and the file for an arbitrary waveform, if `WGENerator<wg>:ARBGen[:SOURce]` is set to `ARBitrary`.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<Folder>

**Usage:** Asynchronous command

**Manual operation:** See "Open" on page 911

---

**WGENerator<wg>:ARBGen:OPEN**

Loads the arbitrary waveform, which is selected with the `WGENerator<wg>:ARBGen:NAME` command.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Open" on page 911

---

**WGENerator<wg>:ARBGen:RUNMode** <RunMode>

Sets the duration for which the signal of the arbitrary generator will be output after the trigger event.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<RunMode> SINGle | REPetitive
*RST: REPetitive

**Usage:** Asynchronous command

**Manual operation:** See "Run mode" on page 910

---

**WGENerator<wg>:ARBGen:RUNSingle**

Executes a single period of the arbitrary signal generator, if `WGENerator<wg>:ARBGen:RUNMode` is set to `SINGle`.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Run single" on page 910

---

**WGENerator<wg>:ARBGen:SAMPles?**

Returns the number of samples in the loaded waveform.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Return values:**
<NumSamples> Range: 0 to 32E+3
Increment: 10
*RST: 0
Default unit: pts

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Number of samples" on page 911

---

**WGENerator<wg>:ARBGen:SRATe** <SampleRate>

Sets the sample rate for the arbitrary waveform.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<SampleRate> Range: 1 to 312500000
Increment: 10
*RST: 1000000
Default unit: Sa/s

**Usage:** Asynchronous command

**Manual operation:** See "Sample rate" on page 911

**WGENerator<wg>:ARBGen[:SOURce]** <WaveformSource>

Selects the source of the arbitrary waveform.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<WaveformSource> ARBitrary
*RST: ARBitrary

**Usage:** Asynchronous command

**Manual operation:** See "Arb wfm source" on page 911

### 18.20.1.4 Sweep settings

WGENerator<wg>:SWEep[:STATe] ..... 2043
WGENerator<wg>:SWEep:TYPE ..... 2043
WGENerator<wg>:SWEep:FSTart ..... 2044
WGENerator<wg>:SWEep:TIME ..... 2044
WGENerator<wg>:SWEep:FEND ..... 2044

**WGENerator<wg>:SWEep[:STATe]** <Sweep>

Enables or disables the sweeping.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<Sweep> OFF | ON
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Sweep state" on page 908

**WGENerator<wg>:SWEep:TYPE** <Type>

Sets the type of the sweep, a linear or logarithmic change of the frequency.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<Type> LIN | LOG
*RST: LIN

**Usage:** Asynchronous command

**Manual operation:** See "Sweep type" on page 908

**WGENerator<wg>:SWEep:FSTart** <StartFrequency>

Sets the start frequency of the sweep signal.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<StartFrequency>
Range: 0.001 to 50E+6
Increment: 0.001
*RST: 1000
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Start frequency" on page 908

**WGENerator<wg>:SWEep:TIME** <Time>

Sets the duration of the sweep.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<Time>
Range: 0.001 to 500
Increment: 1
*RST: 0.001
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Sweep time" on page 908

**WGENerator<wg>:SWEep:FEND** <StopFrequency>

Sets the stop frequency of the sweep signal.

**Suffix:**
<wg> Irrelevant, omit the suffix.

**Parameters:**
<StopFrequency>
Range: 0.001 to 50E+6
Increment: 0.001
*RST: 1000000
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Stop frequency" on page 908

## 18.21 Status reporting

This section describes the remote commands that are used to read the status registers.

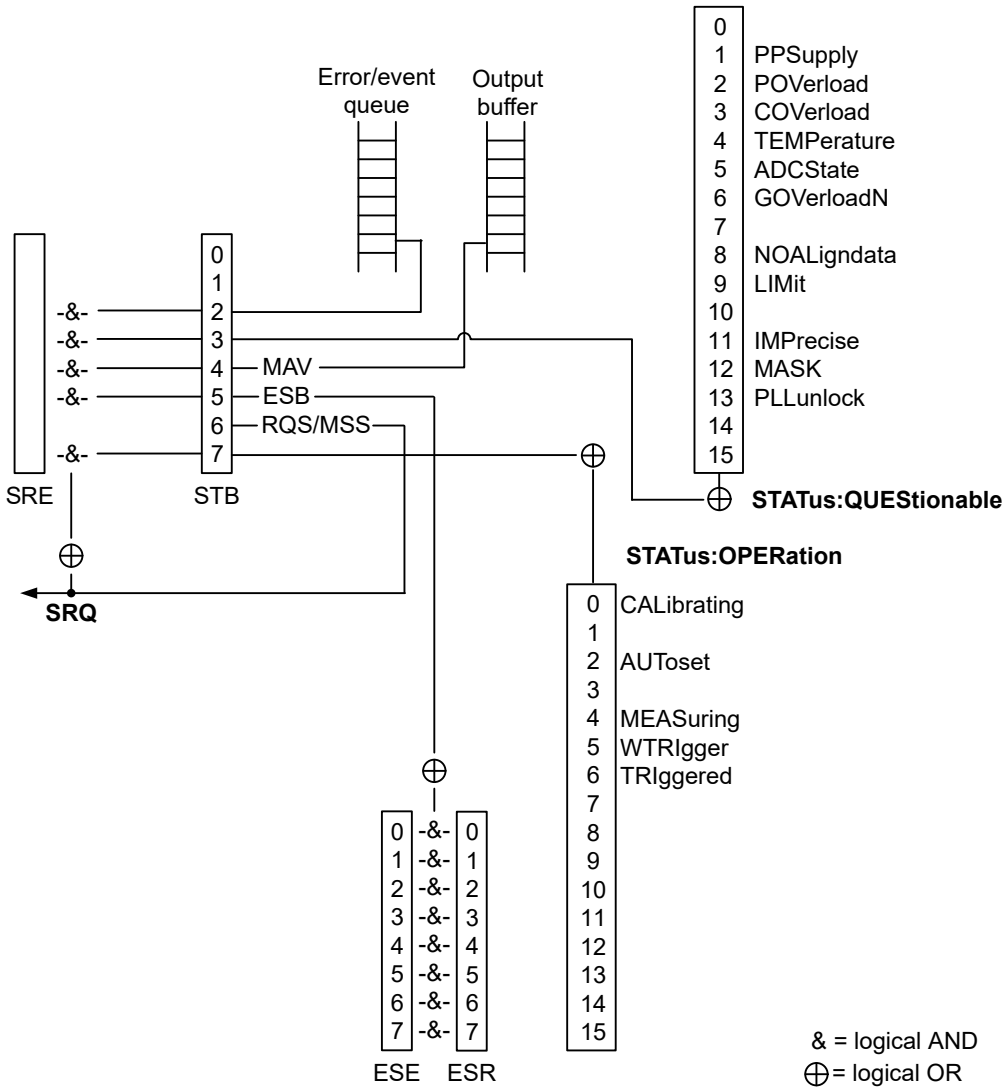



*Figure 18-1: Overview of the status registers*

For detailed information on the structure, hierarchy, and contents of the status registers, see Section 17.5, "Remote control - status reporting system", on page 927.

- General commands....2046
- STATus:OPERation register....2046
- STATus:QUEStionable registers....2048
- Reading out the CONDition part....2050
- Reading out the EVENt part....2050
- Controlling the ENABle part....2051
- Controlling the negative transition part....2052

- Controlling the positive transition part....2052
- Other status commands....2053
- Programming tips and examples....2054

### 18.21.1 General commands

**STATus:PRESet**

Resets the status registers.

All `PTRansition` bits are set to 1, i.e. all transitions from 0 to 1 are detected.

All `NTRansition` bits are set to 0, i.e. a transition from 1 to 0 in a `CONDition` bit is not detected.

All `EVENt` bits are set to 0.

The `ENABle` bits of `STATus:OPERation` and `STATus:QUEStionable` are set to 0, i.e. all events in these registers are not passed on.

**Usage:** Setting only
SCPI confirmed

### 18.21.2 STATus:OPERation register

`STATus:OPERation` commands provide information on the activity of the instrument.

See also: Section 17.5.3.3, "STATus:OPERation register", on page 932.

STATus:OPERation:CONDition?....2046
STATus:OPERation:ENABle....2047
STATus:OPERation[:EVENt]?....2047
STATus:OPERation:NTRansition....2047
STATus:OPERation:PTRansition....2048

**STATus:OPERation:CONDition?**

Returns the bit of the action the instrument is currently executing. The contents of the STATus:OPERation register is retained.

The bit assignment is described in Section 17.5.3.3, "STATus:OPERation register", on page 932.

**Usage:** Query only
SCPI confirmed
Asynchronous command

**STATus:OPERation:ENABle** <Enable>

Controls the ENABle part of the STATus:OPERation register. The ENABle defines which events in the EVENt part of the status register are forwarded to the OPERation summary bit (bit 7) of the status byte. The status byte can be used to create a service request.

**Parameters:**

<Enable> Range: 1 to 65535
Increment: 1

**Example:** `STATus:OPERation:ENABle 5`
The ALIGnment event (bit 0) and AUToset event (bit 2) are forwarded to the OPERation summary bit of the status byte.

**STATus:OPERation[:EVENt]?**

The `CONDition` command returns information on actions the instrument is currently executing. The contents of the register is retained.

The `EVENt` command returns information on actions the instrument has executed since the last reading. Reading the `EVENt` register deletes its contents.

Bits:

- 0 = ALIGnment
- 2 = AUToset
- 4= MEASuring
- 5= WTRIgger (wait for trigger)
- 6= TRIggered

**Example:** For an example, see Example "Querying with STATus:OPERation:CONDition" on page 2055.

**Usage:** Query only
SCPI confirmed
Asynchronous command

**STATus:OPERation:NTRansition** <SumBit>

The command sets the bits of the NTRansition part of the STATus:OPERation register. A bit set in the NTRansition register causes a bit transition from 1 to 0 in the CONDition register to produce an entry in the EVENt register.

Reading the information does not clear the register.

**Parameters:**

<SumBit> Range: 0 to 65535
Increment: 1

**Usage:** SCPI confirmed
Asynchronous command

**STATus:OPERation:PTRansition** <SumBit>

The command sets the bits of the PTRansition part of the STATus:OPERation register. A bit set in the PTRansition register causes a bit transition from 0 to 1 in the CONDition register to produce an entry in the EVENt register.

Reading the information does not clear the register.

**Parameters:**

<SumBit> Range: 0 to 65535
Increment: 1

**Usage:** SCPI confirmed
Asynchronous command

## 18.21.3 STATus:QUEStionable registers

The commands of the `STATus:QUEStionable` subsystem control the status reporting structures of the `STATus:QUEStionable` registers.

See also: Section 17.5.3.4, "STATus:QUEStionable register", on page 933.

The query of the `[:EVENt]` and `[:ENABLe]` commands, return a list of the affected sources:

***Table 18-3: Source values for STATus:QUEStionable:...:[:EVENt] and STATus:QUEStionable:...:[:ENABLe]***

| Bits | Source values | Description |
|---|---|---|
| For all bits | NONE | |
| | ALL | All available sources are activated |
| PPSupply | PRobe<m> | Active probe |
| POVerload | PRobe<m> | Active probe |
| COVerload | CHAN<m> | Analog channels |
| | WARNCHAN<m> | Warning overload channels |
| | EXTTRIGGERIN | External analog signal connected to the external trigger input |
| | TRIGGEROUT | Trigger out signal |
| TEMPerature | WINSTRUMENT | Temperature warning: indicates that the device is getting hot and should be cooled. |
| | EINSTRUMENT | Temperature error: the device is too hot and will automatically shut down in a few seconds. |
| | WGENERATOR1 | Temperature warning on waveform generator 1 |
| | WGENERATOR2 | Temperature warning on waveform generator 2 |
| ADCState | CPCHAN<m> | Positive clipping for analog channels |
| | CNCHAN<m> | Negative clipping for analog channels |

| Bits | Source values | Description |
|---|---|---|
| | CPPRobe<m> | Positive clipping for probes |
| | CNPRobe<m> | Negative clipping for probes |
| GOVerload | WGENerator<m> | Available generators |
| | MEAS<mg> | Enabled measurements |
| IMPRecise | MEAS<mg> | Enabled measurements |
| MASK | MASK<m> | Enabled masks |
| PLLunlock | PLL100 | |

Some of the status commands can be used interchangeably:

***Table 18-4: Overview: STATus:QUEStionable and respective CHANnel, MEASurement and WGENerator commands***

| STATus:QUEStionable register | MEASurement |
|---|---|
| STATus:QUEStionable:COVerload<br>Bit: CHANnel<ch> | CHANnel<ch>:OVERload:STATus |
| STATus:QUEStionable:COVerload<br>Bit: WCHannel<ch> | CHANnel<ch>:WARNoverload:STATus |
| STATus:QUEStionable:COVerload<br>Bit: EXTTRIGGERIN | TRIGger:ANEDge:OVERload:STATus |
| STATus:QUEStionable:COVerload<br>Bit: TRIGGEROUT | TRIGger:ACTions:OUT:OVERload:STATus |
| STATus:QUEStionable:ADCState<br>Bit: CPCHannel<ch> | CHANnel<ch>:ADCState:STATus:PCLipping |
| STATus:QUEStionable:ADCState<br>Bit: CNCHannel<ch> | CHANnel<ch>:ADCState:STATus:NCLipping |
| STATus:QUEStionable:PPSupply<br>Bit: CNCHannel<ch> | CHANnel<ch>PPSupply:STATus |
| STATus:QUEStionable:IMPRecise<br>Bit: MEAS<mg> | MEASurement<mg>:IMPRecise:STATus |
| STATus:QUEStionable:GOVerload<br>Bit: WGENerator<wg> | WGENerator<wg>:GOVerload |
| STATus:QUEStionable:<br>TEMPerature<br>Bit: WGENerator<wg> | WGENerator<wg>:TEMPerature |

### 18.21.4 Reading out the CONDition part

See also: Section 18.21.3, "STATus:QUEStionable registers", on page 2048.

**CHANnel<ch>:OVERload:STATus:CONDition?**
**CHANnel<ch>:ADCState:STATus:NCLipping:CONDition?**
**CHANnel<ch>:ADCState:STATus:PCLipping:CONDition?**
**CHANnel<ch>:WARNoverload:STATus:CONDition?**
**CHANnel<ch>:PPSupply:STATus:CONDition?**
**MEASurement<mg>:IMPRecise:STATus:CONDition?**
**STATus:QUEStionable:MASK:CONDition?**
**TRIGger:ACTions:OUT:OVERload:STATus:CONDition?**
**TRIGger:ANEDge:OVERload:STATus:CONDition?**
**WGENerator<wg>:GOVerload:STATus:CONDition?**
**WGENerator<wg>:TEMPerature:STATus:CONDition?**
**STATus:QUEStionable:ADCState:CONDition?**
**STATus:QUEStionable:COVerload:CONDition?**
**STATus:QUEStionable:GOVerload:CONDition?**
**STATus:QUEStionable:IMPRecise:CONDition?**
**STATus:QUEStionable:PLL:CONDition?**
**STATus:QUEStionable:PPSupply:CONDition?**
**STATus:QUEStionable:TEMPerature:CONDition?**

Returns the contents of the `CONDition` part of the status register to check for questionable instrument or measurement states. This part contains information on the action currently being performed in the instrument.

Reading the `CONDition` registers does not delete the contents since it indicates the current hardware status.

**Return values:**
<Value> bit dependent, comma-separated values

**Usage:** Query only

### 18.21.5 Reading out the EVENt part

**CHANnel<ch>:ADCState:STATus:NCLipping[:EVENt]?**
**CHANnel<ch>:ADCState:STATus:PCLipping[:EVENt]?**
**CHANnel<ch>:OVERload:STATus[:EVENt]** <Value>
**CHANnel<ch>:WARNoverload:STATus[:EVENt]?**
**CHANnel<ch>:PPSupply:STATus[:EVENt]** <Value>
**MEASurement<mg>:IMPRecise:STATus[:EVENt]?**
**STATus:QUEStionable:MASK[:EVENt]?**
**TRIGger:ACTions:OUT:OVERload:STATus[:EVENt]** <Value>
**TRIGger:ANEDge:OVERload:STATus[:EVENt]** <Value>
**WGENerator<wg>:GOVerload:STATus[:EVENt]** <Value>
**WGENerator<wg>:TEMPerature:STATus[:EVENt]** <Value>
**STATus:QUEStionable:ADCState[:EVENt]?**
**STATus:QUEStionable:COVerload[:EVENt]?**

**STATus:QUEStionable:GOVerload[:EVENt]?**
**STATus:QUEStionable:IMPRecise[:EVENt]?**
**STATus:QUEStionable:PLL[:EVENt]?**
**STATus:QUEStionable:PPSupply[:EVENt]?**
**STATus:QUEStionable:TEMPerature[:EVENt]?**

Returns the contents of the `EVENt` part of the status register to check if an event has occurred since the last reading.

Reading an `EVENt` register deletes its contents.

**Return values:**

<Value> bit dependent, comma-separated values

See Source values for STATus:QUEStionable:...:[:EVENt] and STATus:QUEStionable:...:[:ENABLe] for a list of the return values.

**Usage:** Query only

### 18.21.6 Controlling the ENABle part

See also: Section 18.21.3, "STATus:QUEStionable registers", on page 2048.

**CHANnel<ch>:OVERload:STATus:ENABle** <Value>
**CHANnel<ch>:ADCState:STATus:NCLipping:ENABle** <Value>
**CHANnel<ch>:ADCState:STATus:PCLipping:ENABle** <Value>
**CHANnel<ch>:WARNoverload:STATus:ENABle** <Value>
**CHANnel<ch>:PPSupply:STATus:ENABle** <Value>
**MEASurement<mg>:IMPRecise:STATus:ENABle** <Value>
**STATus:QUEStionable:MASK:ENABle** <Value>
**TRIGger:ACTions:OUT:OVERload:STATus:ENABle** <Value>
**TRIGger:ANEDge:OVERload:STATus:ENABle** <Value>
**WGENerator<wg>:GOVerload:STATus:ENABle** <Value>
**WGENerator<wg>:TEMPerature:STATus:ENABle** <Value>
**STATus:QUEStionable:ADCState:ENABle** <Value>
**STATus:QUEStionable:COVerload:ENABle** <Value>
**STATus:QUEStionable:GOVerload:ENABle** <Value>
**STATus:QUEStionable:IMPRecise:ENABle** <Value>
**STATus:QUEStionable:PLL:ENABle** <Value>
**STATus:QUEStionable:PPSupply:ENABle** <Value>
**STATus:QUEStionable:TEMPerature:ENABle** <Value>

Sets the `ENABle` part that allows true conditions in the EVENt part to be reported for the summary bit in the status byte.

These events can be used for a service request. If a bit in the `ENABle` part is 1, and the corresponding `EVENt` bit is true, a positive transition occurs in the summary bit. This transition is reported to the next higher level.

See Source values for STATus:QUEStionable:...:[:EVENt] and STATus:QUEStionable:...:[:ENABLe] for a list of the return values.

**Parameters:**
<Value> bit dependent, comma-separated values

### 18.21.7 Controlling the negative transition part

See also: Section 18.21.3, "STATus:QUEStionable registers", on page 2048.

**CHANnel<ch>:ADCState:STATus:NCLipping:NTRansition** <Value>
**CHANnel<ch>:ADCState:STATus:PCLipping:NTRansition** <Value>
**CHANnel<ch>:OVERload:STATus:NTRansition** <Value>
**CHANnel<ch>:PPSupply:STATus:NTRansition** <Value>
**CHANnel<ch>:WARNoverload:STATus:NTRansition** <Value>
**MEASurement<mg>:IMPRecise:STATus:NTRansition** <Value>
**STATus:QUEStionable:MASK:NTRansition** <Value>
**TRIGger:ACTions:OUT:OVERload:STATus:NTRansition** <Value>
**TRIGger:ACTions:OUT:OVERload:STATus:PTRansition** <Value>
**TRIGger:ANEDge:OVERload:STATus:NTRansition** <Value>
**WGENerator<wg>:GOVerload:STATus:NTRansition** <Value>
**WGENerator<wg>:TEMPerature:STATus:NTRansition** <Value>
**STATus:QUEStionable:ADCState:NTRansition** <Value>
**STATus:QUEStionable:COVerload:NTRansition** <Value>
**STATus:QUEStionable:GOVerload:NTRansition** <Value>
**STATus:QUEStionable:IMPRecise:NTRansition** <Value>
**STATus:QUEStionable:PLL:NTRansition** <Value>
**STATus:QUEStionable:PPSupply:NTRansition** <Value>
**STATus:QUEStionable:TEMPerature:NTRansition** <Value>

Sets the negative transition filter.

If a bit is set, a transition from 1 to 0 in the condition part causes an entry to be made in the corresponding bit of the `EVENt` part of the register.

**Parameters:**
<Value> Source dependent, comma-separated values

### 18.21.8 Controlling the positive transition part

See also: Section 18.21.3, "STATus:QUEStionable registers", on page 2048.

**CHANnel<ch>:ADCState:STATus:NCLipping:PTRansition** <Value>
**CHANnel<ch>:ADCState:STATus:PCLipping:PTRansition** <Value>
**CHANnel<ch>:OVERload:STATus:PTRansition** <Value>
**CHANnel<ch>:PPSupply:STATus:PTRansition** <Value>
**CHANnel<ch>:WARNoverload:STATus:PTRansition** <Value>
**MEASurement<mg>:IMPRecise:STATus:PTRansition** <Value>
**STATus:QUEStionable:MASK:PTRansition** <Value>
**TRIGger:ACTions:OUT:OVERload:STATus:PTRansition** <Value>
**TRIGger:ANEDge:OVERload:STATus:PTRansition** <Value>
**WGENerator<wg>:GOVerload:STATus:PTRansition** <Value>
**WGENerator<wg>:TEMPerature:STATus:PTRansition** <Value>

**STATus:QUEStionable:ADCState:PTRansition** <Value>
**STATus:QUEStionable:COVerload:PTRansition** <Value>
**STATus:QUEStionable:GOVerload:PTRansition** <Value>
**STATus:QUEStionable:IMPRecise:PTRansition** <Value>
**STATus:QUEStionable:PLL:PTRansition** <Value>
**STATus:QUEStionable:PPSupply:PTRansition** <Value>
**STATus:QUEStionable:TEMPerature:PTRansition** <Value>

Sets the positive transition filter.

If a bit is set, a transition from 0 to 1 in the condition part causes an entry to be made in the corresponding bit of the `EVENt` part of the register.

**Parameters:**
<Value> Source dependent, comma-separated values

### 18.21.9 Other status commands

**STATus:QUEStionable:PLL:RCONdition** <Value>

Reset condition, allows to continue after an PLL unlock of the external reference.

**Setting parameters:**
<Value> PLL100

**Usage:** Setting only

### 18.21.10 Programming tips and examples

**Example: Observing the limits violation of a measurement**

The following example describes how to use the status register commands to observe if a violation of a measurement limit has occurred.

```
*SRE 8
// Enables the STATus:QUEStionable bit in the service-request (SRE)

STATus:QUEStionable:ENABle 512
// Enables the LIMIt bit of the STATus:QUEStionable register

MEASurement1:ENABle ON
MEASurement2:ENABle ON
MEASurement3:ENABle ON
// Enables measurement 1,2,3


STATus:QUEStionable:LIMit:ENABle ALL
// Enables the limit check for all active measurements

STATus:QUEStionable:LIMit:ENABle?
MEAS1,MEAS2,MEAS3
//Queries which measurement were active at the time the
//STATus:QUEStionable:LIMit:ENABle was send

*SRQ?
// A service request was sent

*STB?
72
// STATus:QUEStionable is set:
// #72= 64 + 8 = 2^6 + 2^3
// 64 : bit 6 is awlays on when a SRQ is sent
// 8: bit 3 for STATus:QUEStionable

STATus:QUEStionable:EVENt?
512
// 512= 2^9,

STATus:QUEStionable:LIMit:EVENt?
MEAS1,MEAS2
// Measurements 1 and 2 have exceeded the limit
// The event register is cleared after the query


STATus:QUEStionable:LIMit:EVENt?
NONE
// The event bit has been set to 0 after the first query
```

```
STATus:QUEStionable:LIMit:CONDition?
MEAS1,MEAS2
// Measurements 1 and 2 are still exceeding the limit


STATus:QUEStionable:EVENt?
0

// Waiting for an event
*SRQ?

// Measurement 3 exceeds the limit

STATus:QUEStionable:LIMit:CONDition?
MEAS1,MEAS2,MEAS3
// Measurements 1,2 and 3 are currently exceeding the limit


STATus:QUEStionable:LIMit:EVENt?
MEAS3
// Measurement 3 has newly  exceeded the limit
// The event register is cleared after the query
```

### Example: Querying with STATus:OPERation:CONDition

The following example shows how the result of `STATus:OPERation:CONDition?` changes during a single acquisition.

```
SINGLe
// Measurement starts

STATus:OPERation:CONDition?
16
//Measuring

STATus:OPERation:CONDition?
48
//Pre trigger waiting time is over (16 Measuring + 32 WaitForTrigger)

STATus:OPERation:CONDition?
112
//Trigger is deactivated (16 Measuring + 32 WaitForTrigger + 64 Triggered)

STATus:OPERation:CONDition?
96
// Measurement has finished (32 WaitForTrigger + 64 Triggered)
```

### Query OPC status in the event status register

If you activate a time-consuming operation and wait for completion with `*OPC?`, a time-out could occur before the operation is finished and you do not receive the returned

"1". In addition, the test program is blocked while waiting with `*OPC?`. It is not possible to process other (not interdependent) commands in the meantime or to communicate with other instruments.

Thus, for time-consuming operations, you can avoid blocking the communication by sending the operation complete command `*OPC`:

```
*CLS
*OPC
```

Afterwards you can poll the operation complete status in the event status register with `*ESR?`.

This query returns the content of the event status register and afterwards clears the content. See also Section 17.5.3.2, "Event status register (ESR) and event status enable register (ESE)", on page 931.

## 18.22 Deprecated commands

The following commands are provided for compatibility to previous oscilloscope versions only. For new remote control programs, use the specified alternative commands.

### 18.22.1 Base instrument

| Legacy command | Replaced by | Comment |
|---|---|---|
| PROBe<ch>:SETup:OFFSet:TOPMeter | `PROBe<ch>:SETup:ADVanced:PMToffset` on page 1015 | Replaced since FW 2.10 |
| PROBe<ch>:SETup:ZAXV | `PROBe<ch>:SETup:EPATtenuator` on page 1012 | Replaced since FW 2.10 |

***Table 18-5: Status reporting***

| Legacy command | Replaced by | Comment |
|---|---|---|
| MEASurement<mg>:MARGin:STATus:CONDition?<br>MEASurement<mg>:MARGin:STATus:ENABle<br>MEASurement<mg>:MARGin:STATus:NTRansition<br>MEASurement<mg>:MARGin:STATus:PTRansition<br>MEASurement<mg>:MARGin:STATus[:EVENt] | Use `MEASurement<mg>:LIMit:STATus:` commands | Obsolete since FW 2.9 |
| STATus:QUEStionable:MARGin:CONDition?<br>STATus:QUEStionable:MARGin:ENABle<br>STATus:QUEStionable:MARGin:NTRansition<br>STATus:QUEStionable:MARGin:PTRansition<br>STATus:QUEStionable:MARGin[:EVENt]? | Use `STATus:QUEStionable:LIMit:` commands | Obsolete since FW 2.9 |

### 18.22.2 Applications

*Table 18-6: Signal configuration*

| Legacy command | Replaced by | Comment |
|---|---|---|
| TREFerence<m>:REFLevel | TREFerence<m>:CLK:REFLevel on page 1085 | Replaced since FW 2.8 |
| MDEFaults:SETTings<m>:ENABle | --- | Obsolete since FW 2.10 |

*Table 18-7: Waveform generator*

| Legacy command | Replaced by | Comment |
|---|---|---|
| WGENerator<wg>:MODulation:NLABsolute | - | Obsolete since FW 2.8 |
| WGENerator<wg>:VOLTage:HIGH | - | Obsolete since FW 2.9 |
| WGENerator<wg>:VOLTage:LOW | - | Obsolete since FW 2.9 |

*Table 18-8: Power analysis*

| Legacy command | Replaced by | Comment |
|---|---|---|
| POWer<m>:HARMonics:MEASurement:REAL-power[:ACTual] | POWer<m>:HARMonics:MEASurement:POWer:ACTive[:ACTual]? on page 1269 | Replaced since FW 2.8 |
| POWer<m>:QUALity:DISPlay:POWer:REALpower | POWer<m>:QUALity:RESult:POWer:ACTive[:ACTual]? on page 1259 | Replaced since FW 2.8 |
| POWer<m>:QUALity:RESult:POWer:REAL-power:AVERage<br>POWer<m>:QUALity:RESult:POWer:REAL-power:EVTCount<br>POWer<m>:QUALity:RESult:POWer:REAL-power:NPEak<br>POWer<m>:QUALity:RESult:POWer:REAL-power:PPEak<br>POWer<m>:QUALity:RESult:POWer:REAL-power:RMS<br>POWer<m>:QUALity:RESult:POWer:REAL-power:STDDev<br>POWer<m>:QUALity:RESult:POWer:REAL-power:WFMCount<br>POWer<m>:QUALity:RESult:POWer:REAL-power[:ACTual] | POWer<m>:QUALity:RESult:POWer:ACTive:AVERage? on page 1259<br>POWer<m>:QUALity:RESult:POWer:ACTive:EVTCount? on page 1259<br>POWer<m>:QUALity:RESult:POWer:ACTive:NPEak? on page 1259<br>POWer<m>:QUALity:RESult:POWer:ACTive:PPEak? on page 1259<br>POWer<m>:QUALity:RESult:POWer:ACTive:RMS? on page 1259<br>POWer<m>:QUALity:RESult:POWer:ACTive:STDDev? on page 1259<br>POWer<m>:QUALity:RESult:POWer:ACTive:WFMCount? on page 1259<br>POWer<m>:QUALity:RESult:POWer:ACTive[:ACTual]? on page 1259 | Replaced since FW 2.8 |
| POWer<m>:HARMonics:DISPlay:FREQuency:STARt[:VALue]<br>POWer<m>:HARMonics:DISPlay:FREQuency:STOP[:VALue] | POWer<m>:HARMonics:DISPlay:HARMonics:STARt[:VALue] on page 1266<br>POWer<m>:HARMonics:DISPlay:HARMonics:STOP[:VALue] on page 1266 | Replaced since FW 2.9 |

## 18.22.3 Serial protocols

### 18.22.3.1 CAN

*Table 18-9: CAN decode results*

| Legacy command | Replaced by | Comment |
|---|---|---|
| SBUS<sb>:CAN:FRAMe<fr>:FERCause<br>SBUS<sb>:CAN:FRAMe<fr>:STUFf | - | No longer needed since FW 2.9 |

### 18.22.3.2 Manchester

*Table 18-10: Manchester decode results*

| Legacy command | Replaced by | Comment |
|---|---|---|
| SBUS<sb>:MANCh:FRAMe<fr>:RES1?<br>SBUS<sb>:MANCh:FRAMe<fr>:RES2?<br>SBUS<sb>:MANCh:FRAMe<fr>:RES3? | SBUS<sb>:MANCh:FRAMe<fr>:CONE?<br>SBUS<sb>:MANCh:FRAMe<fr>:CTWO?<br>SBUS<sb>:MANCh:FRAMe<fr>:CTHRee? | Replaced since FW 2.9 |

### 18.22.3.3 NRZ clocked

*Table 18-11: NRZ clocked decode results*

| Legacy command | Replaced by | Comment |
|---|---|---|
| SBUS<sb>:NRZC:FRAMe<fr>:RES1?<br>SBUS<sb>:NRZC:FRAMe<fr>:RES2?<br>SBUS<sb>:NRZC:FRAMe<fr>:RES3? | SBUS<sb>:NRZC:FRAMe<fr>:CONE?<br>SBUS<sb>:NRZC:FRAMe<fr>:CTWO?<br>SBUS<sb>:NRZC:FRAMe<fr>:CTHRee? | Replaced since FW 2.9 |

### 18.22.3.4 NRZ unclocked

*Table 18-12: NRZ unclocked decode results*

| Legacy command | Replaced by | Comment |
|---|---|---|
| SBUS<sb>:NRZU:FRAMe<fr>:RES1?<br>SBUS<sb>:NRZU:FRAMe<fr>:RES2?<br>SBUS<sb>:NRZU:FRAMe<fr>:RES3? | SBUS<sb>:NRZU:FRAMe<fr>:CONE?<br>SBUS<sb>:NRZU:FRAMe<fr>:CTWO?<br>SBUS<sb>:NRZU:FRAMe<fr>:CTHRee? | Replaced since FW 2.9 |

# 19 Maintenance and support

The instrument does not need periodic maintenance. Only the cleaning of the instrument is essential.

We advise to check the nominal data from time to time.

To protect the front panel and to transport the instrument to another workplace safely and easily, various accessories are provided. Refer to the specifications document for available covers and cases and their order numbers.

## 19.1 Cleaning

How to clean the product is described in "Cleaning the product" on page 28.

Do not use any liquids for cleaning. Cleaning agents, solvents, acids and bases can damage the front panel labeling, plastic parts and display.

## 19.2 Changing fuses

If the product does not start, it is possible that a blown fuse is the cause.

The product is protected by 2 fuses of type Size 5x20 mm, 250V~, T5H (slow-blow), IEC 60127-2 (order no. 0099.6735.00), order number 0099.6735.00.

1. **WARNING!** The fuse is part of the AC power supply. Handling the fuse while the power is on can lead to electric shock.

   Before changing the fuse:

   a) Set the switch on the power supply to position [0].
   b) Disconnect the product from the power source.

2. The fuse slot is on the rear panel between the mains switch and AC power supply connector.
   Pull out the fuse holder.
3. Check the condition of the fuse.
4. Replace the blown fuse. Only use a fuse of the specified type.
5. Insert the fuse holder into its slot until it latches.

## 19.3 Contacting customer support

**Technical support – where and when you need it**

For quick, expert help with any Rohde & Schwarz product, contact our customer support center. A team of highly qualified engineers provides support and works with you to find a solution to your query on any aspect of the operation, programming or applications of Rohde & Schwarz products.

**Contact information**

Contact our customer support center at www.rohde-schwarz.com/support, or follow this QR code:



*Figure 19-1: QR code to the Rohde & Schwarz support page*

## 19.4 Information for technical support

If you encounter problems that you cannot solve yourself, contact your Rohde & Schwarz support center, see Contacting customer support.

The support center finds solutions more quickly and efficiently, if you provide them with information on the instrument and an error description. To create, collect and save the required information, you can create a service report.

The service report is a ZIP file with a complete bug report, all relevant setup information, reporting and log files, alignment files, and the instrument configuration.

1. Open the "Settings" > "Maintenance" dialog.
2. Select the "Service" tab.
3. Tap "Create report".
   If a USB flash drive is connected, the report is saved on the USB flash drive. Otherwise, the report is saved in the user data folder `/home/storage/userData`.
4. Create a support ticket that describes the problem, and attach the report file.

See also: Section 5.7.5, "Service", on page 116.

## 19.5 Data security

If you have to send the instrument to the service, or if the instrument is used in a secured environment, consider the document "Instrument Security Procedures" that is delivered on the MXO 3 webpage.

## 19.6 Transporting

### Lifting and carrying

See: "Lifting and carrying the instrument" on page 24

### Packing

Use the original packaging material. It consists of antistatic wrap for electrostatic protection and packing material designed for the product.

If you do not have the original packaging, use similar materials that provide the same level of protection. You can also contact your local Rohde & Schwarz service center for advice.

### Securing

When moving the MXO 3 in a vehicle or using transporting equipment, make sure that the MXO 3 is properly secured. Only use items intended for securing objects.

### Transport altitude

Unless otherwise specified in the specifications document, the maximum transport altitude without pressure compensation is 4500 m above sea level.

## 19.7 Storage

Protect the product against dust. Ensure that the environmental conditions, e.g. temperature range and climatic load, meet the values specified in the specifications document.

## 19.8 Disposal

Rohde & Schwarz is committed to making careful, ecologically sound use of natural resources and minimizing the environmental footprint of our products. Help us by disposing of waste in a way that causes minimum environmental impact.

### Disposing of electrical and electronic equipment

A product that is labeled as follows cannot be disposed of in normal household waste after it has come to the end of its life. Even disposal via the municipal collection points for waste electrical and electronic equipment is not permitted.



*Figure 19-2: Labeling in line with EU directive WEEE*

Rohde & Schwarz has developed a disposal concept for the eco-friendly disposal or recycling of waste material. As a manufacturer, Rohde & Schwarz completely fulfills its obligation to take back and dispose of electrical and electronic waste. Contact your local service representative to dispose of the product.

# List of commands

*CAL?....947
*CLS....947
*IDN?....947
*OPC....947
*OPT?....948
*RCL....948
*RST....948
*SAV....948
*SRE....949
*STB?....949
*TRG....949
*TST?....949
*WAI....950
ACQuire:AVAilable?....994
ACQuire:AVERage?....994
ACQuire:COUNt....994
ACQuire:CURRent?....995
ACQuire:DRESolution?....2026
ACQuire:HISTory:CURRent....1102
ACQuire:HISTory:ISODate?....1102
ACQuire:HISTory:PLAY....1102
ACQuire:HISTory:REPLay....1103
ACQuire:HISTory:STARt....1103
ACQuire:HISTory:STOP....1103
ACQuire:HISTory:TPACq....1103
ACQuire:HISTory:TSABsolute?....1104
ACQuire:HISTory:TSDate?....1104
ACQuire:HISTory:TSRelative?....1104
ACQuire:HISTory:TSRReference?....1105
ACQuire:HISTory[:STATe]....1105
ACQuire:INTerpolate....995
ACQuire:POINts:ARATe?....996
ACQuire:POINts:DVALue?....2026
ACQuire:POINts:MAXimum....996
ACQuire:POINts:MODE....996
ACQuire:POINts[:VALue]....995
ACQuire:POMemory?....997
ACQuire:RESolution?....997
ACQuire:RLReal?....997
ACQuire:ROLLmode:OSCapture....993
ACQuire:ROLLmode:POINts?....993
ACQuire:SEGMented:MAX....1028
ACQuire:SEGMented:STATe....1028
ACQuire:SRATe:MINimum....998
ACQuire:SRATe:MODE....998
ACQuire:SRATe[:VALue]....998
ACQuire:SRReal?....999

ACQuire:TYPE....999
AUToscale....990
CALCulate:MATH<m>:DATA:HEADer?....1144
CALCulate:MATH<m>:DATA:STYPe?....1145
CALCulate:MATH<m>:DATA[:VALues]?....1145
CALCulate:MATH<m>:ENVSelection....1101
CALCulate:MATH<m>:LABel....1100
CALCulate:MATH<m>:STATe....1098
CALCulate:MATH<m>:UNIT....1100
CALCulate:MATH<m>:VERTical:OFFSet....1101
CALCulate:MATH<m>:VERTical:SCALe:MODE....1101
CALCulate:MATH<m>:VERTical:SCALe[:VALue]....1100
CALCulate:MATH<m>[:EXPRession][:DEFine]....1098
CALCulate:SPECtrum<sp>:COUPling:FROM....1191
CALCulate:SPECtrum<sp>:COUPling:MODE....1191
CALCulate:SPECtrum<sp>:COUPling[:ENABle]....1191
CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution]:ADJusted?....1192
CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution]:AUTO....1192
CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution]:RATio....1192
CALCulate:SPECtrum<sp>:FREQuency:BANDwidth[:RESolution][:VALue]....1193
CALCulate:SPECtrum<sp>:FREQuency:CENTer....1193
CALCulate:SPECtrum<sp>:FREQuency:SCALe....1193
CALCulate:SPECtrum<sp>:FREQuency:SPAN....1194
CALCulate:SPECtrum<sp>:FREQuency:STARt....1194
CALCulate:SPECtrum<sp>:FREQuency:STOP....1194
CALCulate:SPECtrum<sp>:FREQuency:WINDow:TYPE....1195
CALCulate:SPECtrum<sp>:GATE:POSition....1200
CALCulate:SPECtrum<sp>:GATE:STARt....1200
CALCulate:SPECtrum<sp>:GATE:STOP....1201
CALCulate:SPECtrum<sp>:GATE:WIDTh....1201
CALCulate:SPECtrum<sp>:MAGNitude:LEVel....1196
CALCulate:SPECtrum<sp>:MAGNitude:RANGe....1196
CALCulate:SPECtrum<sp>:MAGNitude:SCALe....1196
CALCulate:SPECtrum<sp>:OVERlap:ENABle....1206
CALCulate:SPECtrum<sp>:OVERlap:FACTor:ADJusted?....1206
CALCulate:SPECtrum<sp>:OVERlap:FACTor:VALue....1206
CALCulate:SPECtrum<sp>:OVERlap:FCOunt:ADJusted?....1207
CALCulate:SPECtrum<sp>:OVERlap:FCOunt:LIMit:ENABle....1207
CALCulate:SPECtrum<sp>:OVERlap:FCOunt:VALue....1207
CALCulate:SPECtrum<sp>:OVERlap:FWIDth?....1208
CALCulate:SPECtrum<sp>:PEXCursion....1198
CALCulate:SPECtrum<sp>:PLISt:COUNt?....1202
CALCulate:SPECtrum<sp>:PLISt:LABel:BORDer....981
CALCulate:SPECtrum<sp>:PLISt:LABel:FREQuency[:STATe]....1204
CALCulate:SPECtrum<sp>:PLISt:LABel:INVert....1205
CALCulate:SPECtrum<sp>:PLISt:LABel:MAXCount....1204
CALCulate:SPECtrum<sp>:PLISt:LABel:MAXCount....1205
CALCulate:SPECtrum<sp>:PLISt:LABel[:STATe]....1205
CALCulate:SPECtrum<sp>:PLISt:MAXCount....1202
CALCulate:SPECtrum<sp>:PLISt:MODE....1202

CALCulate:SPECtrum<sp>:PLISt:RESult[:VALue]? 1203
CALCulate:SPECtrum<sp>:PLISt:SORT:COLumn 1204
CALCulate:SPECtrum<sp>:PLISt:SORT:ORDer 1204
CALCulate:SPECtrum<sp>:PLISt:SOURce 1203
CALCulate:SPECtrum<sp>:PLISt:SPOW 1203
CALCulate:SPECtrum<sp>:PLISt[:STATe] 1202
CALCulate:SPECtrum<sp>:PRESet 1196
CALCulate:SPECtrum<sp>:SOURce 1197
CALCulate:SPECtrum<sp>:STATe 1198
CALCulate:SPECtrum<sp>:THReshold 1198
CALCulate:SPECtrum<sp>:WAVeform:AVERage:COUNt 1199
CALCulate:SPECtrum<sp>:WAVeform:AVERage:DATA:HEADer? 1145
CALCulate:SPECtrum<sp>:WAVeform:AVERage:DATA[:VALues]? 1146
CALCulate:SPECtrum<sp>:WAVeform:AVERage:ENABle 1199
CALCulate:SPECtrum<sp>:WAVeform:MAXimum:DATA:HEADer? 1145
CALCulate:SPECtrum<sp>:WAVeform:MAXimum:DATA[:VALues]? 1146
CALCulate:SPECtrum<sp>:WAVeform:MAXimum:ENABle 1199
CALCulate:SPECtrum<sp>:WAVeform:MINimum:DATA:HEADer? 1145
CALCulate:SPECtrum<sp>:WAVeform:MINimum:DATA[:VALues]? 1146
CALCulate:SPECtrum<sp>:WAVeform:MINimum:ENABle 1199
CALCulate:SPECtrum<sp>:WAVeform:NORMal:DATA:FRAMe<fr>[:VALues]? 1146
CALCulate:SPECtrum<sp>:WAVeform:NORMal:DATA:HEADer? 1145
CALCulate:SPECtrum<sp>:WAVeform:NORMal:DATA[:VALues]? 1146
CALCulate:SPECtrum<sp>:WAVeform:NORMal[:ENABle] 1200
CALibration:DATE? 988
CALibration:RESult? 988
CALibration:TIME? 988
CALibration[:ALL] 988
CHANnel<ch>:ADCState:STATus:NCLipping:CONDition? 2050
CHANnel<ch>:ADCState:STATus:NCLipping:ENABle 2051
CHANnel<ch>:ADCState:STATus:NCLipping:NTRansition 2052
CHANnel<ch>:ADCState:STATus:NCLipping:PTRansition 2052
CHANnel<ch>:ADCState:STATus:NCLipping[:EVENt]? 2050
CHANnel<ch>:ADCState:STATus:PCLipping:CONDition? 2050
CHANnel<ch>:ADCState:STATus:PCLipping:ENABle 2051
CHANnel<ch>:ADCState:STATus:PCLipping:NTRansition 2052
CHANnel<ch>:ADCState:STATus:PCLipping:PTRansition 2052
CHANnel<ch>:ADCState:STATus:PCLipping[:EVENt]? 2050
CHANnel<ch>:BANDwidth 1003
CHANnel<ch>:COUPling 1001
CHANnel<ch>:DATA:HEADer? 1142
CHANnel<ch>:DATA[:VALues]? 1142
CHANnel<ch>:EATScale 1003
CHANnel<ch>:EATTenuation 1003
CHANnel<ch>:IMPedance 1004
CHANnel<ch>:INVert 1002
CHANnel<ch>:OFFSet 1000
CHANnel<ch>:OVERload:STATus:CONDition? 2050
CHANnel<ch>:OVERload:STATus:ENABle 2051
CHANnel<ch>:OVERload:STATus:NTRansition 2052

CHANnel<ch>:OVERload:STATus:PTRansition....2052
CHANnel<ch>:OVERload:STATus[:EVENt]....2050
CHANnel<ch>:POSition....1001
CHANnel<ch>:PPSupply:STATus:CONDition?....2050
CHANnel<ch>:PPSupply:STATus:ENABle....2051
CHANnel<ch>:PPSupply:STATus:NTRansition....2052
CHANnel<ch>:PPSupply:STATus:PTRansition....2052
CHANnel<ch>:PPSupply:STATus[:EVENt]....2050
CHANnel<ch>:RANGe....1000
CHANnel<ch>:SCALe....1000
CHANnel<ch>:SKEW:TIME....1002
CHANnel<ch>:STATe....999
CHANnel<ch>:WARNoverload:STATus:CONDition?....2050
CHANnel<ch>:WARNoverload:STATus:ENABle....2051
CHANnel<ch>:WARNoverload:STATus:NTRansition....2052
CHANnel<ch>:WARNoverload:STATus:PTRansition....2052
CHANnel<ch>:WARNoverload:STATus[:EVENt]?....2050
CURSor<cu>:AOFF....1176
CURSor<cu>:COUNt?....1176
CURSor<cu>:DISPlay:VALues....1184
CURSor<cu>:FFT:SETCenter....1188
CURSor<cu>:FFT:TOCenter....1188
CURSor<cu>:FUNCtion....1177
CURSor<cu>:HORizontal<n>:LABel....1183
CURSor<cu>:LABel....1183
CURSor<cu>:MAXimum:LEFT....1188
CURSor<cu>:MAXimum:NEXT....1189
CURSor<cu>:MAXimum:RIGHt....1188
CURSor<cu>:MAXimum[:PEAK]....1188
CURSor<cu>:PEXCursion....1189
CURSor<cu>:SIAD....1184
CURSor<cu>:SMODe....1177
CURSor<cu>:SOURce....1178
CURSor<cu>:SSCReen....1182
CURSor<cu>:SSOurce....1180
CURSor<cu>:STATe....1177
CURSor<cu>:STYLe....1189
CURSor<cu>:TRACking[:STATe]....1183
CURSor<cu>:USSource....1180
CURSor<cu>:VERTical<n>:LABel....1184
CURSor<cu>:X1ENvelope....1185
CURSor<cu>:X1Position....1180
CURSor<cu>:X2ENvelope....1185
CURSor<cu>:X2Position....1181
CURSor<cu>:XCOupling....1185
CURSor<cu>:XDELta:INVerse?....1186
CURSor<cu>:XDELta[:VALue]?....1186
CURSor<cu>:Y1Position....1181
CURSor<cu>:Y2Position....1182
CURSor<cu>:YCOupling....1185

CURSor<cu>:YDELta:SLOPe?....1187
CURSor<cu>:YDELta[:VALue]?....1187
DIGital<m>:DATA:HEADer?....1147
DIGital<m>:DATA[:VALues]?....1147
DIGital<m>:HYSTeresis....2015
DIGital<m>:LABel....2016
DIGital<m>:PROBe<ch>[:ENABle]?....2015
DIGital<m>:SIZE....2016
DIGital<m>:SKEW....2016
DIGital<m>:STATe....2016
DIGital<m>:TECHnology....2017
DIGital<m>:THCoupling....2017
DIGital<m>:THReshold....2017
DISPlay:ANNotation:ARRow<m>:CLEar....970
DISPlay:ANNotation:ARRow<m>:COLor....970
DISPlay:ANNotation:ARRow<m>:DIRection....970
DISPlay:ANNotation:ARRow<m>:HEIGht....970
DISPlay:ANNotation:ARRow<m>:HORizontal:POSition....971
DISPlay:ANNotation:ARRow<m>:REMove....971
DISPlay:ANNotation:ARRow<m>:VERTical:POSition....971
DISPlay:ANNotation:ARRow<m>:WIDTh....972
DISPlay:ANNotation:ARRow<m>[:VALue]....972
DISPlay:ANNotation:CLEar....966
DISPlay:ANNotation:COLor<m>:CATalog?....966
DISPlay:ANNotation:PLINe<m>:CLEar....974
DISPlay:ANNotation:PLINe<m>:COLor....975
DISPlay:ANNotation:PLINe<m>:EXTend....975
DISPlay:ANNotation:PLINe<m>:REMove....975
DISPlay:ANNotation:PLINe<m>[:VALue]....976
DISPlay:ANNotation:RECTangle<m>:CLEar....967
DISPlay:ANNotation:RECTangle<m>:COLor....967
DISPlay:ANNotation:RECTangle<m>:HEIGht....967
DISPlay:ANNotation:RECTangle<m>:HORizontal:POSition....968
DISPlay:ANNotation:RECTangle<m>:REMove....968
DISPlay:ANNotation:RECTangle<m>:VERTical:POSition....968
DISPlay:ANNotation:RECTangle<m>:WIDTh....969
DISPlay:ANNotation:RECTangle<m>[:VALue]....969
DISPlay:ANNotation:TEXT<m>:CLEar....972
DISPlay:ANNotation:TEXT<m>:COLor....973
DISPlay:ANNotation:TEXT<m>:FONTsize....973
DISPlay:ANNotation:TEXT<m>:HORizontal:POSition....973
DISPlay:ANNotation:TEXT<m>:REMove....974
DISPlay:ANNotation:TEXT<m>:VERTical:POSition....974
DISPlay:ANNotation:TEXT<m>[:VALue]....974
DISPlay:ANNotation:TRANsparency....966
DISPlay:BACKlight:DIMMing....984
DISPlay:BACKlight[:BRIGtness]....983
DISPlay:CLR....984
DISPlay:COLor:SIGNal:ASSign....977
DISPlay:COLor:SIGNal:CATalog?....976

DISPlay:COLor:SIGNal:COLor ... 977
DISPlay:COLor:SIGNal:USE ... 977
DISPlay:DIAGram:CROSshair ... 978
DISPlay:DIAGram:FINegrid ... 978
DISPlay:DIAGram:GRID ... 978
DISPlay:DIAGram:LABels ... 979
DISPlay:DIAGram:STYLe ... 983
DISPlay:DIAGram:XFIXed ... 979
DISPlay:DIAGram:XPDivlabel ... 979
DISPlay:DIAGram:YFIXed ... 979
DISPlay:DIALog:FONTsize ... 980
DISPlay:DIALog:TRANsparency ... 980
DISPlay:INTensity ... 983
DISPlay:MEASurements:LEVels ... 981
DISPlay:PERSistence:INFinite ... 982
DISPlay:PERSistence:RESet ... 982
DISPlay:PERSistence:TIME ... 982
DISPlay:PERSistence[:STATe] ... 982
DISPlay:RESult:FONTsize ... 980
DISPlay:SIGNal:LABel ... 1002
DISPlay:TOOLbar:COUNt? ... 963
DISPlay:TOOLbar:DESelect ... 963
DISPlay:TOOLbar:RESTore ... 964
EXPort:HISTogram<m>:DATA:HEADer? ... 1149
EXPort:HISTogram<m>:DATA[:VALues]? ... 1150
EXPort:HISTogram<m>:NAME ... 1133
EXPort:HISTogram<m>:NORMalize ... 1132
EXPort:HISTogram<m>:SAVE ... 1133
EXPort:HMEasurement<m>:DATA:HEADer? ... 1149
EXPort:HMEasurement<m>:DATA[:VALues]? ... 1150
EXPort:HMEasurement<m>:NAME ... 1133
EXPort:HMEasurement<m>:NORMalize ... 1132
EXPort:HMEasurement<m>:SAVE ... 1133
EXPort:RESult:AUTonaming:NAME ... 1135
EXPort:RESult:AUTonaming:PATH ... 1136
EXPort:RESult:AUTonaming:TYPE ... 1136
EXPort:RESult:NAME ... 1136
EXPort:RESult:SAVE ... 1136
EXPort:RESult:SELect:CURSor ... 1136
EXPort:RESult:SELect:FRA:MARGin ... 1245
EXPort:RESult:SELect:FRA:MARKer ... 1245
EXPort:RESult:SELect:FRA:RESult ... 1245
EXPort:RESult:SELect:MEASurement ... 1137
EXPort:RESult:SELect:P3PHase<m> ... 1314
EXPort:RESult:SELect:PLISt ... 1137
EXPort:RESult:SELect:POWer<m> ... 1249
EXPort:WAVeform:ABORt ... 1127
EXPort:WAVeform:AUTonaming:NAME ... 1127
EXPort:WAVeform:AUTonaming:PATH ... 1127
EXPort:WAVeform:AUTonaming:TYPE ... 1128

EXPort:WAVeform:CURSorset....1128
EXPort:WAVeform:DATA[:VALues]?....1150
EXPort:WAVeform:GATE....1128
EXPort:WAVeform:NAME....1128
EXPort:WAVeform:SAVE....1129
EXPort:WAVeform:SCOPe....1129
EXPort:WAVeform:SOURce....1130
EXPort:WAVeform:STARt....1131
EXPort:WAVeform:STOP....1132
FORMat:BORDer....951
FORMat:BPATtern....952
FORMat[:DATA]....950
FRANalysis:AMPLitude:ENABle....1233
FRANalysis:AMPLitude:MODE....1229
FRANalysis:AMPLitude:OFFSet....1233
FRANalysis:AMPLitude:PROFile:APOint....1231
FRANalysis:AMPLitude:PROFile:COUNt....1231
FRANalysis:AMPLitude:PROFile:MODE....1231
FRANalysis:AMPLitude:PROFile:POINt<m>:AMPLitude....1231
FRANalysis:AMPLitude:PROFile:POINt<m>:FREQuency....1231
FRANalysis:AMPLitude:PROFile:POINt<m>:REMove....1232
FRANalysis:AMPLitude:PROFile:SORT....1232
FRANalysis:AMPLitude:SCALe....1233
FRANalysis:AUToscale....1223
FRANalysis:AUToscale:RESult[:ENABle]....1223
FRANalysis:CALibration:CALibration....1236
FRANalysis:CALibration:RESult?....1236
FRANalysis:CALibration:STATe....1236
FRANalysis:ENABle....1223
FRANalysis:FREQuency:DATA?....1246
FRANalysis:FREQuency:MODE....1224
FRANalysis:FREQuency:STARt....1224
FRANalysis:FREQuency:STOP....1224
FRANalysis:GAIN:DATA?....1246
FRANalysis:GAIN:ENABle....1233
FRANalysis:GAIN:OFFSet....1233
FRANalysis:GAIN:SCALe....1234
FRANalysis:GENerator:AMPLitude....1230
FRANalysis:GENerator:LOAD....1230
FRANalysis:GENerator[:CHANnel]....1230
FRANalysis:HDEFinition[:ENABle]....1224
FRANalysis:IMPedance:DATA?....1246
FRANalysis:IMPedance:ENABle....1234
FRANalysis:IMPedance:SCALe....1234
FRANalysis:IMPedance:STARt....1234
FRANalysis:IMPedance:STOP....1234
FRANalysis:IMPedance[:MODE]....1225
FRANalysis:INPut[:SOURce]....1225
FRANalysis:MARGin:GAIN:FREQuency?....1237
FRANalysis:MARGin:GAIN:VALue?....1237

FRANalysis:MARGin:PHASe:FREQuency? 1237
FRANalysis:MARGin:PHASe:VALue? 1237
FRANalysis:MARGin:STATe 1237
FRANalysis:MARKer<m>:DIFFerence:FREQuency? 1238
FRANalysis:MARKer<m>:DIFFerence:GAIN? 1239
FRANalysis:MARKer<m>:DIFFerence:IMPedance? 1239
FRANalysis:MARKer<m>:DIFFerence:PHASe? 1239
FRANalysis:MARKer<m>:DIFFerence:REFerence<rc>? 1244
FRANalysis:MARKer<m>:FREQuency 1238
FRANalysis:MARKer<m>:GAIN? 1238
FRANalysis:MARKer<m>:IMPedance? 1238
FRANalysis:MARKer<m>:INDex? 1239
FRANalysis:MARKer<m>:PHASe? 1238
FRANalysis:MARKer<m>:REFerence<rc>? 1244
FRANalysis:MARKer<m>:SSCReen 1240
FRANalysis:MARKer<m>:STATe 1240
FRANalysis:MEASurement:DELay:MODE 1225
FRANalysis:MEASurement:DELay:PERiod[:NUMBer] 1226
FRANalysis:MEASurement:DELay:STATe 1225
FRANalysis:MEASurement:DELay[:TIME] 1226
FRANalysis:MEASurement:POINt[:DISPlay] 1226
FRANalysis:MEASurement:RBW 1227
FRANalysis:MODE 1227
FRANalysis:OUTPut[:SOURce] 1227
FRANalysis:PHASe:DATA? 1247
FRANalysis:PHASe:ENABle 1235
FRANalysis:PHASe:MAXimum 1227
FRANalysis:PHASe:OFFSet 1235
FRANalysis:PHASe:SCALe 1235
FRANalysis:POINts:LOGarithmic 1228
FRANalysis:POINts:MODE 1228
FRANalysis:POINts:TOTal 1228
FRANalysis:REFCurve<rc>:CLEar 1241
FRANalysis:REFCurve<rc>:DATA? 1241
FRANalysis:REFCurve<rc>:ENABle 1241
FRANalysis:REFCurve<rc>:NAME 1242
FRANalysis:REFCurve<rc>:OFFSet 1242
FRANalysis:REFCurve<rc>:OPEN 1242
FRANalysis:REFCurve<rc>:SAVE 1243
FRANalysis:REFCurve<rc>:SCALe 1243
FRANalysis:REFCurve<rc>:SOURce 1243
FRANalysis:REFCurve<rc>:TOORiginal 1243
FRANalysis:REFCurve<rc>:UPDate 1244
FRANalysis:REFCurve<rc>:VMODe 1244
FRANalysis:REPeat 1229
FRANalysis:RESet 1229
FRANalysis:RESult:STATe 1240
FRANalysis:STATe 1229
GATE<g>:ABSolute:STARt 1080
GATE<g>:ABSolute:STOP 1080

GATE<g>:CURSor....1079
GATE<g>:ENABle....1078
GATE<g>:GCOupling....1078
GATE<g>:MODE....1080
GATE<g>:RELative:STARt....1080
GATE<g>:RELative:STOP....1080
GATE<g>:SHOW....1081
GATE<g>:ZDIagram....1079
GENerator:RCL....1126
GENerator:SAV....1126
HCOPy:DATA?....1140
HCOPy:DESTination<m>....1138
HCOPy:DEVice<m>:INVerse....1138
HCOPy:DEVice<m>:LANGuage....1138
HCOPy:IMMediate<m>....1139
HCOPy:IMMediate<m>:NEXT....1139
HCOPy:ISBA....1140
HCOPy:SSD....1140
HCOPy:WBKG....1140
HDEFinition:BFTYpe....1029
HDEFinition:BWIDth....1029
HDEFinition:RESolution?....1029
HDEFinition:STATe....1030
HISTogram<m>:ADD....1114
HISTogram<m>:COUNt?....1114
HISTogram<m>:ENABle....1114
HISTogram<m>:MODE....1115
HISTogram<m>:REMove....1115
HISTogram<m>:RESet....1115
HISTogram<m>:SOURce....1115
HISTogram<m>:WINDow:ENABle....1116
HISTogram<m>:WINDow:HORizontal:ABSolute:POSition....1117
HISTogram<m>:WINDow:HORizontal:ABSolute:SPAN....1117
HISTogram<m>:WINDow:HORizontal:ABSolute:STARt....1117
HISTogram<m>:WINDow:HORizontal:ABSolute:STOP....1117
HISTogram<m>:WINDow:HORizontal:MODE....1116
HISTogram<m>:WINDow:HORizontal:RELative:POSition....1117
HISTogram<m>:WINDow:HORizontal:RELative:SPAN....1117
HISTogram<m>:WINDow:HORizontal:RELative:STARt....1117
HISTogram<m>:WINDow:HORizontal:RELative:STOP....1117
HISTogram<m>:WINDow:VERTical:ABSolute:POSition....1118
HISTogram<m>:WINDow:VERTical:ABSolute:SPAN....1118
HISTogram<m>:WINDow:VERTical:ABSolute:STARt....1118
HISTogram<m>:WINDow:VERTical:ABSolute:STOP....1118
HISTogram<m>:WINDow:VERTical:MODE....1116
HISTogram<m>:WINDow:VERTical:RELative:POSition....1118
HISTogram<m>:WINDow:VERTical:RELative:SPAN....1118
HISTogram<m>:WINDow:VERTical:RELative:STARt....1118
HISTogram<m>:WINDow:VERTical:RELative:STOP....1118
LAYout<ly>:ACTive....958

LAYout<ly>:COUNt? 958
LAYout<ly>:DIAGram<da>:COUNt? 959
LAYout<ly>:DIAGram<da>:LABel 960
LAYout<ly>:DIAGram<da>:SOURce 960
LAYout<ly>:DIAGram<da>[:ENABle] 959
LAYout<ly>:LABel 959
LAYout<ly>:NODE<no>:CHILdren<o>:CONTent<p>:ID 961
LAYout<ly>:NODE<no>:CHILdren<o>:CONTent<p>:TYPE 961
LAYout<ly>:NODE<no>:COUNt? 960
LAYout<ly>:NODE<no>:RATio 962
LAYout<ly>:NODE<no>:STYPe 962
LAYout<ly>:NODE<no>[:ENABle] 960
LAYout<ly>:RESult<n>:HORizontal:RATio 962
LAYout<ly>:RESult<n>:VERTical:RATio 963
LAYout<ly>:RPOSition 963
LAYout<ly>:SACTive 958
LAYout<ly>:ZOOM<zo>:COUNt? 1090
LAYout<ly>:ZOOM<zo>:DIAG? 1090
LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:POSition 1090
LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:SPAN 1091
LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:STARt 1091
LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:STOP 1091
LAYout<ly>:ZOOM<zo>:HORizontal:ABSolute:WIDTh 1092
LAYout<ly>:ZOOM<zo>:HORizontal:MODE 1092
LAYout<ly>:ZOOM<zo>:HORizontal:RELative:POSition 1092
LAYout<ly>:ZOOM<zo>:HORizontal:RELative:SPAN 1093
LAYout<ly>:ZOOM<zo>:HORizontal:RELative:STARt 1093
LAYout<ly>:ZOOM<zo>:HORizontal:RELative:STOP 1094
LAYout<ly>:ZOOM<zo>:HORizontal:RELative:WIDTh 1094
LAYout<ly>:ZOOM<zo>:SOURce 1094
LAYout<ly>:ZOOM<zo>:SSCReen 1098
LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:POSition 1095
LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:RANGe 1095
LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:SPAN 1095
LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:STARt 1095
LAYout<ly>:ZOOM<zo>:VERTical:ABSolute:STOP 1096
LAYout<ly>:ZOOM<zo>:VERTical:MODE 1096
LAYout<ly>:ZOOM<zo>:VERTical:RELative:POSition 1096
LAYout<ly>:ZOOM<zo>:VERTical:RELative:SPAN 1097
LAYout<ly>:ZOOM<zo>:VERTical:RELative:STARt 1097
LAYout<ly>:ZOOM<zo>:VERTical:RELative:STOP 1097
LAYout<ly>:ZOOM<zo>:VERTical:RELative:WIDTh 1097
LAYout<ly>:ZOOM<zo>[:ENABle] 1090
LAYout<ly>[:ENABle] 958
MEASurement<mg>:ACTive 1152
MEASurement<mg>:AMPTime:CSLope 1157
MEASurement<mg>:AMPTime:DELay<n>:DIRection 1158
MEASurement<mg>:AMPTime:DELay<n>:SLOPe 1159
MEASurement<mg>:AMPTime:DTOTrigger<n>:SLOPe 1158
MEASurement<mg>:AMPTime:ESLope 1157

MEASurement<mg>:AMPTime:PHASe:UNIT ..... 1159
MEASurement<mg>:AMPTime:PSLope ..... 1158
MEASurement<mg>:AMPTime:PTCount ..... 1157
MEASurement<mg>:CLEar ..... 1155
MEASurement<mg>:COUNt? ..... 1152
MEASurement<mg>:DISPlay:REFLevel<rl>:LEVels ..... 1155
MEASurement<mg>:DISPlay:RESults ..... 1155
MEASurement<mg>:ENVSelect ..... 1156
MEASurement<mg>:FSRC ..... 1153
MEASurement<mg>:GATE ..... 1155
MEASurement<mg>:HISTogram:PROBability:LIMit ..... 1160
MEASurement<mg>:HISTogram:PROBability:TYPE ..... 1160
MEASurement<mg>:IMPRecise:STATus:CONDition? ..... 2050
MEASurement<mg>:IMPRecise:STATus:ENABle ..... 2051
MEASurement<mg>:IMPRecise:STATus:NTRansition ..... 2052
MEASurement<mg>:IMPRecise:STATus:PTRansition ..... 2052
MEASurement<mg>:IMPRecise:STATus[:EVENt]? ..... 2050
MEASurement<mg>:JITTer:NCYCles ..... 1161
MEASurement<mg>:JITTer:POLarity ..... 1162
MEASurement<mg>:JITTer:REFLevel<rl> ..... 1162
MEASurement<mg>:JITTer:RELPolarity ..... 1163
MEASurement<mg>:JITTer:SLOPe ..... 1163
MEASurement<mg>:JITTer:TREF ..... 1164
MEASurement<mg>:JITTer:UNIT ..... 1164
MEASurement<mg>:MAIN ..... 1154
MEASurement<mg>:MNOMeas ..... 1170
MEASurement<mg>:MULTiple ..... 1170
MEASurement<mg>:PROTocol:F2Name ..... 1165
MEASurement<mg>:PROTocol:FD2Name ..... 1166
MEASurement<mg>:PROTocol:FD2Value ..... 1166
MEASurement<mg>:PROTocol:FDName ..... 1165
MEASurement<mg>:PROTocol:FDValue ..... 1165
MEASurement<mg>:PROTocol:FNAMe ..... 1164
MEASurement<mg>:REFLevel<rl>:REFerence ..... 1159
MEASurement<mg>:REFLevel<rl>:RESult:LOWer? ..... 1077
MEASurement<mg>:REFLevel<rl>:RESult:MIDDle? ..... 1077
MEASurement<mg>:REFLevel<rl>:RESult:SIGHigh? ..... 1078
MEASurement<mg>:REFLevel<rl>:RESult:SIGLow? ..... 1078
MEASurement<mg>:REFLevel<rl>:RESult:UPPer? ..... 1077
MEASurement<mg>:RESult:AVG? ..... 1166
MEASurement<mg>:RESult:EVENts:COUNt? ..... 1167
MEASurement<mg>:RESult:EVENts:STARt? ..... 1168
MEASurement<mg>:RESult:EVENts:STOP? ..... 1168
MEASurement<mg>:RESult:EVENts:VALue? ..... 1168
MEASurement<mg>:RESult:EVTCount? ..... 1166
MEASurement<mg>:RESult:NPEak? ..... 1166
MEASurement<mg>:RESult:PPEak? ..... 1167
MEASurement<mg>:RESult:RMS? ..... 1167
MEASurement<mg>:RESult:RSTDdev? ..... 1167
MEASurement<mg>:RESult:STARt? ..... 1167

MEASurement<mg>:RESult:STDDev? 1167
MEASurement<mg>:RESult:STOP? 1167
MEASurement<mg>:RESult:WFMCount? 1167
MEASurement<mg>:RESult[:ACTual]? 1166
MEASurement<mg>:SOURce 1152
MEASurement<mg>:SSRC 1154
MEASurement<mg>:STATistics:ARESet 1169
MEASurement<mg>:STATistics:HISTogram:AUToscale 1173
MEASurement<mg>:STATistics:HISTogram:HBINs 1174
MEASurement<mg>:STATistics:HISTogram:HORizontal:OFFSet 1174
MEASurement<mg>:STATistics:HISTogram:HORizontal:SCALe 1175
MEASurement<mg>:STATistics:HISTogram:RESet 1175
MEASurement<mg>:STATistics:HISTogram[:ENABle] 1174
MEASurement<mg>:STATistics:RESet 1155
MEASurement<mg>:STATistics[:ENABle] 1169
MEASurement<mg>:TRACk:AUTO 1171
MEASurement<mg>:TRACk:CONTiunous 1171
MEASurement<mg>:TRACk:DATA:HEADer? 1172
MEASurement<mg>:TRACk:DATA[:VALues]? 1173
MEASurement<mg>:TRACk:OFFSet 1171
MEASurement<mg>:TRACk:SCALe 1172
MEASurement<mg>:TRACk[:STATe] 1170
MEASurement<mg>[:ENABle] 1152
METer:BANDwidth 1357
METer:DVMeter<m>:ACRMs:ENABle 1358
METer:DVMeter<m>:ACRMs:RESult? 1358
METer:DVMeter<m>:DC:ENABle 1358
METer:DVMeter<m>:DC:RESult? 1358
METer:DVMeter<m>:DCRMs:ENABle 1359
METer:DVMeter<m>:DCRMs:RESult? 1359
METer:DVMeter<m>:ENABle? 1357
METer:MTIMe 1359
MMEMory:ATTRibute 1124
MMEMory:AUSave:ENABle 985
MMEMory:AUSave:INTerval 985
MMEMory:AUTonaming:DEFaultpath 986
MMEMory:AUTonaming:INDex 985
MMEMory:AUTonaming:PREFix 985
MMEMory:AUTonaming:RESall 986
MMEMory:AUTonaming:RESPath 986
MMEMory:AUTonaming:TEXT 986
MMEMory:AUTonaming:TIME 985
MMEMory:AUTonaming:USERtext 985
MMEMory:CATalog:LENGth? 1122
MMEMory:CATalog? 1121
MMEMory:CDIRectory 1121
MMEMory:COPY 1122
MMEMory:DATA 1123
MMEMory:DCATalog:LENGth? 1120
MMEMory:DCATalog? 1120

MMEMory:DELete....1123
MMEMory:DRIVes?....1120
MMEMory:GENerator:RCL....1125
MMEMory:GENerator:SAV....1126
MMEMory:LOAD:STATe....1125
MMEMory:MDIRectory....1121
MMEMory:MOVE....1123
MMEMory:NAME....1138
MMEMory:NDRive<m>:CONNect....955
MMEMory:NDRive<m>:DCONnect....956
MMEMory:NDRive<m>:LPATh....956
MMEMory:NDRive<m>:NPATh....956
MMEMory:NDRive<m>:PWORd....956
MMEMory:NDRive<m>:UNAMe....957
MMEMory:RCL....1125
MMEMory:RDIRectory....1121
MMEMory:SAV....1124
MMEMory:STORe:STATe....1125
MTESt<m>:ADD....1209
MTESt<m>:AUTomask:UPDate....1216
MTESt<m>:AUTomask:VALues:BOTTom?....1218
MTESt<m>:AUTomask:VALues:TOP?....1218
MTESt<m>:AUTomask:XTOLerance:DIVisions....1216
MTESt<m>:AUTomask:XTOLerance:PHYSical....1216
MTESt<m>:AUTomask:XTUNit....1217
MTESt<m>:AUTomask:YTOLerance:DIVisions....1217
MTESt<m>:AUTomask:YTOLerance:PHYSical....1217
MTESt<m>:AUTomask:YTUNit....1217
MTESt<m>:COUNt?....1209
MTESt<m>:DIAGram....1210
MTESt<m>:IMEXport:NAME....1210
MTESt<m>:IMEXport:OPEN....1210
MTESt<m>:IMEXport:SAVE....1210
MTESt<m>:ONViolation:BEEP....1218
MTESt<m>:ONViolation:SCReenshot....1219
MTESt<m>:ONViolation:SRESults....1219
MTESt<m>:ONViolation:STOP....1219
MTESt<m>:ONViolation:TRIGgerout....1220
MTESt<m>:ONViolation:WFMSave....1220
MTESt<m>:REMove....1211
MTESt<m>:RESult:COUNt:FWAVeforms?....1221
MTESt<m>:RESult:COUNt:PWAVeforms?....1221
MTESt<m>:RESult:COUNt:WAVeforms?....1221
MTESt<m>:RESult:FRATe?....1221
MTESt<m>:RESult[:RESult]?....1222
MTESt<m>:SEGMent<n>:ADD....1212
MTESt<m>:SEGMent<n>:COUNt?....1212
MTESt<m>:SEGMent<n>:POINt<o>:ADD....1212
MTESt<m>:SEGMent<n>:POINt<o>:COUNt?....1213
MTESt<m>:SEGMent<n>:POINt<o>:REMove....1213

MTESt<m>:SEGMent<n>:POINt<o>:VALid? 1213
MTESt<m>:SEGMent<n>:POINt<o>:X 1214
MTESt<m>:SEGMent<n>:POINt<o>:Y 1214
MTESt<m>:SEGMent<n>:REMove 1215
MTESt<m>:SEGMent<n>:VALid? 1215
MTESt<m>:SOURce 1211
MTESt<m>:STATe 1211
MTESt<m>:VISible 1211
P3PHase<m>:EFFiciency:DISPlay:WAVeform:INPut:TOTal:POWer[:ENABle] 1346
P3PHase<m>:EFFiciency:DISPlay:WAVeform:OUTPut:TOTal:POWer[:ENABle] 1346
P3PHase<m>:EFFiciency:GATE 1345
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:HYSTeresis 1350
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:LLEVel 1350
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:MLEVel 1351
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:ULEVel 1351
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:LMODe 1351
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:HYSTeresis 1352
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:LOWer 1352
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MIDDle 1353
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MODE 1353
P3PHase<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:UPPer 1354
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W1V1a 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W2V2a 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W3V3a 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CURRent:W3VN3a 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CYCLe:FILTer 1345
P3PHase<m>:EFFiciency:INPut<n>:SOURce:CYCLe:SOURce 1344
P3PHase<m>:EFFiciency:INPut<n>:SOURce:LINes:W2V2a 1343
P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W1V1a 1344
P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W2V2a 1344
P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W3V3a 1344
P3PHase<m>:EFFiciency:INPut<n>:SOURce:VOLTage:W3VN3a 1344
P3PHase<m>:EFFiciency:INPut<n>:SOURce:WIRing 1342
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:DATA:HEADer? 1355
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:DATA[:VALues]? 1356
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:VERTical:OFFSet 1347
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:VERTical:SCALe:MODE 1348
P3PHase<m>:EFFiciency:INPut<n>:TOTal:POWer:VERTical:SCALe[:VALue] 1348
P3PHase<m>:EFFiciency:INPut<n>:TYPE 1342
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:HYSTeresis 1350
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:LLEVel 1350
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:MLEVel 1351
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:ULEVel 1351
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:LMODe 1351
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:HYSTeresis 1352
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:LOWer 1352
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MIDDle 1353
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MODE 1353
P3PHase<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:UPPer 1354
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W1V1a 1343

P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W2V2a....1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W3V3a....1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CURRent:W3VN3a....1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CYCLe:FILTer....1345
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:CYCLe:SOURce....1344
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:LINes:W2V2a....1343
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W1V1a....1344
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W2V2a....1344
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W3V3a....1344
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:VOLTage:W3VN3a....1344
P3PHase<m>:EFFiciency:OUTPut<n>:SOURce:WIRing....1342
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:DATA:HEADer?....1355
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:DATA[:VALues]?....1356
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:VERTical:OFFSet....1347
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:VERTical:SCALe:MODE....1348
P3PHase<m>:EFFiciency:OUTPut<n>:TOTal:POWer:VERTical:SCALe[:VALue]....1348
P3PHase<m>:EFFiciency:OUTPut<n>:TYPE....1342
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:AVERage?....1338
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:EVTCount?....1339
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:NPEak?....1339
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:PPEak?....1339
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:RMS?....1340
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:STDDev?....1340
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer:WFMCount?....1340
P3PHase<m>:EFFiciency:RESult:INPut<n>:TOTal:POWer[:ACTual]?....1338
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:AVERage?....1338
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:EVTCount?....1339
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:NPEak?....1339
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:PPEak?....1339
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:RMS?....1340
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:STDDev?....1340
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer:WFMCount?....1340
P3PHase<m>:EFFiciency:RESult:OUTPut<n>:TOTal:POWer[:ACTual]?....1338
P3PHase<m>:EFFiciency:STATistic:ENABle....1337
P3PHase<m>:EFFiciency:STATistic:RESet....1337
P3PHase<m>:EFFiciency:STATistic:WFMCount?....1337
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:AVERage?....1338
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:EVTCount?....1339
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:NPEak?....1339
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:PPEak?....1339
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:RMS?....1340
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:STDDev?....1340
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult:WFMCount?....1340
P3PHase<m>:EFFiciency:TOTal:EFFiciency:RESult[:ACTual]?....1338
P3PHase<m>:HARMonics:AVAilable<n>....1334
P3PHase<m>:HARMonics:DISPlay:BARGraph:VERTical:SCALe:TYPE....1333
P3PHase<m>:HARMonics:DISPlay:BARGraph[:ENABle]....1332
P3PHase<m>:HARMonics:DISPlay:HARMonics:STARt[:VALue]....1333
P3PHase<m>:HARMonics:DISPlay:HARMonics:STOP[:VALue]....1333
P3PHase<m>:HARMonics:DISPlay:HARMonics[:DISPlay]....1332

P3PHase<m>:HARMonics:DISPlay:POWer<n>:WAVeform[:ENABle] 1346
P3PHase<m>:HARMonics:DISPlay:TOTal:POWer:WAVeform[:ENABle] 1346
P3PHase<m>:HARMonics:FREQuency:EN 1330
P3PHase<m>:HARMonics:FREQuency:MIL 1330
P3PHase<m>:HARMonics:MEASurement:CSOurce:FREQuency<n>[:ACTual]? 1334
P3PHase<m>:HARMonics:MEASurement:POWer<n>:ACTive[:ACTual]? 1334
P3PHase<m>:HARMonics:MEASurement:THDFundament<n>[:ACTual]? 1335
P3PHase<m>:HARMonics:MEASurement:THDRms<n>[:ACTual]? 1335
P3PHase<m>:HARMonics:PFACtor:MODE 1331
P3PHase<m>:HARMonics:PFACtor:USER 1331
P3PHase<m>:HARMonics:POWer<n>:DATA:HEADer? 1355
P3PHase<m>:HARMonics:POWer<n>:DATA[:VALues]? 1356
P3PHase<m>:HARMonics:POWer<n>:VERTical:OFFSet 1346
P3PHase<m>:HARMonics:POWer<n>:VERTical:SCALe:MODE 1347
P3PHase<m>:HARMonics:POWer<n>:VERTical:SCALe[:VALue] 1348
P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:HYSTeresis 1350
P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:LLEVel 1350
P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:MLEVel 1351
P3PHase<m>:HARMonics:REFLevel<rl>:ABSolute:ULEVel 1351
P3PHase<m>:HARMonics:REFLevel<rl>:LMODe 1351
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:HYSTeresis 1352
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:LOWer 1352
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:MIDDle 1353
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:MODE 1353
P3PHase<m>:HARMonics:REFLevel<rl>:RELative:UPPer 1354
P3PHase<m>:HARMonics:RESult<n>:MAGNitude<o>:PERCent[:ACTual]? 1336
P3PHase<m>:HARMonics:RESult<n>:MAGNitude<o>:RMS[:ACTual]? 1336
P3PHase<m>:HARMonics:RESult<n>:VIOLation<o>? 1335
P3PHase<m>:HARMonics:RESult<n>[:FREQuency]? 1335
P3PHase<m>:HARMonics:REVision 1330
P3PHase<m>:HARMonics:RPOWer:MODE 1331
P3PHase<m>:HARMonics:RPOWer:USER 1332
P3PHase<m>:HARMonics:SOURce:CURRent:W2V2a 1343
P3PHase<m>:HARMonics:SOURce:CURRent:W3V3a 1343
P3PHase<m>:HARMonics:SOURce:CURRent:W3VN3a 1343
P3PHase<m>:HARMonics:SOURce:CYCLe:FILTer 1345
P3PHase<m>:HARMonics:SOURce:CYCLe[:SOURce] 1344
P3PHase<m>:HARMonics:SOURce:INOut 1342
P3PHase<m>:HARMonics:SOURce:LINes:W2V2a 1343
P3PHase<m>:HARMonics:SOURce:VOLTage:W2V2a 1344
P3PHase<m>:HARMonics:SOURce:VOLTage:W3V3a 1344
P3PHase<m>:HARMonics:SOURce:VOLTage:W3VN3a 1344
P3PHase<m>:HARMonics:SOURce:WIRing 1342
P3PHase<m>:HARMonics:STANdard 1329
P3PHase<m>:HARMonics:TOTal:POWer:DATA:HEADer? 1355
P3PHase<m>:HARMonics:TOTal:POWer:DATA[:VALues]? 1356
P3PHase<m>:HARMonics:TOTal:POWer:VERTical:OFFSet 1347
P3PHase<m>:HARMonics:TOTal:POWer:VERTical:SCALe:MODE 1348
P3PHase<m>:HARMonics:TOTal:POWer:VERTical:SCALe[:VALue] 1348
P3PHase<m>:QUALity:DISPlay:CURRent:CRESTfactor<n>[:ENABle] 1317

P3PHase<m>:QUALity:DISPlay:CURRent:PEAK<n>[:ENABle] 1317
P3PHase<m>:QUALity:DISPlay:CURRent:RMS<n>[:ENABle] 1318
P3PHase<m>:QUALity:DISPlay:POWer<n>:ACTive[:ENABle] 1318
P3PHase<m>:QUALity:DISPlay:POWer<n>:APParent[:ENABle] 1318
P3PHase<m>:QUALity:DISPlay:POWer<n>:PFACtor[:ENABle] 1319
P3PHase<m>:QUALity:DISPlay:POWer<n>:PHASe[:ENABle] 1319
P3PHase<m>:QUALity:DISPlay:POWer<n>:REACtive[:ENABle] 1319
P3PHase<m>:QUALity:DISPlay:POWer<n>:WAVeform[:ENABle] 1346
P3PHase<m>:QUALity:DISPlay:TOTal:POWer:WAVeform[:ENABle] 1346
P3PHase<m>:QUALity:DISPlay:VOLTage:CREStfactor<n>[:ENABle] 1320
P3PHase<m>:QUALity:DISPlay:VOLTage:FREQuency<n>[:ENABle] 1320
P3PHase<m>:QUALity:DISPlay:VOLTage:PEAK<n>[:ENABle] 1320
P3PHase<m>:QUALity:DISPlay:VOLTage:RMS<n>[:ENABle] 1320
P3PHase<m>:QUALity:FREQuency:USER 1315
P3PHase<m>:QUALity:FREQuency[:VALue] 1315
P3PHase<m>:QUALity:GATE 1345
P3PHase<m>:QUALity:PHASor:CURRent:MAXimum 1316
P3PHase<m>:QUALity:PHASor:CURRent:OFFSet 1316
P3PHase<m>:QUALity:PHASor:FSCReen 1317
P3PHase<m>:QUALity:PHASor:RESult:VOLTage<n>:ANGLe[:ACTual]? 1329
P3PHase<m>:QUALity:PHASor:VOLTage:MAXimum 1316
P3PHase<m>:QUALity:PHASor:VOLTage:OFFSet 1316
P3PHase<m>:QUALity:PHASor[:ENABle] 1316
P3PHase<m>:QUALity:POWer<n>:DATA:HEADer? 1355
P3PHase<m>:QUALity:POWer<n>:DATA[:VALues]? 1356
P3PHase<m>:QUALity:POWer<n>:VERTical:OFFSet 1346
P3PHase<m>:QUALity:POWer<n>:VERTical:SCALe:MODE 1347
P3PHase<m>:QUALity:POWer<n>:VERTical:SCALe[:VALue] 1348
P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:HYSTeresis 1350
P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:LLEVel 1350
P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:MLEVel 1351
P3PHase<m>:QUALity:REFLevel<rl>:ABSolute:ULEVel 1351
P3PHase<m>:QUALity:REFLevel<rl>:LMODe 1351
P3PHase<m>:QUALity:REFLevel<rl>:RELative:HYSTeresis 1352
P3PHase<m>:QUALity:REFLevel<rl>:RELative:LOWer 1352
P3PHase<m>:QUALity:REFLevel<rl>:RELative:MIDDle 1353
P3PHase<m>:QUALity:REFLevel<rl>:RELative:MODE 1353
P3PHase<m>:QUALity:REFLevel<rl>:RELative:UPPer 1354
P3PHase<m>:QUALity:RESult:CSOurce:FREQuency<n>[:ACTual]? 1321
P3PHase<m>:QUALity:RESult:CURRent:CREStfactor<n>[:ACTual]? 1322
P3PHase<m>:QUALity:RESult:CURRent:PEAK<n>[:ACTual]? 1322
P3PHase<m>:QUALity:RESult:CURRent:RMS<n>[:ACTual]? 1323
P3PHase<m>:QUALity:RESult:POWer:ACTive<n>[:ACTual]? 1324
P3PHase<m>:QUALity:RESult:POWer:APParent<n>[:ACTual]? 1324
P3PHase<m>:QUALity:RESult:POWer:PFACtor<n>[:ACTual]? 1325
P3PHase<m>:QUALity:RESult:POWer:PHASe<n>[:ACTual]? 1325
P3PHase<m>:QUALity:RESult:POWer:REACtive<n>[:ACTual]? 1326
P3PHase<m>:QUALity:RESult:TOTal:CURRent:CREStfactor<n>[:ACTual]? 1322
P3PHase<m>:QUALity:RESult:TOTal:CURRent:PEAK<n>[:ACTual]? 1323
P3PHase<m>:QUALity:RESult:TOTal:CURRent:RMS<n>[:ACTual]? 1323

P3PHase<m>:QUALity:RESult:TOTal:POWer:ACTive<n>[:ACTual]? 1324
P3PHase<m>:QUALity:RESult:TOTal:POWer:APParent<n>[:ACTual]? 1324
P3PHase<m>:QUALity:RESult:TOTal:POWer:PFACtor<n>[:ACTual]? 1325
P3PHase<m>:QUALity:RESult:TOTal:POWer:PHASe<n>[:ACTual]? 1326
P3PHase<m>:QUALity:RESult:TOTal:POWer:REACtive<n>[:ACTual]? 1326
P3PHase<m>:QUALity:RESult:TOTal:VOLTage:CREStfactor<n>[:ACTual]? 1327
P3PHase<m>:QUALity:RESult:TOTal:VOLTage:PEAK<n>[:ACTual]? 1328
P3PHase<m>:QUALity:RESult:TOTal:VOLTage:RMS<n>[:ACTual]? 1328
P3PHase<m>:QUALity:RESult:VOLTage:CREStfactor<n>[:ACTual]? 1327
P3PHase<m>:QUALity:RESult:VOLTage:PEAK<n>[:ACTual]? 1327
P3PHase<m>:QUALity:RESult:VOLTage:RMS<n>[:ACTual]? 1328
P3PHase<m>:QUALity:SOURce:CURRent:W2V2a 1343
P3PHase<m>:QUALity:SOURce:CURRent:W3V3a 1343
P3PHase<m>:QUALity:SOURce:CURRent:W3VN3a 1343
P3PHase<m>:QUALity:SOURce:CYCLe:FILTer 1345
P3PHase<m>:QUALity:SOURce:CYCLe[:SOURce] 1344
P3PHase<m>:QUALity:SOURce:INOut 1342
P3PHase<m>:QUALity:SOURce:LINes:W2V2a 1343
P3PHase<m>:QUALity:SOURce:VOLTage:W2V2a 1344
P3PHase<m>:QUALity:SOURce:VOLTage:W3V3a 1344
P3PHase<m>:QUALity:SOURce:VOLTage:W3VN3a 1344
P3PHase<m>:QUALity:SOURce:WIRing 1342
P3PHase<m>:QUALity:TOTal:POWer:DATA:HEADer? 1355
P3PHase<m>:QUALity:TOTal:POWer:DATA[:VALues]? 1356
P3PHase<m>:QUALity:TOTal:POWer:VERTical:OFFSet 1347
P3PHase<m>:QUALity:TOTal:POWer:VERTical:SCALe:MODE 1348
P3PHase<m>:QUALity:TOTal:POWer:VERTical:SCALe[:VALue] 1348
P3PHase<m>:TYPE 1314
P3PHase<m>[:ENABle] 1314
PBUS<pb>:BIT<n>:LABel 2018
PBUS<pb>:BIT<n>:SKEW 2019
PBUS<pb>:BIT<n>[:STATe] 2019
PBUS<pb>:CLEar 2020
PBUS<pb>:CLOCk 2020
PBUS<pb>:CLON 2020
PBUS<pb>:CLSLope 2020
PBUS<pb>:DATA:FORMat 1148
PBUS<pb>:DATA:HEADer? 1148
PBUS<pb>:DATA[:VALues]? 1148
PBUS<pb>:DECTable:COUNt? 2021
PBUS<pb>:DECTable:DATA? 2021
PBUS<pb>:DECTable:SHOW 2021
PBUS<pb>:DIGSignals:POSition 2022
PBUS<pb>:DIGSignals:SCALe 2022
PBUS<pb>:DISPlay:SHBU 2022
PBUS<pb>:DISPlay:SHDI 2022
PBUS<pb>:HYSTeresis<n> 2023
PBUS<pb>:POSition 2023
PBUS<pb>:SCALe 2024
PBUS<pb>:SKEW 2024

PBUS<pb>:STATe....2024
PBUS<pb>:TECHnology....2025
PBUS<pb>:THCoupling....2025
PBUS<pb>:THReshold<n>....2025
POWer<m>:ASKew[:EXECute]....1249
POWer<m>:EFFiciency:DISPlay:RESult:OUTPut<n>:EFFiciency[:ENABle]....1289
POWer<m>:EFFiciency:DISPlay:RESult:OUTPut<n>:POWer[:ENABle]....1289
POWer<m>:EFFiciency:DISPlay:RESult:TOTal<n>:EFFiciency[:ENABle]....1289
POWer<m>:EFFiciency:DISPlay:RESult:TOTal<n>:OPOWer[:ENABle]....1289
POWer<m>:EFFiciency:DISPlay:WAVeform:INPut<n>:POWer[:ENABle]....1304
POWer<m>:EFFiciency:DISPlay:WAVeform:OUTPut<n>:POWer[:ENABle]....1304
POWer<m>:EFFiciency:DISPlay:WAVeform:TOTal<n>:OPOWer[:ENABle]....1304
POWer<m>:EFFiciency:GATE....1289
POWer<m>:EFFiciency:INPut<n>:CURRent[:SOURce]....1286
POWer<m>:EFFiciency:INPut<n>:POWer:DATA:HEADer?....1312
POWer<m>:EFFiciency:INPut<n>:POWer:DATA[:VALues]?....1313
POWer<m>:EFFiciency:INPut<n>:POWer:VERTical:OFFSet....1304
POWer<m>:EFFiciency:INPut<n>:POWer:VERTical:SCALe:MODE....1305
POWer<m>:EFFiciency:INPut<n>:POWer:VERTical:SCALe[:VALue]....1305
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:HYSTeresis....1307
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:LLEVel....1307
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:MLEVel....1308
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:ABSolute:ULEVel....1308
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:LMODe....1309
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:HYSTeresis....1309
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:LOWer....1309
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MIDDle....1310
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:MODE....1310
POWer<m>:EFFiciency:INPut<n>:REFLevel<rl>:RELative:UPPer....1311
POWer<m>:EFFiciency:INPut<n>:VOLTage[:SOURce]....1287
POWer<m>:EFFiciency:INPut<n>[:TYPE]....1287
POWer<m>:EFFiciency:ONUMber?....1287
POWer<m>:EFFiciency:OUTPut<n>:CURRent[:SOURce]....1288
POWer<m>:EFFiciency:OUTPut<n>:POWer:DATA:HEADer?....1312
POWer<m>:EFFiciency:OUTPut<n>:POWer:DATA[:VALues]?....1313
POWer<m>:EFFiciency:OUTPut<n>:POWer:VERTical:OFFSet....1304
POWer<m>:EFFiciency:OUTPut<n>:POWer:VERTical:SCALe:MODE....1305
POWer<m>:EFFiciency:OUTPut<n>:POWer:VERTical:SCALe[:VALue]....1305
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:HYSTeresis....1307
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:LLEVel....1307
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:MLEVel....1308
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:ABSolute:ULEVel....1308
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:LMODe....1309
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:HYSTeresis....1309
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:LOWer....1309
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MIDDle....1310
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:MODE....1310
POWer<m>:EFFiciency:OUTPut<n>:REFLevel<rl>:RELative:UPPer....1311
POWer<m>:EFFiciency:OUTPut<n>:VOLTage[:SOURce]....1288
POWer<m>:EFFiciency:OUTPut<n>[:TYPE]....1288

POWer<m>:EFFiciency:RESult:INPut:POWer:AVERage? 1291
POWer<m>:EFFiciency:RESult:INPut:POWer:EVTCount? 1291
POWer<m>:EFFiciency:RESult:INPut:POWer:NPEak? 1291
POWer<m>:EFFiciency:RESult:INPut:POWer:PPEak? 1292
POWer<m>:EFFiciency:RESult:INPut:POWer:RMS? 1292
POWer<m>:EFFiciency:RESult:INPut:POWer:STDDev? 1292
POWer<m>:EFFiciency:RESult:INPut:POWer:WFMCount? 1293
POWer<m>:EFFiciency:RESult:INPut:POWer[:ACTual]? 1290
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:AVERage? 1291
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:EVTCount? 1291
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:NPEak? 1291
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:PPEak? 1292
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:RMS? 1292
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:STDDev? 1292
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency:WFMCount? 1293
POWer<m>:EFFiciency:RESult:TOTal:EFFiciency[:ACTual]? 1290
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:AVERage? 1291
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:EVTCount? 1291
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:NPEak? 1291
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:PPEak? 1292
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:RMS? 1292
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:STDDev? 1292
POWer<m>:EFFiciency:RESult:TOTal:OPOWer:WFMCount? 1293
POWer<m>:EFFiciency:RESult:TOTal:OPOWer[:ACTual]? 1290
POWer<m>:EFFiciency:STATistics:RESet 1250
POWer<m>:EFFiciency:STATistics:WFMCount? 1250
POWer<m>:EFFiciency:STATistics[:ENABle] 1249
POWer<m>:EFFiciency:TOTal<n>:OPOWer:DATA:HEADer? 1312
POWer<m>:EFFiciency:TOTal<n>:OPOWer:DATA[:VALues]? 1313
POWer<m>:EFFiciency:TOTal<n>:OPOWer:VERTical:OFFSet 1304
POWer<m>:EFFiciency:TOTal<n>:OPOWer:VERTical:SCALe:MODE 1305
POWer<m>:EFFiciency:TOTal<n>:OPOWer:VERTical:SCALe[:VALue] 1305
POWer<m>:HARMonics:AVAilable? 1268
POWer<m>:HARMonics:DISPlay:BARGraph:VERTical:SCALe:TYPE 1266
POWer<m>:HARMonics:DISPlay:HARMonics:STARt[:VALue] 1266
POWer<m>:HARMonics:DISPlay:HARMonics:STOP[:VALue] 1266
POWer<m>:HARMonics:DISPlay:HARMonics[:DISPlay] 1266
POWer<m>:HARMonics:DISPlay:POWer:WAVeform:ENABle 1304
POWer<m>:HARMonics:FREQuency:EN 1265
POWer<m>:HARMonics:FREQuency:MIL 1265
POWer<m>:HARMonics:FREQuency:RTCA 1265
POWer<m>:HARMonics:MEASurement:FREQuency[:ACTual]? 1269
POWer<m>:HARMonics:MEASurement:POWer:ACTive[:ACTual]? 1269
POWer<m>:HARMonics:MEASurement:THDFundament[:ACTual]? 1269
POWer<m>:HARMonics:MEASurement:THDRms[:ACTual]? 1269
POWer<m>:HARMonics:PFACtor:USER 1267
POWer<m>:HARMonics:PFACtor[:MODE] 1268
POWer<m>:HARMonics:POWer:DATA:HEADer? 1312
POWer<m>:HARMonics:POWer:DATA[:VALues]? 1313
POWer<m>:HARMonics:POWer:VERTical:OFFSet 1304

POWer<m>:HARMonics:POWer:VERTical:SCALe:MODE ..... 1305
POWer<m>:HARMonics:POWer:VERTical:SCALe[:VALue] ..... 1305
POWer<m>:HARMonics:REFLevel<rl>:ABSolute:HYSTeresis ..... 1307
POWer<m>:HARMonics:REFLevel<rl>:ABSolute:LLEVel ..... 1307
POWer<m>:HARMonics:REFLevel<rl>:ABSolute:MLEVel ..... 1308
POWer<m>:HARMonics:REFLevel<rl>:ABSolute:ULEVel ..... 1308
POWer<m>:HARMonics:REFLevel<rl>:LMODe ..... 1309
POWer<m>:HARMonics:REFLevel<rl>:RELative:HYSTeresis ..... 1309
POWer<m>:HARMonics:REFLevel<rl>:RELative:LOWer ..... 1310
POWer<m>:HARMonics:REFLevel<rl>:RELative:MIDDle ..... 1310
POWer<m>:HARMonics:REFLevel<rl>:RELative:MODE ..... 1311
POWer<m>:HARMonics:REFLevel<rl>:RELative:UPPer ..... 1311
POWer<m>:HARMonics:RESult<n>:MAGNitude:PERCent:MAXimum? ..... 1270
POWer<m>:HARMonics:RESult<n>:MAGNitude:PERCent:MINimum? ..... 1270
POWer<m>:HARMonics:RESult<n>:MAGNitude:PERCent[:ACTual]? ..... 1271
POWer<m>:HARMonics:RESult<n>:MAGNitude:RMS:MAXimum? ..... 1271
POWer<m>:HARMonics:RESult<n>:MAGNitude:RMS:MINimum? ..... 1271
POWer<m>:HARMonics:RESult<n>:MAGNitude:RMS[:ACTual]? ..... 1272
POWer<m>:HARMonics:RESult<n>:VIOLation? ..... 1270
POWer<m>:HARMonics:RESult<n>[:FREQuency]? ..... 1270
POWer<m>:HARMonics:REVision ..... 1264
POWer<m>:HARMonics:RPOWer:USER ..... 1267
POWer<m>:HARMonics:RPOWer[:MODE] ..... 1267
POWer<m>:HARMonics:SOURce:CURRent ..... 1263
POWer<m>:HARMonics:SOURce[:VOLTage] ..... 1264
POWer<m>:HARMonics:STANdard ..... 1264
POWer<m>:ONOFf:INPut:AC:ABSolute[:VALue] ..... 1283
POWer<m>:ONOFf:INPut:DC:ABSolute[:VALue] ..... 1283
POWer<m>:ONOFf:INPut:TYPE ..... 1283
POWer<m>:ONOFf:INPut[:SOURce] ..... 1284
POWer<m>:ONOFf:ONUMber ..... 1284
POWer<m>:ONOFf:OUTPut<n>:DC:ABSolute[:VALue] ..... 1284
POWer<m>:ONOFf:OUTPut<n>:DISPlay:RESult[:ENABle] ..... 1284
POWer<m>:ONOFf:OUTPut<n>:DISPlay:RLINe[:ENABle] ..... 1285
POWer<m>:ONOFf:OUTPut<n>[:SOURce] ..... 1285
POWer<m>:ONOFf:RESult<n>:TIME[:ACTual]? ..... 1286
POWer<m>:ONOFf[:TYPE] ..... 1285
POWer<m>:QUALity:DISPlay:CURRent:CREStfactor ..... 1252
POWer<m>:QUALity:DISPlay:CURRent:PEAK ..... 1253
POWer<m>:QUALity:DISPlay:CURRent:RMS ..... 1253
POWer<m>:QUALity:DISPlay:POWer:ACTive ..... 1253
POWer<m>:QUALity:DISPlay:POWer:APParent ..... 1253
POWer<m>:QUALity:DISPlay:POWer:PFACtor ..... 1254
POWer<m>:QUALity:DISPlay:POWer:PHASe ..... 1254
POWer<m>:QUALity:DISPlay:POWer:REACtive ..... 1254
POWer<m>:QUALity:DISPlay:POWer:WAVeform:ENABle ..... 1304
POWer<m>:QUALity:DISPlay:VOLTage:CREStfactor ..... 1254
POWer<m>:QUALity:DISPlay:VOLTage:FREQuency ..... 1255
POWer<m>:QUALity:DISPlay:VOLTage:PEAK ..... 1255
POWer<m>:QUALity:DISPlay:VOLTage:RMS ..... 1255

POWer<m>:QUALity:FREQuency:USER....1252
POWer<m>:QUALity:FREQuency[:VALue]....1251
POWer<m>:QUALity:GATE....1252
POWer<m>:QUALity:POWer:DATA:HEADer?....1312
POWer<m>:QUALity:POWer:DATA[:VALues]?....1313
POWer<m>:QUALity:POWer:VERTical:OFFSet....1304
POWer<m>:QUALity:POWer:VERTical:SCALe:MODE....1305
POWer<m>:QUALity:POWer:VERTical:SCALe[:VALue]....1305
POWer<m>:QUALity:REFLevel<rl>:ABSolute:HYSTeresis....1307
POWer<m>:QUALity:REFLevel<rl>:ABSolute:LLEVel....1307
POWer<m>:QUALity:REFLevel<rl>:ABSolute:MLEVel....1308
POWer<m>:QUALity:REFLevel<rl>:ABSolute:ULEVel....1308
POWer<m>:QUALity:REFLevel<rl>:LMODe....1309
POWer<m>:QUALity:REFLevel<rl>:RELative:HYSTeresis....1309
POWer<m>:QUALity:REFLevel<rl>:RELative:LOWer....1310
POWer<m>:QUALity:REFLevel<rl>:RELative:MIDDle....1310
POWer<m>:QUALity:REFLevel<rl>:RELative:MODE....1311
POWer<m>:QUALity:REFLevel<rl>:RELative:UPPer....1311
POWer<m>:QUALity:RESult:CURRent:CRESt factor:AVERage?....1257
POWer<m>:QUALity:RESult:CURRent:CRESt factor:EVTCount?....1257
POWer<m>:QUALity:RESult:CURRent:CRESt factor:NPEak?....1257
POWer<m>:QUALity:RESult:CURRent:CRESt factor:PPEak?....1257
POWer<m>:QUALity:RESult:CURRent:CRESt factor:RMS?....1257
POWer<m>:QUALity:RESult:CURRent:CRESt factor:STDDev?....1257
POWer<m>:QUALity:RESult:CURRent:CRESt factor:WFMCount?....1257
POWer<m>:QUALity:RESult:CURRent:CRESt factor[:ACTual]?....1257
POWer<m>:QUALity:RESult:CURRent:PEAK:AVERage?....1258
POWer<m>:QUALity:RESult:CURRent:PEAK:EVTCount?....1258
POWer<m>:QUALity:RESult:CURRent:PEAK:NPEak?....1258
POWer<m>:QUALity:RESult:CURRent:PEAK:PPEak?....1258
POWer<m>:QUALity:RESult:CURRent:PEAK:RMS?....1258
POWer<m>:QUALity:RESult:CURRent:PEAK:STDDev?....1258
POWer<m>:QUALity:RESult:CURRent:PEAK:WFMCount?....1258
POWer<m>:QUALity:RESult:CURRent:PEAK[:ACTual]?....1258
POWer<m>:QUALity:RESult:CURRent:RMS:AVERage?....1258
POWer<m>:QUALity:RESult:CURRent:RMS:EVTCount?....1258
POWer<m>:QUALity:RESult:CURRent:RMS:NPEak?....1258
POWer<m>:QUALity:RESult:CURRent:RMS:PPEak?....1258
POWer<m>:QUALity:RESult:CURRent:RMS:RMS?....1258
POWer<m>:QUALity:RESult:CURRent:RMS:STDDev?....1258
POWer<m>:QUALity:RESult:CURRent:RMS:WFMCount?....1258
POWer<m>:QUALity:RESult:CURRent:RMS[:ACTual]?....1258
POWer<m>:QUALity:RESult:POWer:ACTive:AVERage?....1259
POWer<m>:QUALity:RESult:POWer:ACTive:EVTCount?....1259
POWer<m>:QUALity:RESult:POWer:ACTive:NPEak?....1259
POWer<m>:QUALity:RESult:POWer:ACTive:PPEak?....1259
POWer<m>:QUALity:RESult:POWer:ACTive:RMS?....1259
POWer<m>:QUALity:RESult:POWer:ACTive:STDDev?....1259
POWer<m>:QUALity:RESult:POWer:ACTive:WFMCount?....1259
POWer<m>:QUALity:RESult:POWer:ACTive[:ACTual]?....1259

POWer<m>:QUALity:RESult:POWer:APParent:AVERage?....1259
POWer<m>:QUALity:RESult:POWer:APParent:EVTCount?....1259
POWer<m>:QUALity:RESult:POWer:APParent:NPEak?....1259
POWer<m>:QUALity:RESult:POWer:APParent:PPEak?....1259
POWer<m>:QUALity:RESult:POWer:APParent:RMS?....1259
POWer<m>:QUALity:RESult:POWer:APParent:STDDev?....1259
POWer<m>:QUALity:RESult:POWer:APParent:WFMCount?....1259
POWer<m>:QUALity:RESult:POWer:APParent[:ACTual]?....1259
POWer<m>:QUALity:RESult:POWer:PFACtor:AVERage?....1260
POWer<m>:QUALity:RESult:POWer:PFACtor:EVTCount?....1260
POWer<m>:QUALity:RESult:POWer:PFACtor:NPEak?....1260
POWer<m>:QUALity:RESult:POWer:PFACtor:PPEak?....1260
POWer<m>:QUALity:RESult:POWer:PFACtor:RMS?....1260
POWer<m>:QUALity:RESult:POWer:PFACtor:STDDev?....1260
POWer<m>:QUALity:RESult:POWer:PFACtor:WFMCount?....1260
POWer<m>:QUALity:RESult:POWer:PFACtor[:ACTual]?....1260
POWer<m>:QUALity:RESult:POWer:PHASe:AVERage?....1260
POWer<m>:QUALity:RESult:POWer:PHASe:EVTCount?....1260
POWer<m>:QUALity:RESult:POWer:PHASe:NPEak?....1260
POWer<m>:QUALity:RESult:POWer:PHASe:PPEak?....1260
POWer<m>:QUALity:RESult:POWer:PHASe:RMS?....1260
POWer<m>:QUALity:RESult:POWer:PHASe:STDDev?....1260
POWer<m>:QUALity:RESult:POWer:PHASe:WFMCount?....1260
POWer<m>:QUALity:RESult:POWer:PHASe[:ACTual]?....1260
POWer<m>:QUALity:RESult:POWer:REACtive:AVERage?....1261
POWer<m>:QUALity:RESult:POWer:REACtive:EVTCount?....1261
POWer<m>:QUALity:RESult:POWer:REACtive:NPEak?....1261
POWer<m>:QUALity:RESult:POWer:REACtive:PPEak?....1261
POWer<m>:QUALity:RESult:POWer:REACtive:RMS?....1261
POWer<m>:QUALity:RESult:POWer:REACtive:STDDev?....1261
POWer<m>:QUALity:RESult:POWer:REACtive:WFMCount?....1261
POWer<m>:QUALity:RESult:POWer:REACtive[:ACTual]?....1261
POWer<m>:QUALity:RESult:VOLTage:CREStfactor:AVERage?....1261
POWer<m>:QUALity:RESult:VOLTage:CREStfactor:EVTCount?....1261
POWer<m>:QUALity:RESult:VOLTage:CREStfactor:NPEak?....1261
POWer<m>:QUALity:RESult:VOLTage:CREStfactor:PPEak?....1261
POWer<m>:QUALity:RESult:VOLTage:CREStfactor:RMS?....1261
POWer<m>:QUALity:RESult:VOLTage:CREStfactor:STDDev?....1261
POWer<m>:QUALity:RESult:VOLTage:CREStfactor:WFMCount?....1261
POWer<m>:QUALity:RESult:VOLTage:CREStfactor[:ACTual]?....1261
POWer<m>:QUALity:RESult:VOLTage:FREQuency:AVERage?....1261
POWer<m>:QUALity:RESult:VOLTage:FREQuency:EVTCount?....1261
POWer<m>:QUALity:RESult:VOLTage:FREQuency:NPEak?....1261
POWer<m>:QUALity:RESult:VOLTage:FREQuency:PPEak?....1262
POWer<m>:QUALity:RESult:VOLTage:FREQuency:RMS?....1262
POWer<m>:QUALity:RESult:VOLTage:FREQuency:STDDev?....1262
POWer<m>:QUALity:RESult:VOLTage:FREQuency:WFMCount?....1262
POWer<m>:QUALity:RESult:VOLTage:FREQuency[:ACTual]?....1261
POWer<m>:QUALity:RESult:VOLTage:PEAK:AVERage?....1262
POWer<m>:QUALity:RESult:VOLTage:PEAK:EVTCount?....1262

POWer<m>:QUALity:RESult:VOLTage:PEAK:NPEak? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK:PPEak? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK:RMS? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK:STDDev? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK:WFMCount? 1262
POWer<m>:QUALity:RESult:VOLTage:PEAK[:ACTual]? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS:AVERage? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS:EVTCount? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS:NPEak? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS:PPEak? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS:RMS? 1262
POWer<m>:QUALity:RESult:VOLTage:RMS:STDDev? 1263
POWer<m>:QUALity:RESult:VOLTage:RMS:WFMCount? 1263
POWer<m>:QUALity:RESult:VOLTage:RMS[:ACTual]? 1262
POWer<m>:QUALity:SOURce:CURRent 1251
POWer<m>:QUALity:SOURce:CYCLe 1251
POWer<m>:QUALity:SOURce[:VOLTage] 1251
POWer<m>:QUALity:STATistics:RESet 1250
POWer<m>:QUALity:STATistics:WFMCount? 1250
POWer<m>:QUALity:STATistics[:ENABle] 1249
POWer<m>:SOA:LIMit:APPLy 1295
POWer<m>:SOA:LIMit:IMAX 1295
POWer<m>:SOA:LIMit:PMAX 1295
POWer<m>:SOA:LIMit:VMAX 1296
POWer<m>:SOA:MTESt<n>:IMEXport:NAME 1293
POWer<m>:SOA:MTESt<n>:IMEXport:OPEN 1293
POWer<m>:SOA:MTESt<n>:IMEXport:SAVE 1294
POWer<m>:SOA:MTESt<n>:ONViolation:BEEP 1300
POWer<m>:SOA:MTESt<n>:ONViolation:SCReenshot 1300
POWer<m>:SOA:MTESt<n>:ONViolation:STOP 1300
POWer<m>:SOA:MTESt<n>:ONViolation:TRIGgerout 1301
POWer<m>:SOA:MTESt<n>:ONViolation:WFMSave 1301
POWer<m>:SOA:MTESt<n>:SEGMent<o>:ADD 1296
POWer<m>:SOA:MTESt<n>:SEGMent<o>:COUNt? 1296
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:ADD 1297
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:COUNt? 1297
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:REMove 1298
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:VALid? 1298
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:X 1298
POWer<m>:SOA:MTESt<n>:SEGMent<o>:POINt<p>:Y 1299
POWer<m>:SOA:MTESt<n>:SEGMent<o>:REMove 1299
POWer<m>:SOA:RESult<n>:COUNt:FWAVeforms? 1302
POWer<m>:SOA:RESult<n>:COUNt:PWAVeforms? 1302
POWer<m>:SOA:RESult<n>:COUNt:WAVeforms? 1302
POWer<m>:SOA:RESult<n>:FRATe? 1303
POWer<m>:SOA:RESult<n>[:RESult]? 1303
POWer<m>:SOA:SOURce:CURRent 1294
POWer<m>:SOA:SOURce:VOLTage 1294
POWer<m>:SWITching:DISPlay:CONDuction:COLor 1276
POWer<m>:SWITching:DISPlay:CONDuction[:ENABle] 1275

POWer<m>:SWITching:DISPlay:LABel[:ENABle] 1276
POWer<m>:SWITching:DISPlay:NCONduction:COLor 1276
POWer<m>:SWITching:DISPlay:NCONduction[:ENABle] 1275
POWer<m>:SWITching:DISPlay:POWer:WAVeform[:ENABle] 1304
POWer<m>:SWITching:DISPlay:TOFF:COLor 1276
POWer<m>:SWITching:DISPlay:TOFF[:ENABle] 1275
POWer<m>:SWITching:DISPlay:TON:COLor 1276
POWer<m>:SWITching:DISPlay:TON[:ENABle] 1275
POWer<m>:SWITching:DISPlay:TOTal[:ENABle] 1276
POWer<m>:SWITching:DISPlay:TYPE 1273
POWer<m>:SWITching:POWer:DATA:HEADer? 1312
POWer<m>:SWITching:POWer:DATA[:VALues]? 1313
POWer<m>:SWITching:POWer:VERTical:OFFSet 1304
POWer<m>:SWITching:POWer:VERTical:SCALe:MODE 1305
POWer<m>:SWITching:POWer:VERTical:SCALe[:VALue] 1305
POWer<m>:SWITching:REGion:CONDuction:STOP 1274
POWer<m>:SWITching:REGion:CONDuction[:STARt] 1273
POWer<m>:SWITching:REGion:NCONduction:STOP 1275
POWer<m>:SWITching:REGion:NCONduction[:STARt] 1274
POWer<m>:SWITching:REGion:TOFF:STOP 1274
POWer<m>:SWITching:REGion:TOFF[:STARt] 1274
POWer<m>:SWITching:REGion:TON:STOP 1273
POWer<m>:SWITching:REGion:TON[:STARt] 1273
POWer<m>:SWITching:REGion:TOTal:STOP 1275
POWer<m>:SWITching:REGion:TOTal[:STARt] 1273
POWer<m>:SWITching:RESult:CONDuction:ENERgy:AVERage? 1279
POWer<m>:SWITching:RESult:CONDuction:ENERgy:EVTCount? 1279
POWer<m>:SWITching:RESult:CONDuction:ENERgy:NPEak? 1280
POWer<m>:SWITching:RESult:CONDuction:ENERgy:PPEak? 1280
POWer<m>:SWITching:RESult:CONDuction:ENERgy:RMS? 1281
POWer<m>:SWITching:RESult:CONDuction:ENERgy:STDDev? 1281
POWer<m>:SWITching:RESult:CONDuction:ENERgy:WFMCount? 1282
POWer<m>:SWITching:RESult:CONDuction:ENERgy[:ACTual]? 1278
POWer<m>:SWITching:RESult:CONDuction:POWer:AVERage? 1279
POWer<m>:SWITching:RESult:CONDuction:POWer:EVTCount? 1279
POWer<m>:SWITching:RESult:CONDuction:POWer:NPEak? 1280
POWer<m>:SWITching:RESult:CONDuction:POWer:PPEak? 1280
POWer<m>:SWITching:RESult:CONDuction:POWer:RMS? 1281
POWer<m>:SWITching:RESult:CONDuction:POWer:STDDev? 1281
POWer<m>:SWITching:RESult:CONDuction:POWer:WFMCount? 1282
POWer<m>:SWITching:RESult:CONDuction:POWer[:ACTual]? 1278
POWer<m>:SWITching:RESult:NCONduction:ENERgy:AVERage? 1279
POWer<m>:SWITching:RESult:NCONduction:ENERgy:EVTCount? 1279
POWer<m>:SWITching:RESult:NCONduction:ENERgy:NPEak? 1280
POWer<m>:SWITching:RESult:NCONduction:ENERgy:PPEak? 1280
POWer<m>:SWITching:RESult:NCONduction:ENERgy:RMS? 1281
POWer<m>:SWITching:RESult:NCONduction:ENERgy:STDDev? 1281
POWer<m>:SWITching:RESult:NCONduction:ENERgy:WFMCount? 1282
POWer<m>:SWITching:RESult:NCONduction:ENERgy[:ACTual]? 1278
POWer<m>:SWITching:RESult:NCONduction:POWer:AVERage? 1279

POWer<m>:SWITching:RESult:NCONduction:POWer:EVTCount? 1279
POWer<m>:SWITching:RESult:NCONduction:POWer:NPEak? 1280
POWer<m>:SWITching:RESult:NCONduction:POWer:PPEak? 1280
POWer<m>:SWITching:RESult:NCONduction:POWer:RMS? 1281
POWer<m>:SWITching:RESult:NCONduction:POWer:STDDev? 1281
POWer<m>:SWITching:RESult:NCONduction:POWer:WFMCount? 1282
POWer<m>:SWITching:RESult:NCONduction:POWer[:ACTual]? 1278
POWer<m>:SWITching:RESult:TOFF:ENERgy:AVERage? 1279
POWer<m>:SWITching:RESult:TOFF:ENERgy:EVTCount? 1279
POWer<m>:SWITching:RESult:TOFF:ENERgy:NPEak? 1280
POWer<m>:SWITching:RESult:TOFF:ENERgy:PPEak? 1280
POWer<m>:SWITching:RESult:TOFF:ENERgy:RMS? 1281
POWer<m>:SWITching:RESult:TOFF:ENERgy:STDDev? 1281
POWer<m>:SWITching:RESult:TOFF:ENERgy:WFMCount? 1282
POWer<m>:SWITching:RESult:TOFF:ENERgy[:ACTual]? 1278
POWer<m>:SWITching:RESult:TOFF:POWer:AVERage? 1279
POWer<m>:SWITching:RESult:TOFF:POWer:EVTCount? 1279
POWer<m>:SWITching:RESult:TOFF:POWer:NPEak? 1280
POWer<m>:SWITching:RESult:TOFF:POWer:PPEak? 1280
POWer<m>:SWITching:RESult:TOFF:POWer:RMS? 1281
POWer<m>:SWITching:RESult:TOFF:POWer:STDDev? 1281
POWer<m>:SWITching:RESult:TOFF:POWer:WFMCount? 1282
POWer<m>:SWITching:RESult:TOFF:POWer[:ACTual]? 1279
POWer<m>:SWITching:RESult:TON:ENERgy:AVERage? 1279
POWer<m>:SWITching:RESult:TON:ENERgy:EVTCount? 1279
POWer<m>:SWITching:RESult:TON:ENERgy:NPEak? 1280
POWer<m>:SWITching:RESult:TON:ENERgy:PPEak? 1280
POWer<m>:SWITching:RESult:TON:ENERgy:RMS? 1281
POWer<m>:SWITching:RESult:TON:ENERgy:STDDev? 1281
POWer<m>:SWITching:RESult:TON:ENERgy:WFMCount? 1282
POWer<m>:SWITching:RESult:TON:ENERgy[:ACTual]? 1279
POWer<m>:SWITching:RESult:TON:POWer:AVERage? 1279
POWer<m>:SWITching:RESult:TON:POWer:EVTCount? 1279
POWer<m>:SWITching:RESult:TON:POWer:NPEak? 1280
POWer<m>:SWITching:RESult:TON:POWer:PPEak? 1280
POWer<m>:SWITching:RESult:TON:POWer:RMS? 1281
POWer<m>:SWITching:RESult:TON:POWer:STDDev? 1281
POWer<m>:SWITching:RESult:TON:POWer:WFMCount? 1282
POWer<m>:SWITching:RESult:TON:POWer[:ACTual]? 1279
POWer<m>:SWITching:RESult:TOTal:ENERgy:AVERage? 1279
POWer<m>:SWITching:RESult:TOTal:ENERgy:EVTCount? 1280
POWer<m>:SWITching:RESult:TOTal:ENERgy:NPEak? 1280
POWer<m>:SWITching:RESult:TOTal:ENERgy:PPEak? 1281
POWer<m>:SWITching:RESult:TOTal:ENERgy:RMS? 1281
POWer<m>:SWITching:RESult:TOTal:ENERgy:STDDev? 1282
POWer<m>:SWITching:RESult:TOTal:ENERgy:WFMCount? 1282
POWer<m>:SWITching:RESult:TOTal:ENERgy[:ACTual]? 1279
POWer<m>:SWITching:RESult:TOTal:POWer:AVERage? 1279
POWer<m>:SWITching:RESult:TOTal:POWer:EVTCount? 1280
POWer<m>:SWITching:RESult:TOTal:POWer:NPEak? 1280

POWer<m>:SWITching:RESult:TOTal:POWer:PPEak? 1281
POWer<m>:SWITching:RESult:TOTal:POWer:RMS? 1281
POWer<m>:SWITching:RESult:TOTal:POWer:STDDev? 1282
POWer<m>:SWITching:RESult:TOTal:POWer:WFMCount? 1282
POWer<m>:SWITching:RESult:TOTal:POWer[:ACTual]? 1279
POWer<m>:SWITching:SOURce:CURRent 1272
POWer<m>:SWITching:SOURce[:VOLTage] 1273
POWer<m>:SWITching:STATistics:ENABle 1249
POWer<m>:SWITching:STATistics:RESet 1250
POWer<m>:SWITching:STATistics:WFMCount? 1250
POWer<m>:TYPE 1248
POWer<m>[:ENABle] 1248
PROBe<ch>:ID:PARTnumber? 1025
PROBe<ch>:ID:PRDate? 1025
PROBe<ch>:ID:SRNumber? 1025
PROBe<ch>:ID:SWVersion? 1025
PROBe<ch>:PMETer:RESults:COMMon? 1013
PROBe<ch>:PMETer:RESults:DIFFerential? 1013
PROBe<ch>:PMETer:RESults:NEGative? 1013
PROBe<ch>:PMETer:RESults:POSitive? 1014
PROBe<ch>:PMETer:RESults:SINGle? 1012
PROBe<ch>:PMETer:STATe 1012
PROBe<ch>:SETup:ACCoupling 1011
PROBe<ch>:SETup:ADAPter 1024
PROBe<ch>:SETup:ADVanced:AUDioverload 1014
PROBe<ch>:SETup:ADVanced:FILTer 1014
PROBe<ch>:SETup:ADVanced:PMToffset 1015
PROBe<ch>:SETup:ADVanced:PMToffset 1021
PROBe<ch>:SETup:ADVanced:RANGe 1015
PROBe<ch>:SETup:ADVanced:RDEFaults 1023
PROBe<ch>:SETup:ADVanced:UNIT 1021
PROBe<ch>:SETup:ALIGnment:GAIN:EXECute 1022
PROBe<ch>:SETup:ALIGnment:WRITe 1022
PROBe<ch>:SETup:ALIGnment:ZERO:EXECute 1022
PROBe<ch>:SETup:ATTenuation:DEFProbe 1006
PROBe<ch>:SETup:ATTenuation:MANual 1005
PROBe<ch>:SETup:ATTenuation:MODE 1005
PROBe<ch>:SETup:ATTenuation:TDEFprobe 1024
PROBe<ch>:SETup:ATTenuation:UNIT 1007
PROBe<ch>:SETup:ATTenuation[:AUTO]? 1005
PROBe<ch>:SETup:BANDwidth? 1007
PROBe<ch>:SETup:CAPacitance? 1026
PROBe<ch>:SETup:CMOFfset 1011
PROBe<ch>:SETup:CMOFfset 1016
PROBe<ch>:SETup:DCRange:MAX? 1026
PROBe<ch>:SETup:DCRange:MIN? 1026
PROBe<ch>:SETup:DEGauss 1020
PROBe<ch>:SETup:DISPlaydiff 1011
PROBe<ch>:SETup:DMOFfset 1017
PROBe<ch>:SETup:EPATtenuator 1012

PROBe<ch>:SETup:GAIN:AUTO? 1019
PROBe<ch>:SETup:GAIN:MANual 1019
PROBe<ch>:SETup:IMPedance? 1026
PROBe<ch>:SETup:LASer:STATe? 1021
PROBe<ch>:SETup:MODE 1010
PROBe<ch>:SETup:NAME? 1008
PROBe<ch>:SETup:NOFFset 1017
PROBe<ch>:SETup:OFFSet:AZERo 1008
PROBe<ch>:SETup:OFFSet:STPRobe 1020
PROBe<ch>:SETup:OFFSet:TOMean 1007
PROBe<ch>:SETup:OFFSet:USEautozero 1008
PROBe<ch>:SETup:OFFSet:ZADJust 1020
PROBe<ch>:SETup:POFFset 1017
PROBe<ch>:SETup:PRMode 1016
PROBe<ch>:SETup:STATe? 1009
PROBe<ch>:SETup:TERM:ADJust 1018
PROBe<ch>:SETup:TERM:MEASure? 1018
PROBe<ch>:SETup:TERM:MODE 1018
PROBe<ch>:SETup:TERM:STATe 1019
PROBe<ch>:SETup:TIPModel:INFO? 1023
PROBe<ch>:SETup:TIPModel:NAME? 1023
PROBe<ch>:SETup:TYPE? 1009
REFCurve<rc>:ABORt 1106
REFCurve<rc>:AXIS[:XMODe] 1109
REFCurve<rc>:CLEar 1106
REFCurve<rc>:DATA:HEADer? 1143
REFCurve<rc>:DATA:STYPe? 1144
REFCurve<rc>:DATA[:VALues]? 1144
REFCurve<rc>:HMODe 1110
REFCurve<rc>:NAME 1106
REFCurve<rc>:OFFSet 1107
REFCurve<rc>:OPEN 1107
REFCurve<rc>:POSition 1110
REFCurve<rc>:RESCale:HORizontal:FACTor 1111
REFCurve<rc>:RESCale:HORizontal:OFFSet 1111
REFCurve<rc>:RESCale:HORizontal:STATe 1111
REFCurve<rc>:RESCale:VERTical:FACTor 1112
REFCurve<rc>:RESCale:VERTical:OFFSet 1112
REFCurve<rc>:RESCale:VERTical:STATe 1112
REFCurve<rc>:RESTore 1107
REFCurve<rc>:SAVE 1107
REFCurve<rc>:SCALe 1110
REFCurve<rc>:SOURce 1108
REFCurve<rc>:STATe 1109
REFCurve<rc>:TOORiginal 1113
REFCurve<rc>:UPDate 1109
REFCurve<rc>:VMODe 1113
REFLevel<rl>:ABSolute:HYSTeresis 1074
REFLevel<rl>:ABSolute:LLEVel 1074
REFLevel<rl>:ABSolute:MLEVel 1075

REFLevel<rl>:ABSolute:ULEVel....1075
REFLevel<rl>:AUTO:MODE....1073
REFLevel<rl>:ENABle....1073
REFLevel<rl>:LMODe....1073
REFLevel<rl>:RELative:HYSTeresis....1075
REFLevel<rl>:RELative:LOWer....1076
REFLevel<rl>:RELative:MIDDle....1076
REFLevel<rl>:RELative:MODE....1076
REFLevel<rl>:RELative:UPPer....1077
RUN....989
RUNCont....989
RUNSingle....990
SAVeset:CONFig:PREView....1126
SAVeset:CONFig:UPReferences....1126
SBUS<sb>:ARINc:BRMode....1772
SBUS<sb>:ARINc:BRValue....1772
SBUS<sb>:ARINc:FILTer:BIT....1778
SBUS<sb>:ARINc:FILTer:CHKall....1777
SBUS<sb>:ARINc:FILTer:CLR....1777
SBUS<sb>:ARINc:FILTer:DMAX....1779
SBUS<sb>:ARINc:FILTer:DMIN....1779
SBUS<sb>:ARINc:FILTer:DOPerator....1780
SBUS<sb>:ARINc:FILTer:ERENable....1779
SBUS<sb>:ARINc:FILTer:ERRor<n>:ENABle....1779
SBUS<sb>:ARINc:FILTer:FIENable....1780
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:ENABle....1780
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:BIT....1778
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1779
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1779
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1780
SBUS<sb>:ARINc:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1780
SBUS<sb>:ARINc:FILTer:FRENable....1780
SBUS<sb>:ARINc:FILTer:INVert....1778
SBUS<sb>:ARINc:FILTer:RST....1778
SBUS<sb>:ARINc:FORMat:ADDFrame....1782
SBUS<sb>:ARINc:FORMat:CLR....1783
SBUS<sb>:ARINc:FORMat:FCOunt?....1783
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ADDField....1783
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:COLor....1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1788
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1786
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1787
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:COLor....1785
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1789
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1788
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1787
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1786
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:NAME....1785
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:PARA....1790
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:PARB....1790

SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1786
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1789
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:FLDCount?....1783
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID1Field....1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID1Value....1785
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID2Field....1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID2Value....1785
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID3Field....1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:ID3Value....1785
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:NAME....1784
SBUS<sb>:ARINc:FORMat:FRAMe<fr>:TYPE....1784
SBUS<sb>:ARINc:FORMat:LOAD....1782
SBUS<sb>:ARINc:FORMat:SAVE....1782
SBUS<sb>:ARINc:FORMat:SELect....1781
SBUS<sb>:ARINc:MAXGap:BITS....1775
SBUS<sb>:ARINc:MAXGap:SELect....1775
SBUS<sb>:ARINc:MINGap:BITS....1775
SBUS<sb>:ARINc:MINGap:SELect....1776
SBUS<sb>:ARINc:NEWList....1774
SBUS<sb>:ARINc:POLarity....1773
SBUS<sb>:ARINc:POSition....1776
SBUS<sb>:ARINc:SCALe....1776
SBUS<sb>:ARINc:SOURce....1773
SBUS<sb>:ARINc:SWTindex?....1794
SBUS<sb>:ARINc:SWTTime?....1794
SBUS<sb>:ARINc:SYMBols....1774
SBUS<sb>:ARINc:THReshold:HIGH....1773
SBUS<sb>:ARINc:THReshold:HYSTeresis....1774
SBUS<sb>:ARINc:THReshold:LOW....1774
SBUS<sb>:ARINc:WCOunt?....1795
SBUS<sb>:ARINc:WORD<w>:BITRate?....1795
SBUS<sb>:ARINc:WORD<w>:DATA?....1797
SBUS<sb>:ARINc:WORD<w>:FLD<fl>:FVALue?....1797
SBUS<sb>:ARINc:WORD<w>:FLD<fl>:LABel?....1796
SBUS<sb>:ARINc:WORD<w>:FLD<fl>:STATe?....1796
SBUS<sb>:ARINc:WORD<w>:FLD<fl>:VALue?....1797
SBUS<sb>:ARINc:WORD<w>:FLDCount?....1795
SBUS<sb>:ARINc:WORD<w>:LABel?....1798
SBUS<sb>:ARINc:WORD<w>:PATTern?....1798
SBUS<sb>:ARINc:WORD<w>:SDI?....1798
SBUS<sb>:ARINc:WORD<w>:SSM?....1799
SBUS<sb>:ARINc:WORD<w>:STARt?....1799
SBUS<sb>:ARINc:WORD<w>:STATe?....1799
SBUS<sb>:ARINc:WORD<w>:STOP?....1800
SBUS<sb>:ARINc:WORD<w>:SYMBol?....1800
SBUS<sb>:CAN:BITRate....1656
SBUS<sb>:CAN:FAST:HYSTeresis....1658
SBUS<sb>:CAN:FAST:THReshold....1658
SBUS<sb>:CAN:FCOunt?....1691
SBUS<sb>:CAN:FDATa:DBITrate....1657

SBUS<sb>:CAN:FDATa:SAMPlepoint....1657
SBUS<sb>:CAN:FILTer:BIT....1662
SBUS<sb>:CAN:FILTer:CHKall....1661
SBUS<sb>:CAN:FILTer:CLR....1661
SBUS<sb>:CAN:FILTer:DMAX....1662
SBUS<sb>:CAN:FILTer:DMIN....1663
SBUS<sb>:CAN:FILTer:DOPerator....1663
SBUS<sb>:CAN:FILTer:ERENable....1664
SBUS<sb>:CAN:FILTer:ERRor<n>:ENABle....1664
SBUS<sb>:CAN:FILTer:FIENable....1664
SBUS<sb>:CAN:FILTer:FRAMe<fr>:ENABle....1664
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:BIT....1662
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1662
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1663
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1663
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1664
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1665
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1665
SBUS<sb>:CAN:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1665
SBUS<sb>:CAN:FILTer:FRENable....1664
SBUS<sb>:CAN:FILTer:IMAX....1665
SBUS<sb>:CAN:FILTer:IMIN....1665
SBUS<sb>:CAN:FILTer:INVert....1662
SBUS<sb>:CAN:FILTer:IOPerator....1665
SBUS<sb>:CAN:FILTer:RST....1662
SBUS<sb>:CAN:FORMat:ADDFrame....1667
SBUS<sb>:CAN:FORMat:CLR....1668
SBUS<sb>:CAN:FORMat:FCOunt?....1668
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ADDField....1668
SBUS<sb>:CAN:FORMat:FRAMe<fr>:COLor....1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1673
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1671
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1672
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:COLor....1670
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1674
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1673
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1672
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1671
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:NAME....1670
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:PARA....1675
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:PARB....1675
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1671
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1674
SBUS<sb>:CAN:FORMat:FRAMe<fr>:FLDCount?....1668
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID1Field....1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID1Value....1670
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID2Field....1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID2Value....1670
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID3Field....1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:ID3Value....1670

SBUS<sb>:CAN:FORMat:FRAMe<fr>:NAME ... 1669
SBUS<sb>:CAN:FORMat:FRAMe<fr>:TYPE ... 1669
SBUS<sb>:CAN:FORMat:LOAD ... 1667
SBUS<sb>:CAN:FORMat:SAVE ... 1667
SBUS<sb>:CAN:FORMat:SELect ... 1666
SBUS<sb>:CAN:FRAMe<fr>:ACKState? ... 1692
SBUS<sb>:CAN:FRAMe<fr>:ACKValue? ... 1692
SBUS<sb>:CAN:FRAMe<fr>:CSSTate? ... 1694
SBUS<sb>:CAN:FRAMe<fr>:CSValue? ... 1694
SBUS<sb>:CAN:FRAMe<fr>:DATA? ... 1695
SBUS<sb>:CAN:FRAMe<fr>:DBITrate? ... 1702
SBUS<sb>:CAN:FRAMe<fr>:DLCState? ... 1695
SBUS<sb>:CAN:FRAMe<fr>:DLCValue? ... 1695
SBUS<sb>:CAN:FRAMe<fr>:FDATa<o>:BRS? ... 1702
SBUS<sb>:CAN:FRAMe<fr>:FDATa<o>:ESI? ... 1701
SBUS<sb>:CAN:FRAMe<fr>:FLD<fl>:FVALue? ... 1694
SBUS<sb>:CAN:FRAMe<fr>:FLD<fl>:LABel? ... 1693
SBUS<sb>:CAN:FRAMe<fr>:FLD<fl>:STATe? ... 1692
SBUS<sb>:CAN:FRAMe<fr>:FLD<fl>:VALue? ... 1693
SBUS<sb>:CAN:FRAMe<fr>:FLDCount? ... 1691
SBUS<sb>:CAN:FRAMe<fr>:IDSTate? ... 1696
SBUS<sb>:CAN:FRAMe<fr>:IDTYpe? ... 1696
SBUS<sb>:CAN:FRAMe<fr>:IDValue? ... 1696
SBUS<sb>:CAN:FRAMe<fr>:NBITrate? ... 1702
SBUS<sb>:CAN:FRAMe<fr>:NDBYtes? ... 1697
SBUS<sb>:CAN:FRAMe<fr>:SBC? ... 1700
SBUS<sb>:CAN:FRAMe<fr>:STARt? ... 1697
SBUS<sb>:CAN:FRAMe<fr>:STATus? ... 1697
SBUS<sb>:CAN:FRAMe<fr>:STOP? ... 1698
SBUS<sb>:CAN:FRAMe<fr>:SYMBol? ... 1698
SBUS<sb>:CAN:FRAMe<fr>:TYPE? ... 1699
SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:AF? ... 1699
SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:FCRC? ... 1703
SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:PCRC? ... 1699
SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:SDT? ... 1700
SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:SEC? ... 1700
SBUS<sb>:CAN:FRAMe<fr>:XDATa<o>:VCID? ... 1701
SBUS<sb>:CAN:NEWList ... 1660
SBUS<sb>:CAN:POSition ... 1660
SBUS<sb>:CAN:SAMPlepoint ... 1656
SBUS<sb>:CAN:SCALe ... 1660
SBUS<sb>:CAN:SIC:HYSTeresis ... 1657
SBUS<sb>:CAN:SIC:THReshold ... 1658
SBUS<sb>:CAN:SOURce ... 1656
SBUS<sb>:CAN:SWTindex? ... 1690
SBUS<sb>:CAN:SWTTime? ... 1690
SBUS<sb>:CAN:SYMBols ... 1659
SBUS<sb>:CAN:TRCVmode ... 1659
SBUS<sb>:CAN:TYPE ... 1655
SBUS<sb>:CAN:XDATa:DBITrate ... 1658

SBUS<sb>:CAN:XDATa:SAMPlepoint....1659
SBUS<sb>:EXPResult:DETail....1365
SBUS<sb>:EXPResult:EXTension....1366
SBUS<sb>:EXPResult:PATH....1366
SBUS<sb>:EXPResult:SAVE....1365
SBUS<sb>:EXPResult:TIME....1366
SBUS<sb>:FORMat....1363
SBUS<sb>:I2C:FCOunt?....1492
SBUS<sb>:I2C:FILTer:BIT....1472
SBUS<sb>:I2C:FILTer:CHKall....1468
SBUS<sb>:I2C:FILTer:CLR....1468
SBUS<sb>:I2C:FILTer:DMAX....1469
SBUS<sb>:I2C:FILTer:DMIN....1469
SBUS<sb>:I2C:FILTer:DOPerator....1470
SBUS<sb>:I2C:FILTer:ERENable....1470
SBUS<sb>:I2C:FILTer:ERRor<n>:ENABle....1470
SBUS<sb>:I2C:FILTer:FIENable....1472
SBUS<sb>:I2C:FILTer:FRAMe<fr>:ENABle....1469
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:BIT....1472
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1469
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1469
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1470
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1472
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1471
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1471
SBUS<sb>:I2C:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1471
SBUS<sb>:I2C:FILTer:FRENable....1469
SBUS<sb>:I2C:FILTer:IMAX....1471
SBUS<sb>:I2C:FILTer:IMIN....1471
SBUS<sb>:I2C:FILTer:INVert....1468
SBUS<sb>:I2C:FILTer:IOPerator....1471
SBUS<sb>:I2C:FILTer:RST....1468
SBUS<sb>:I2C:FORMat:ADDFrame....1473
SBUS<sb>:I2C:FORMat:CLR....1474
SBUS<sb>:I2C:FORMat:FCOunt?....1474
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ADDField....1475
SBUS<sb>:I2C:FORMat:FRAMe<fr>:COLor....1476
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1480
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1478
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1479
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:COLor....1477
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1481
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1480
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1479
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1478
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:NAME....1477
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:PARA....1481
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:PARB....1481
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1478
SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1481

SBUS<sb>:I2C:FORMat:FRAMe<fr>:FLDCount? 1475
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID1Field 1476
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID1Value 1477
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID2Field 1476
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID2Value 1477
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID3Field 1476
SBUS<sb>:I2C:FORMat:FRAMe<fr>:ID3Value 1477
SBUS<sb>:I2C:FORMat:FRAMe<fr>:NAME 1475
SBUS<sb>:I2C:FORMat:FRAMe<fr>:TYPE 1476
SBUS<sb>:I2C:FORMat:LOAD 1474
SBUS<sb>:I2C:FORMat:SAVE 1474
SBUS<sb>:I2C:FORMat:SELect 1473
SBUS<sb>:I2C:FRAMe<fr>:AACCess? 1492
SBUS<sb>:I2C:FRAMe<fr>:ACCess? 1492
SBUS<sb>:I2C:FRAMe<fr>:ACOMplete? 1493
SBUS<sb>:I2C:FRAMe<fr>:ADBStart? 1493
SBUS<sb>:I2C:FRAMe<fr>:ADDRess? 1493
SBUS<sb>:I2C:FRAMe<fr>:ADEVice? 1494
SBUS<sb>:I2C:FRAMe<fr>:AMODe? 1494
SBUS<sb>:I2C:FRAMe<fr>:ASTart? 1494
SBUS<sb>:I2C:FRAMe<fr>:BITRate? 1495
SBUS<sb>:I2C:FRAMe<fr>:DATA? 1497
SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:FVALue? 1495
SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:LABel? 1496
SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:STARt? 1496
SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:STATe? 1496
SBUS<sb>:I2C:FRAMe<fr>:FLD<fl>:VALue? 1497
SBUS<sb>:I2C:FRAMe<fr>:FLDCount? 1495
SBUS<sb>:I2C:FRAMe<fr>:RWBStart? 1498
SBUS<sb>:I2C:FRAMe<fr>:STARt? 1498
SBUS<sb>:I2C:FRAMe<fr>:STATus? 1498
SBUS<sb>:I2C:FRAMe<fr>:STOP? 1499
SBUS<sb>:I2C:FRAMe<fr>:SYMBol? 1499
SBUS<sb>:I2C:NEWList 1467
SBUS<sb>:I2C:POSition 1466
SBUS<sb>:I2C:SCALe 1466
SBUS<sb>:I2C:SCL:HYSTeresis 1464
SBUS<sb>:I2C:SCL:SOURce 1465
SBUS<sb>:I2C:SCL:THReshold 1465
SBUS<sb>:I2C:SDA:HYSTeresis 1465
SBUS<sb>:I2C:SDA:SOURce 1465
SBUS<sb>:I2C:SDA:THReshold 1466
SBUS<sb>:I2C:SWTindex? 1491
SBUS<sb>:I2C:SWTTime? 1491
SBUS<sb>:I2C:SYMBols 1466
SBUS<sb>:I3C:FCOunt? 1525
SBUS<sb>:I3C:FILTer:BIT 1508
SBUS<sb>:I3C:FILTer:CHKall 1504
SBUS<sb>:I3C:FILTer:CLR 1505
SBUS<sb>:I3C:FILTer:DMAX 1506

SBUS<sb>:I3C:FILTer:DMIN ... 1506
SBUS<sb>:I3C:FILTer:DOPerator ... 1506
SBUS<sb>:I3C:FILTer:ERENable ... 1507
SBUS<sb>:I3C:FILTer:ERRor<n>:ENABle ... 1507
SBUS<sb>:I3C:FILTer:FIENable ... 1509
SBUS<sb>:I3C:FILTer:FRAMe<fr>:ENABle ... 1505
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1508
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1506
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1506
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1506
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1509
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1507
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1508
SBUS<sb>:I3C:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1508
SBUS<sb>:I3C:FILTer:FRENable ... 1505
SBUS<sb>:I3C:FILTer:IMAX ... 1507
SBUS<sb>:I3C:FILTer:IMIN ... 1508
SBUS<sb>:I3C:FILTer:INVert ... 1505
SBUS<sb>:I3C:FILTer:IOPerator ... 1508
SBUS<sb>:I3C:FILTer:RST ... 1505
SBUS<sb>:I3C:FORMat:ADDFrame ... 1510
SBUS<sb>:I3C:FORMat:CLR ... 1511
SBUS<sb>:I3C:FORMat:FCOunt? ... 1511
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ADDField ... 1511
SBUS<sb>:I3C:FORMat:FRAMe<fr>:COLor ... 1512
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ... 1516
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:BITCount ... 1514
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:BITorder ... 1515
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:COLor ... 1514
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ... 1517
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ... 1517
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FORMat ... 1516
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ... 1514
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:NAME ... 1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:PARA ... 1518
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:PARB ... 1518
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:REPetition ... 1515
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLD<fl>:UNIT ... 1517
SBUS<sb>:I3C:FORMat:FRAMe<fr>:FLDCount? ... 1512
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID1Field ... 1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID1Value ... 1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID2Field ... 1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID2Value ... 1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID3Field ... 1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:ID3Value ... 1513
SBUS<sb>:I3C:FORMat:FRAMe<fr>:NAME ... 1512
SBUS<sb>:I3C:FORMat:FRAMe<fr>:TYPE ... 1512
SBUS<sb>:I3C:FORMat:LOAD ... 1510
SBUS<sb>:I3C:FORMat:SAVE ... 1510
SBUS<sb>:I3C:FORMat:SELect ... 1510

SBUS<sb>:I3C:FRAMe<fr>:ACK? 1528
SBUS<sb>:I3C:FRAMe<fr>:ACKStart? 1529
SBUS<sb>:I3C:FRAMe<fr>:ADDRess? 1525
SBUS<sb>:I3C:FRAMe<fr>:ASTart? 1525
SBUS<sb>:I3C:FRAMe<fr>:BITRate? 1526
SBUS<sb>:I3C:FRAMe<fr>:COMMand? 1531
SBUS<sb>:I3C:FRAMe<fr>:CRC? 1531
SBUS<sb>:I3C:FRAMe<fr>:DATA? 1526
SBUS<sb>:I3C:FRAMe<fr>:FADDress? 1531
SBUS<sb>:I3C:FRAMe<fr>:FCOMmand? 1532
SBUS<sb>:I3C:FRAMe<fr>:FCRC? 1532
SBUS<sb>:I3C:FRAMe<fr>:FLD<fl>:FVALue? 1529
SBUS<sb>:I3C:FRAMe<fr>:FLD<fl>:LABel? 1529
SBUS<sb>:I3C:FRAMe<fr>:FLD<fl>:STATe? 1530
SBUS<sb>:I3C:FRAMe<fr>:FLD<fl>:VALue? 1530
SBUS<sb>:I3C:FRAMe<fr>:FLDCount? 1526
SBUS<sb>:I3C:FRAMe<fr>:RWBit? 1532
SBUS<sb>:I3C:FRAMe<fr>:RWBStart? 1526
SBUS<sb>:I3C:FRAMe<fr>:STARt? 1527
SBUS<sb>:I3C:FRAMe<fr>:STATus? 1527
SBUS<sb>:I3C:FRAMe<fr>:STOP? 1528
SBUS<sb>:I3C:FRAMe<fr>:SYMBol? 1528
SBUS<sb>:I3C:FRAMe<fr>:TYPE? 1532
SBUS<sb>:I3C:MINGap:SELect 1502
SBUS<sb>:I3C:MINGap:WIDTh 1502
SBUS<sb>:I3C:NEWList 1503
SBUS<sb>:I3C:POSition 1502
SBUS<sb>:I3C:SCALe 1503
SBUS<sb>:I3C:SCL:HYSTeresis 1500
SBUS<sb>:I3C:SCL:SOURce 1500
SBUS<sb>:I3C:SCL:THReshold 1501
SBUS<sb>:I3C:SDA:HYSTeresis 1501
SBUS<sb>:I3C:SDA:SOURce 1501
SBUS<sb>:I3C:SDA:THReshold 1502
SBUS<sb>:I3C:SWTindex? 1523
SBUS<sb>:I3C:SWTTime? 1524
SBUS<sb>:I3C:SYMBols 1503
SBUS<sb>:LIN:BITRate 1703
SBUS<sb>:LIN:DATA:HYSTeresis 1704
SBUS<sb>:LIN:DATA:SOURce 1704
SBUS<sb>:LIN:DATA:THReshold 1704
SBUS<sb>:LIN:FCOunt? 1730
SBUS<sb>:LIN:FILTer:BIT 1708
SBUS<sb>:LIN:FILTer:CHKall 1707
SBUS<sb>:LIN:FILTer:CLR 1707
SBUS<sb>:LIN:FILTer:DMAX 1708
SBUS<sb>:LIN:FILTer:DMIN 1709
SBUS<sb>:LIN:FILTer:DOPerator 1709
SBUS<sb>:LIN:FILTer:ERENable 1709
SBUS<sb>:LIN:FILTer:ERRor<n>:ENABle 1709

SBUS<sb>:LIN:FILTer:FIENable....1710
SBUS<sb>:LIN:FILTer:FRAMe<fr>:ENABle....1710
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:BIT....1708
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1708
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1709
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1709
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1710
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1710
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1711
SBUS<sb>:LIN:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1711
SBUS<sb>:LIN:FILTer:FRENable....1710
SBUS<sb>:LIN:FILTer:IMAX....1710
SBUS<sb>:LIN:FILTer:IMIN....1711
SBUS<sb>:LIN:FILTer:INVert....1707
SBUS<sb>:LIN:FILTer:IOPerator....1711
SBUS<sb>:LIN:FILTer:RST....1708
SBUS<sb>:LIN:FORMat:ADDFrame....1713
SBUS<sb>:LIN:FORMat:CLR....1713
SBUS<sb>:LIN:FORMat:FCOunt?....1714
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ADDField....1714
SBUS<sb>:LIN:FORMat:FRAMe<fr>:COLor....1715
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1719
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1717
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1718
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:COLor....1716
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1720
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1719
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1718
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1717
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:NAME....1716
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:PARA....1720
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:PARB....1720
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1717
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1720
SBUS<sb>:LIN:FORMat:FRAMe<fr>:FLDCount?....1714
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID1Field....1715
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID1Value....1716
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID2Field....1715
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID2Value....1716
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID3Field....1715
SBUS<sb>:LIN:FORMat:FRAMe<fr>:ID3Value....1716
SBUS<sb>:LIN:FORMat:FRAMe<fr>:NAME....1714
SBUS<sb>:LIN:FORMat:FRAMe<fr>:TYPE....1715
SBUS<sb>:LIN:FORMat:LOAD....1713
SBUS<sb>:LIN:FORMat:SAVE....1713
SBUS<sb>:LIN:FORMat:SELect....1712
SBUS<sb>:LIN:FRAMe<fr>:BITRate?....1731
SBUS<sb>:LIN:FRAMe<fr>:CSValue?....1733
SBUS<sb>:LIN:FRAMe<fr>:DATA?....1733
SBUS<sb>:LIN:FRAMe<fr>:FLD<fl>:FVALue?....1732

SBUS<sb>:LIN:FRAMe<fr>:FLD<fl>:LABel? 1732
SBUS<sb>:LIN:FRAMe<fr>:FLD<fl>:STATe? 1731
SBUS<sb>:LIN:FRAMe<fr>:FLD<fl>:VALue? 1732
SBUS<sb>:LIN:FRAMe<fr>:FLDCount? 1731
SBUS<sb>:LIN:FRAMe<fr>:IDPValue? 1733
SBUS<sb>:LIN:FRAMe<fr>:IDValue? 1734
SBUS<sb>:LIN:FRAMe<fr>:PIDentifier? 1735
SBUS<sb>:LIN:FRAMe<fr>:STARt? 1734
SBUS<sb>:LIN:FRAMe<fr>:STATus? 1734
SBUS<sb>:LIN:FRAMe<fr>:STOP? 1735
SBUS<sb>:LIN:FRAMe<fr>:SYMBol? 1735
SBUS<sb>:LIN:NEWList 1706
SBUS<sb>:LIN:POLarity 1705
SBUS<sb>:LIN:POSition 1705
SBUS<sb>:LIN:SCALe 1705
SBUS<sb>:LIN:STANdard 1705
SBUS<sb>:LIN:SWTindex? 1729
SBUS<sb>:LIN:SWTTime? 1730
SBUS<sb>:LIN:SYMBols 1706
SBUS<sb>:MANCh:BITRate:SELect 1628
SBUS<sb>:MANCh:BITRate:WIDTh 1628
SBUS<sb>:MANCh:DATA:HYSTeresis 1629
SBUS<sb>:MANCh:DATA:PHASe 1626
SBUS<sb>:MANCh:DATA:POLarity 1626
SBUS<sb>:MANCh:DATA:SOURce 1625
SBUS<sb>:MANCh:DATA:THLower 1629
SBUS<sb>:MANCh:DATA:THUPper 1628
SBUS<sb>:MANCh:ENABle:HYSTeresis 1629
SBUS<sb>:MANCh:ENABle:POLarity 1627
SBUS<sb>:MANCh:ENABle:SOURce 1627
SBUS<sb>:MANCh:ENABle:THReshold 1629
SBUS<sb>:MANCh:FCOunt? 1650
SBUS<sb>:MANCh:FILTer:BIT 1632
SBUS<sb>:MANCh:FILTer:CHKall 1631
SBUS<sb>:MANCh:FILTer:CLR 1631
SBUS<sb>:MANCh:FILTer:DMAX 1633
SBUS<sb>:MANCh:FILTer:DMIN 1633
SBUS<sb>:MANCh:FILTer:DOPerator 1633
SBUS<sb>:MANCh:FILTer:ERENable 1632
SBUS<sb>:MANCh:FILTer:ERRor<n>:ENABle 1632
SBUS<sb>:MANCh:FILTer:FIENable 1634
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:ENABle 1634
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:BIT 1632
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DMAX 1633
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DMIN 1633
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:DOPerator 1633
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:ENABle 1634
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IMAX 1634
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IMIN 1635
SBUS<sb>:MANCh:FILTer:FRAMe<fr>:FLD<fl>:IOPerator 1635

SBUS<sb>:MANCh:FILTer:FRENable....1634
SBUS<sb>:MANCh:FILTer:IMAX....1634
SBUS<sb>:MANCh:FILTer:IMIN....1635
SBUS<sb>:MANCh:FILTer:INVert....1631
SBUS<sb>:MANCh:FILTer:IOPerator....1635
SBUS<sb>:MANCh:FILTer:RST....1632
SBUS<sb>:MANCh:FORMat:ADDFrame....1636
SBUS<sb>:MANCh:FORMat:CLR....1637
SBUS<sb>:MANCh:FORMat:FCOunt?....1637
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:ADDField....1638
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:COLor....1638
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1642
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1639
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1641
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:CLMN....1641
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:COLor....1639
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:CONDition....1640
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1643
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1642
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1640
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:NAME....1639
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:PARA....1643
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:PARB....1643
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1643
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:FLDCount?....1638
SBUS<sb>:MANCh:FORMat:FRAMe<fr>:NAME....1638
SBUS<sb>:MANCh:FORMat:LOAD....1636
SBUS<sb>:MANCh:FORMat:SAVE....1637
SBUS<sb>:MANCh:FRAMe<fr>:CONE?....1654
SBUS<sb>:MANCh:FRAMe<fr>:CTHRee?....1654
SBUS<sb>:MANCh:FRAMe<fr>:CTWO?....1654
SBUS<sb>:MANCh:FRAMe<fr>:FLD<fl>:FVALue?....1652
SBUS<sb>:MANCh:FRAMe<fr>:FLD<fl>:LABel?....1651
SBUS<sb>:MANCh:FRAMe<fr>:FLD<fl>:STATe?....1651
SBUS<sb>:MANCh:FRAMe<fr>:FLD<fl>:VALue?....1652
SBUS<sb>:MANCh:FRAMe<fr>:FLDCount?....1650
SBUS<sb>:MANCh:FRAMe<fr>:STARt?....1653
SBUS<sb>:MANCh:FRAMe<fr>:STATus?....1652
SBUS<sb>:MANCh:FRAMe<fr>:STOP?....1653
SBUS<sb>:MANCh:FRAMe<fr>:TYPE?....1654
SBUS<sb>:MANCh:FRAMe<fr>:WBRate?....1651
SBUS<sb>:MANCh:MINGap:SELect....1627
SBUS<sb>:MANCh:MINGap:WIDTh....1628
SBUS<sb>:MANCh:POSition....1630
SBUS<sb>:MANCh:SCALe....1630
SBUS<sb>:MANCh:SWTindex?....1649
SBUS<sb>:MANCh:SWTTime?....1649
SBUS<sb>:MILStd:FILTer:BIT....1805
SBUS<sb>:MILStd:FILTer:CHKall....1804
SBUS<sb>:MILStd:FILTer:CLR....1804

SBUS<sb>:MILStd:FILTer:DMAX ... 1806
SBUS<sb>:MILStd:FILTer:DMIN ... 1806
SBUS<sb>:MILStd:FILTer:DOPerator ... 1806
SBUS<sb>:MILStd:FILTer:ERENable ... 1805
SBUS<sb>:MILStd:FILTer:ERRor<n>:ENABle ... 1805
SBUS<sb>:MILStd:FILTer:FIENable ... 1808
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:ENABle ... 1808
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1805
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1806
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1806
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1806
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1808
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1807
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1807
SBUS<sb>:MILStd:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1808
SBUS<sb>:MILStd:FILTer:FRENable ... 1808
SBUS<sb>:MILStd:FILTer:IMAX ... 1807
SBUS<sb>:MILStd:FILTer:IMIN ... 1807
SBUS<sb>:MILStd:FILTer:INVert ... 1804
SBUS<sb>:MILStd:FILTer:IOPerator ... 1808
SBUS<sb>:MILStd:FILTer:RST ... 1805
SBUS<sb>:MILStd:FORMat:ADDFrame ... 1810
SBUS<sb>:MILStd:FORMat:CLR ... 1810
SBUS<sb>:MILStd:FORMat:FCOunt? ... 1811
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ADDField ... 1811
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:COLor ... 1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ... 1816
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:BITCount ... 1814
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:BITorder ... 1814
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:COLor ... 1813
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ... 1816
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ... 1816
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:FORMat ... 1815
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ... 1813
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:NAME ... 1813
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:PARA ... 1817
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:PARB ... 1817
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:REPetition ... 1814
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLD<fl>:UNIT ... 1817
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:FLDCount? ... 1811
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID1Field ... 1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID1Value ... 1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID2Field ... 1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID2Value ... 1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID3Field ... 1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:ID3Value ... 1812
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:NAME ... 1811
SBUS<sb>:MILStd:FORMat:FRAMe<fr>:TYPE ... 1812
SBUS<sb>:MILStd:FORMat:LOAD ... 1810
SBUS<sb>:MILStd:FORMat:SAVE ... 1810

SBUS<sb>:MILStd:FORMat:SELect....1809
SBUS<sb>:MILStd:NEWList....1802
SBUS<sb>:MILStd:POLarity....1801
SBUS<sb>:MILStd:POSition....1802
SBUS<sb>:MILStd:SCALe....1803
SBUS<sb>:MILStd:SOURce....1801
SBUS<sb>:MILStd:SWTindex?....1823
SBUS<sb>:MILStd:SWTTime?....1823
SBUS<sb>:MILStd:SYMBols....1803
SBUS<sb>:MILStd:THReshold:HIGH....1801
SBUS<sb>:MILStd:THReshold:HYSTeresis....1802
SBUS<sb>:MILStd:THReshold:LOW....1802
SBUS<sb>:MILStd:WCOunt?....1824
SBUS<sb>:MILStd:WORD<w>:BITRate?....1824
SBUS<sb>:MILStd:WORD<w>:DATA?....1826
SBUS<sb>:MILStd:WORD<w>:FLD<fl>:FVALue?....1826
SBUS<sb>:MILStd:WORD<w>:FLD<fl>:LABel?....1825
SBUS<sb>:MILStd:WORD<w>:FLD<fl>:STATe?....1824
SBUS<sb>:MILStd:WORD<w>:FLD<fl>:VALue?....1825
SBUS<sb>:MILStd:WORD<w>:FLDCount?....1824
SBUS<sb>:MILStd:WORD<w>:INFO?....1826
SBUS<sb>:MILStd:WORD<w>:RTADdress?....1826
SBUS<sb>:MILStd:WORD<w>:STARt?....1827
SBUS<sb>:MILStd:WORD<w>:STATus?....1827
SBUS<sb>:MILStd:WORD<w>:STOP?....1828
SBUS<sb>:MILStd:WORD<w>:SYMBol?....1828
SBUS<sb>:MILStd:WORD<w>:TYPE?....1828
SBUS<sb>:NRZC:CLK:HYSTeresis....1569
SBUS<sb>:NRZC:CLK:POLarity....1566
SBUS<sb>:NRZC:CLK:SOURce....1566
SBUS<sb>:NRZC:CLK:THReshold....1569
SBUS<sb>:NRZC:DATA:HYSTeresis....1570
SBUS<sb>:NRZC:DATA:POLarity....1567
SBUS<sb>:NRZC:DATA:SOURce....1567
SBUS<sb>:NRZC:DATA:THReshold....1569
SBUS<sb>:NRZC:ENABle:HYSTeresis....1570
SBUS<sb>:NRZC:ENABle:POLarity....1568
SBUS<sb>:NRZC:ENABle:SOURce....1568
SBUS<sb>:NRZC:ENABle:THReshold....1570
SBUS<sb>:NRZC:FCOunt?....1591
SBUS<sb>:NRZC:FILTer:BIT....1572
SBUS<sb>:NRZC:FILTer:CHKall....1572
SBUS<sb>:NRZC:FILTer:CLR....1572
SBUS<sb>:NRZC:FILTer:DMAX....1573
SBUS<sb>:NRZC:FILTer:DMIN....1574
SBUS<sb>:NRZC:FILTer:DOPerator....1574
SBUS<sb>:NRZC:FILTer:ERENable....1573
SBUS<sb>:NRZC:FILTer:ERRor<n>:ENABle....1573
SBUS<sb>:NRZC:FILTer:FIENable....1574
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:ENABle....1575

SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1572
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1573
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1574
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1574
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1574
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1575
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1576
SBUS<sb>:NRZC:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1576
SBUS<sb>:NRZC:FILTer:FRENable ... 1575
SBUS<sb>:NRZC:FILTer:IMAX ... 1575
SBUS<sb>:NRZC:FILTer:IMIN ... 1576
SBUS<sb>:NRZC:FILTer:INVert ... 1572
SBUS<sb>:NRZC:FILTer:IOPerator ... 1576
SBUS<sb>:NRZC:FILTer:RST ... 1572
SBUS<sb>:NRZC:FORMat:ADDFrame ... 1577
SBUS<sb>:NRZC:FORMat:CLR ... 1578
SBUS<sb>:NRZC:FORMat:FCOunt? ... 1578
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:ADDField ... 1578
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:COLor ... 1579
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ... 1583
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:BITCount ... 1580
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:BITorder ... 1582
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:CLMN ... 1582
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:COLor ... 1580
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:CONDition ... 1580
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ... 1583
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ... 1583
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:FORMat ... 1581
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:NAME ... 1579
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:PARA ... 1584
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:PARB ... 1584
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLD<fl>:UNIT ... 1584
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:FLDCount? ... 1578
SBUS<sb>:NRZC:FORMat:FRAMe<fr>:NAME ... 1579
SBUS<sb>:NRZC:FORMat:LOAD ... 1577
SBUS<sb>:NRZC:FORMat:SAVE ... 1577
SBUS<sb>:NRZC:FRAMe<fr>:CONE? ... 1595
SBUS<sb>:NRZC:FRAMe<fr>:CTHRee? ... 1595
SBUS<sb>:NRZC:FRAMe<fr>:CTWO? ... 1595
SBUS<sb>:NRZC:FRAMe<fr>:FLD<fl>:FVALue? ... 1593
SBUS<sb>:NRZC:FRAMe<fr>:FLD<fl>:LABel? ... 1592
SBUS<sb>:NRZC:FRAMe<fr>:FLD<fl>:STATe? ... 1592
SBUS<sb>:NRZC:FRAMe<fr>:FLD<fl>:VALue? ... 1593
SBUS<sb>:NRZC:FRAMe<fr>:FLDCount? ... 1591
SBUS<sb>:NRZC:FRAMe<fr>:STARt? ... 1594
SBUS<sb>:NRZC:FRAMe<fr>:STATus? ... 1593
SBUS<sb>:NRZC:FRAMe<fr>:STOP? ... 1594
SBUS<sb>:NRZC:FRAMe<fr>:TYPE? ... 1594
SBUS<sb>:NRZC:FRAMe<fr>:WBRate? ... 1591
SBUS<sb>:NRZC:MINGap:SELect ... 1568

SBUS<sb>:NRZC:MINGap:WIDTh....1569
SBUS<sb>:NRZC:POSition....1570
SBUS<sb>:NRZC:SCALe....1571
SBUS<sb>:NRZC:SWTindex?....1590
SBUS<sb>:NRZC:SWTTime?....1590
SBUS<sb>:NRZU:BITRate:SELect....1598
SBUS<sb>:NRZU:BITRate:WIDTh....1598
SBUS<sb>:NRZU:DATA:HYSTeresis....1599
SBUS<sb>:NRZU:DATA:POLarity....1596
SBUS<sb>:NRZU:DATA:SOURce....1596
SBUS<sb>:NRZU:DATA:THReshold....1599
SBUS<sb>:NRZU:ENABle:HYSTeresis....1599
SBUS<sb>:NRZU:ENABle:POLarity....1597
SBUS<sb>:NRZU:ENABle:SOURce....1597
SBUS<sb>:NRZU:ENABle:THReshold....1599
SBUS<sb>:NRZU:FCOunt?....1620
SBUS<sb>:NRZU:FILTer:BIT....1602
SBUS<sb>:NRZU:FILTer:CHKall....1601
SBUS<sb>:NRZU:FILTer:CLR....1601
SBUS<sb>:NRZU:FILTer:DMAX....1603
SBUS<sb>:NRZU:FILTer:DMIN....1603
SBUS<sb>:NRZU:FILTer:DOPerator....1603
SBUS<sb>:NRZU:FILTer:ERENable....1602
SBUS<sb>:NRZU:FILTer:ERRor<n>:ENABle....1602
SBUS<sb>:NRZU:FILTer:FIENable....1604
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:ENABle....1604
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:BIT....1602
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1603
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1603
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1603
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1604
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1604
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1605
SBUS<sb>:NRZU:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1605
SBUS<sb>:NRZU:FILTer:FRENable....1604
SBUS<sb>:NRZU:FILTer:IMAX....1604
SBUS<sb>:NRZU:FILTer:IMIN....1605
SBUS<sb>:NRZU:FILTer:INVert....1601
SBUS<sb>:NRZU:FILTer:IOPerator....1605
SBUS<sb>:NRZU:FILTer:RST....1602
SBUS<sb>:NRZU:FORMat:ADDFrame....1606
SBUS<sb>:NRZU:FORMat:CLR....1607
SBUS<sb>:NRZU:FORMat:FCOunt?....1607
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:ADDField....1608
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:COLor....1608
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1612
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1609
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1611
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:CLMN....1611
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:COLor....1609

SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:CONDition....1610
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1613
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1612
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1610
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:NAME....1609
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:PARA....1613
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:PARB....1613
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1613
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:FLDCount?....1608
SBUS<sb>:NRZU:FORMat:FRAMe<fr>:NAME....1608
SBUS<sb>:NRZU:FORMat:LOAD....1606
SBUS<sb>:NRZU:FORMat:SAVE....1607
SBUS<sb>:NRZU:FRAMe<fr>:CONE?....1624
SBUS<sb>:NRZU:FRAMe<fr>:CTHRee?....1625
SBUS<sb>:NRZU:FRAMe<fr>:CTWO?....1624
SBUS<sb>:NRZU:FRAMe<fr>:FLD<fl>:FVALue?....1622
SBUS<sb>:NRZU:FRAMe<fr>:FLD<fl>:LABel?....1622
SBUS<sb>:NRZU:FRAMe<fr>:FLD<fl>:STATe?....1621
SBUS<sb>:NRZU:FRAMe<fr>:FLD<fl>:VALue?....1622
SBUS<sb>:NRZU:FRAMe<fr>:FLDCount?....1620
SBUS<sb>:NRZU:FRAMe<fr>:STARt?....1623
SBUS<sb>:NRZU:FRAMe<fr>:STATus?....1622
SBUS<sb>:NRZU:FRAMe<fr>:STOP?....1623
SBUS<sb>:NRZU:FRAMe<fr>:TYPE?....1624
SBUS<sb>:NRZU:FRAMe<fr>:WBRate?....1621
SBUS<sb>:NRZU:MINGap:SELect....1597
SBUS<sb>:NRZU:MINGap:WIDTh....1598
SBUS<sb>:NRZU:POSition....1600
SBUS<sb>:NRZU:SCALe....1600
SBUS<sb>:NRZU:SWTindex?....1619
SBUS<sb>:NRZU:SWTTime?....1619
SBUS<sb>:OSPI:CK:HYSTeresis....1436
SBUS<sb>:OSPI:CK:SOURce....1436
SBUS<sb>:OSPI:CK:THReshold....1437
SBUS<sb>:OSPI:CSEL:HYSTeresis....1435
SBUS<sb>:OSPI:CSEL:POLarity....1435
SBUS<sb>:OSPI:CSEL:SOURce....1435
SBUS<sb>:OSPI:CSEL:THReshold....1435
SBUS<sb>:OSPI:DS:HYSTeresis....1439
SBUS<sb>:OSPI:DS:SOURce....1440
SBUS<sb>:OSPI:DS:THReshold....1440
SBUS<sb>:OSPI:FCOunt?....1457
SBUS<sb>:OSPI:FILTer:BIT....1442
SBUS<sb>:OSPI:FILTer:CHKall....1441
SBUS<sb>:OSPI:FILTer:CLR....1441
SBUS<sb>:OSPI:FILTer:DMAX....1442
SBUS<sb>:OSPI:FILTer:DMIN....1443
SBUS<sb>:OSPI:FILTer:DOPerator....1443
SBUS<sb>:OSPI:FILTer:ERENable....1445
SBUS<sb>:OSPI:FILTer:ERRor<n>:ENABle....1445

SBUS<sb>:OSPI:FILTer:FIENable....1445
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:ENABle....1445
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:BIT....1442
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1442
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1443
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1443
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1445
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1443
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1444
SBUS<sb>:OSPI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1444
SBUS<sb>:OSPI:FILTer:FRENable....1445
SBUS<sb>:OSPI:FILTer:IMAX....1443
SBUS<sb>:OSPI:FILTer:IMIN....1444
SBUS<sb>:OSPI:FILTer:INVert....1441
SBUS<sb>:OSPI:FILTer:IOPerator....1444
SBUS<sb>:OSPI:FILTer:RST....1442
SBUS<sb>:OSPI:FRAMe<fr>:ADDR?....1458
SBUS<sb>:OSPI:FRAMe<fr>:ADFormatted?....1458
SBUS<sb>:OSPI:FRAMe<fr>:BITRate?....1458
SBUS<sb>:OSPI:FRAMe<fr>:CEXTension?....1461
SBUS<sb>:OSPI:FRAMe<fr>:CMD?....1461
SBUS<sb>:OSPI:FRAMe<fr>:CMDFormatted?....1461
SBUS<sb>:OSPI:FRAMe<fr>:CXFormatted?....1462
SBUS<sb>:OSPI:FRAMe<fr>:DATA?....1458
SBUS<sb>:OSPI:FRAMe<fr>:DCOunt?....1462
SBUS<sb>:OSPI:FRAMe<fr>:FDATa?....1459
SBUS<sb>:OSPI:FRAMe<fr>:FLD<fl>:FVALue?....1460
SBUS<sb>:OSPI:FRAMe<fr>:FLD<fl>:LABel?....1460
SBUS<sb>:OSPI:FRAMe<fr>:FLD<fl>:STATe?....1460
SBUS<sb>:OSPI:FRAMe<fr>:FLD<fl>:VALue?....1459
SBUS<sb>:OSPI:FRAMe<fr>:FLDCount?....1459
SBUS<sb>:OSPI:FRAMe<fr>:STARt?....1463
SBUS<sb>:OSPI:FRAMe<fr>:STATus?....1463
SBUS<sb>:OSPI:FRAMe<fr>:STOP?....1463
SBUS<sb>:OSPI:FRAMe<fr>:TYPE?....1462
SBUS<sb>:OSPI:IOFR:HYSTeresis....1437
SBUS<sb>:OSPI:IOFR:POLarity....1437
SBUS<sb>:OSPI:IOFR:POSition....1438
SBUS<sb>:OSPI:IOFR:SCALe....1438
SBUS<sb>:OSPI:IOFR:SOURce....1439
SBUS<sb>:OSPI:IOFR:THReshold....1439
SBUS<sb>:OSPI:IOFV:HYSTeresis....1437
SBUS<sb>:OSPI:IOFV:POLarity....1437
SBUS<sb>:OSPI:IOFV:POSition....1438
SBUS<sb>:OSPI:IOFV:SCALe....1438
SBUS<sb>:OSPI:IOFV:SOURce....1439
SBUS<sb>:OSPI:IOFV:THReshold....1439
SBUS<sb>:OSPI:IOONe:HYSTeresis....1437
SBUS<sb>:OSPI:IOONe:POLarity....1437
SBUS<sb>:OSPI:IOONe:POSition....1438

SBUS<sb>:OSPI:IOONe:SCALe....1438
SBUS<sb>:OSPI:IOONe:SOURce....1438
SBUS<sb>:OSPI:IOONe:THReshold....1439
SBUS<sb>:OSPI:IOSV:HYSTeresis....1437
SBUS<sb>:OSPI:IOSV:POLarity....1437
SBUS<sb>:OSPI:IOSV:POSition....1438
SBUS<sb>:OSPI:IOSV:SCALe....1438
SBUS<sb>:OSPI:IOSV:SOURce....1439
SBUS<sb>:OSPI:IOSV:THReshold....1439
SBUS<sb>:OSPI:IOSX:HYSTeresis....1437
SBUS<sb>:OSPI:IOSX:POLarity....1437
SBUS<sb>:OSPI:IOSX:POSition....1438
SBUS<sb>:OSPI:IOSX:SCALe....1438
SBUS<sb>:OSPI:IOSX:SOURce....1439
SBUS<sb>:OSPI:IOSX:THReshold....1439
SBUS<sb>:OSPI:IOTHree:HYSTeresis....1437
SBUS<sb>:OSPI:IOTHree:POLarity....1437
SBUS<sb>:OSPI:IOTHree:POSition....1438
SBUS<sb>:OSPI:IOTHree:SCALe....1438
SBUS<sb>:OSPI:IOTHree:SOURce....1438
SBUS<sb>:OSPI:IOTHree:THReshold....1439
SBUS<sb>:OSPI:IOTWo:HYSTeresis....1437
SBUS<sb>:OSPI:IOTWo:POLarity....1437
SBUS<sb>:OSPI:IOTWo:POSition....1438
SBUS<sb>:OSPI:IOTWo:SCALe....1438
SBUS<sb>:OSPI:IOTWo:SOURce....1438
SBUS<sb>:OSPI:IOTWo:THReshold....1439
SBUS<sb>:OSPI:IOZero:HYSTeresis....1437
SBUS<sb>:OSPI:IOZero:POLarity....1437
SBUS<sb>:OSPI:IOZero:POSition....1438
SBUS<sb>:OSPI:IOZero:SCALe....1438
SBUS<sb>:OSPI:IOZero:SOURce....1438
SBUS<sb>:OSPI:IOZero:THReshold....1439
SBUS<sb>:OSPI:LDOPcode....1447
SBUS<sb>:OSPI:MODE....1436
SBUS<sb>:OSPI:OPCode:APPend....1446
SBUS<sb>:OSPI:OPCode:DALL....1446
SBUS<sb>:OSPI:OPCode:DELete....1446
SBUS<sb>:OSPI:OPCode:ITEM<n>:ADBYtes....1449
SBUS<sb>:OSPI:OPCode:ITEM<n>:ADMode....1449
SBUS<sb>:OSPI:OPCode:ITEM<n>:CMEXtension....1449
SBUS<sb>:OSPI:OPCode:ITEM<n>:CMMode....1448
SBUS<sb>:OSPI:OPCode:ITEM<n>:CODE....1448
SBUS<sb>:OSPI:OPCode:ITEM<n>:DTMode....1451
SBUS<sb>:OSPI:OPCode:ITEM<n>:DTTYpe....1450
SBUS<sb>:OSPI:OPCode:ITEM<n>:LACYcles....1450
SBUS<sb>:OSPI:OPCode:ITEM<n>:NAME....1448
SBUS<sb>:OSPI:OPCode:RESet....1447
SBUS<sb>:OSPI:OPCode:SIZE?....1447
SBUS<sb>:OSPI:SVOP....1447

SBUS<sb>:OSPI:SWTindex? 1456
SBUS<sb>:OSPI:SWTTime? 1457
SBUS<sb>:QSPI:CSEL:HYSTeresis 1402
SBUS<sb>:QSPI:CSEL:POLarity 1402
SBUS<sb>:QSPI:CSEL:SOURce 1403
SBUS<sb>:QSPI:CSEL:THReshold 1403
SBUS<sb>:QSPI:FCOunt? 1428
SBUS<sb>:QSPI:FILTer:BIT 1412
SBUS<sb>:QSPI:FILTer:CHKall 1412
SBUS<sb>:QSPI:FILTer:CLR 1412
SBUS<sb>:QSPI:FILTer:DMAX 1413
SBUS<sb>:QSPI:FILTer:DMIN 1413
SBUS<sb>:QSPI:FILTer:DOPerator 1414
SBUS<sb>:QSPI:FILTer:ERENable 1415
SBUS<sb>:QSPI:FILTer:ERRor<n>:ENABle 1415
SBUS<sb>:QSPI:FILTer:FIENable 1415
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:ENABle 1416
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:BIT 1412
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:DMAX 1413
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:DMIN 1413
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator 1414
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:ENABle 1415
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:IMAX 1414
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:IMIN 1414
SBUS<sb>:QSPI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator 1415
SBUS<sb>:QSPI:FILTer:FRENable 1416
SBUS<sb>:QSPI:FILTer:IMAX 1414
SBUS<sb>:QSPI:FILTer:IMIN 1414
SBUS<sb>:QSPI:FILTer:INVert 1412
SBUS<sb>:QSPI:FILTer:IOPerator 1415
SBUS<sb>:QSPI:FILTer:RST 1412
SBUS<sb>:QSPI:FRAMe<fr>:ADDR? 1428
SBUS<sb>:QSPI:FRAMe<fr>:ALT? 1428
SBUS<sb>:QSPI:FRAMe<fr>:BITRate? 1428
SBUS<sb>:QSPI:FRAMe<fr>:DATA? 1429
SBUS<sb>:QSPI:FRAMe<fr>:FDATa? 1429
SBUS<sb>:QSPI:FRAMe<fr>:FLD<fl>:FVALue? 1430
SBUS<sb>:QSPI:FRAMe<fr>:FLD<fl>:LABel? 1430
SBUS<sb>:QSPI:FRAMe<fr>:FLD<fl>:STATe? 1431
SBUS<sb>:QSPI:FRAMe<fr>:FLD<fl>:VALue? 1430
SBUS<sb>:QSPI:FRAMe<fr>:FLDCount? 1429
SBUS<sb>:QSPI:FRAMe<fr>:ILBL? 1431
SBUS<sb>:QSPI:FRAMe<fr>:IVALue? 1431
SBUS<sb>:QSPI:FRAMe<fr>:STARt? 1432
SBUS<sb>:QSPI:FRAMe<fr>:STATus? 1432
SBUS<sb>:QSPI:FRAMe<fr>:STOP? 1433
SBUS<sb>:QSPI:FRAMe<fr>:TYPE? 1432
SBUS<sb>:QSPI:INSTruction 1403
SBUS<sb>:QSPI:IOONe:HYSTeresis 1403
SBUS<sb>:QSPI:IOONe:POLarity 1404

SBUS<sb>:QSPI:IOONe:POSition ..... 1404
SBUS<sb>:QSPI:IOONe:SCALe ..... 1404
SBUS<sb>:QSPI:IOONe:SOURce ..... 1404
SBUS<sb>:QSPI:IOONe:THReshold ..... 1405
SBUS<sb>:QSPI:IOTHree:HYSTeresis ..... 1405
SBUS<sb>:QSPI:IOTHree:POLarity ..... 1405
SBUS<sb>:QSPI:IOTHree:POSition ..... 1406
SBUS<sb>:QSPI:IOTHree:SCALe ..... 1406
SBUS<sb>:QSPI:IOTHree:SOURce ..... 1406
SBUS<sb>:QSPI:IOTHree:THReshold ..... 1406
SBUS<sb>:QSPI:IOTWo:HYSTeresis ..... 1407
SBUS<sb>:QSPI:IOTWo:POLarity ..... 1407
SBUS<sb>:QSPI:IOTWo:POSition ..... 1407
SBUS<sb>:QSPI:IOTWo:SCALe ..... 1407
SBUS<sb>:QSPI:IOTWo:SOURce ..... 1408
SBUS<sb>:QSPI:IOTWo:THReshold ..... 1408
SBUS<sb>:QSPI:IOZero:HYSTeresis ..... 1408
SBUS<sb>:QSPI:IOZero:POLarity ..... 1408
SBUS<sb>:QSPI:IOZero:POSition ..... 1409
SBUS<sb>:QSPI:IOZero:SCALe ..... 1409
SBUS<sb>:QSPI:IOZero:SOURce ..... 1409
SBUS<sb>:QSPI:IOZero:THReshold ..... 1409
SBUS<sb>:QSPI:LDOPcode ..... 1417
SBUS<sb>:QSPI:OPCode:APPend ..... 1417
SBUS<sb>:QSPI:OPCode:DALL ..... 1417
SBUS<sb>:QSPI:OPCode:DELete ..... 1417
SBUS<sb>:QSPI:OPCode:ITEM<n>:ADBYtes ..... 1418
SBUS<sb>:QSPI:OPCode:ITEM<n>:ADLanes ..... 1418
SBUS<sb>:QSPI:OPCode:ITEM<n>:ALT ..... 1419
SBUS<sb>:QSPI:OPCode:ITEM<n>:CODE ..... 1419
SBUS<sb>:QSPI:OPCode:ITEM<n>:DATA ..... 1419
SBUS<sb>:QSPI:OPCode:ITEM<n>:DDR ..... 1420
SBUS<sb>:QSPI:OPCode:ITEM<n>:DMCYcles ..... 1420
SBUS<sb>:QSPI:OPCode:ITEM<n>:DTLanes ..... 1420
SBUS<sb>:QSPI:OPCode:ITEM<n>:NAME ..... 1421
SBUS<sb>:QSPI:OPCode:RESet ..... 1417
SBUS<sb>:QSPI:OPCode:SIZE? ..... 1418
SBUS<sb>:QSPI:SCLK:HYSTeresis ..... 1410
SBUS<sb>:QSPI:SCLK:POLarity ..... 1410
SBUS<sb>:QSPI:SCLK:SOURce ..... 1410
SBUS<sb>:QSPI:SCLK:THReshold ..... 1411
SBUS<sb>:QSPI:SVOP ..... 1421
SBUS<sb>:QSPI:SWTindex? ..... 1427
SBUS<sb>:QSPI:SWTTime? ..... 1427
SBUS<sb>:RESult ..... 1364
SBUS<sb>:RFFE:CLOCk:HYSTeresis ..... 1888
SBUS<sb>:RFFE:CLOCk:SOURce ..... 1888
SBUS<sb>:RFFE:CLOCk:THReshold ..... 1888
SBUS<sb>:RFFE:DATA:HYSTeresis ..... 1889
SBUS<sb>:RFFE:DATA:SOURce ..... 1889

SBUS<sb>:RFFE:DATA:THReshold....1889
SBUS<sb>:RFFE:FCOunt?....1913
SBUS<sb>:RFFE:FILTer:BIT....1893
SBUS<sb>:RFFE:FILTer:CHKall....1892
SBUS<sb>:RFFE:FILTer:CLR....1892
SBUS<sb>:RFFE:FILTer:DMAX....1893
SBUS<sb>:RFFE:FILTer:DMIN....1894
SBUS<sb>:RFFE:FILTer:DOPerator....1894
SBUS<sb>:RFFE:FILTer:ERENable....1896
SBUS<sb>:RFFE:FILTer:ERRor<n>:ENABle....1896
SBUS<sb>:RFFE:FILTer:FIENable....1895
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:ENABle....1893
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:BIT....1893
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1893
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1894
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1894
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1895
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1895
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1895
SBUS<sb>:RFFE:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1896
SBUS<sb>:RFFE:FILTer:FRENable....1893
SBUS<sb>:RFFE:FILTer:IMAX....1895
SBUS<sb>:RFFE:FILTer:IMIN....1895
SBUS<sb>:RFFE:FILTer:INVert....1892
SBUS<sb>:RFFE:FILTer:IOPerator....1896
SBUS<sb>:RFFE:FILTer:RST....1892
SBUS<sb>:RFFE:FORMat:ADDFrame....1897
SBUS<sb>:RFFE:FORMat:CLR....1898
SBUS<sb>:RFFE:FORMat:FCOunt?....1898
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ADDField....1899
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:COLor....1900
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1904
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1902
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1903
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:COLor....1901
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1905
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1904
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1903
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1902
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:NAME....1901
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:PARA....1905
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:PARB....1905
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1902
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1905
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:FLDCount?....1899
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID1Field....1900
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID1Value....1901
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID2Field....1900
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID2Value....1901
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID3Field....1900

SBUS<sb>:RFFE:FORMat:FRAMe<fr>:ID3Value ..... 1901
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:NAME ..... 1899
SBUS<sb>:RFFE:FORMat:FRAMe<fr>:TYPE ..... 1900
SBUS<sb>:RFFE:FORMat:LOAD ..... 1898
SBUS<sb>:RFFE:FORMat:SAVE ..... 1898
SBUS<sb>:RFFE:FORMat:SELect ..... 1897
SBUS<sb>:RFFE:FRAMe<fr>:ADDRess? ..... 1913
SBUS<sb>:RFFE:FRAMe<fr>:BCOunt? ..... 1914
SBUS<sb>:RFFE:FRAMe<fr>:DATA? ..... 1914
SBUS<sb>:RFFE:FRAMe<fr>:FLD<fl>:FVALue? ..... 1915
SBUS<sb>:RFFE:FRAMe<fr>:FLD<fl>:NAME? ..... 1915
SBUS<sb>:RFFE:FRAMe<fr>:FLD<fl>:STATe? ..... 1915
SBUS<sb>:RFFE:FRAMe<fr>:FLD<fl>:VALue? ..... 1915
SBUS<sb>:RFFE:FRAMe<fr>:FLDCount? ..... 1914
SBUS<sb>:RFFE:FRAMe<fr>:PADone? ..... 1912
SBUS<sb>:RFFE:FRAMe<fr>:PADZero? ..... 1913
SBUS<sb>:RFFE:FRAMe<fr>:PCTRl? ..... 1913
SBUS<sb>:RFFE:FRAMe<fr>:RBTRate? ..... 1916
SBUS<sb>:RFFE:FRAMe<fr>:SADD? ..... 1916
SBUS<sb>:RFFE:FRAMe<fr>:STARt? ..... 1916
SBUS<sb>:RFFE:FRAMe<fr>:STATe? ..... 1917
SBUS<sb>:RFFE:FRAMe<fr>:STOP? ..... 1917
SBUS<sb>:RFFE:FRAMe<fr>:SYMBol? ..... 1919
SBUS<sb>:RFFE:FRAMe<fr>:TYPE? ..... 1917
SBUS<sb>:RFFE:FRAMe<fr>:WBTRate? ..... 1919
SBUS<sb>:RFFE:GFILter ..... 1890
SBUS<sb>:RFFE:GFWidth ..... 1890
SBUS<sb>:RFFE:NEWList ..... 1919
SBUS<sb>:RFFE:POSition ..... 1890
SBUS<sb>:RFFE:RDMD ..... 1890
SBUS<sb>:RFFE:SCALe ..... 1891
SBUS<sb>:RFFE:SWTindex? ..... 1911
SBUS<sb>:RFFE:SWTTime? ..... 1911
SBUS<sb>:RFFE:SYMBols ..... 1919
SBUS<sb>:RMSBus ..... 1364
SBUS<sb>:SENT:CLKPeriod ..... 1736
SBUS<sb>:SENT:CLKTolerance ..... 1737
SBUS<sb>:SENT:CRCMethod ..... 1737
SBUS<sb>:SENT:CRCVersion ..... 1737
SBUS<sb>:SENT:DATA:HYSTeresis ..... 1738
SBUS<sb>:SENT:DATA:SOURce ..... 1738
SBUS<sb>:SENT:DATA:THReshold ..... 1738
SBUS<sb>:SENT:DNIBbles ..... 1739
SBUS<sb>:SENT:FCOunt? ..... 1763
SBUS<sb>:SENT:FILTer:BIT ..... 1744
SBUS<sb>:SENT:FILTer:CHKall ..... 1743
SBUS<sb>:SENT:FILTer:CLR ..... 1743
SBUS<sb>:SENT:FILTer:DMAX ..... 1745
SBUS<sb>:SENT:FILTer:DMIN ..... 1745
SBUS<sb>:SENT:FILTer:DOPerator ..... 1746

SBUS<sb>:SENT:FILTer:ERENable ... 1748
SBUS<sb>:SENT:FILTer:ERRor<n>:ENABle ... 1748
SBUS<sb>:SENT:FILTer:FIENable ... 1746
SBUS<sb>:SENT:FILTer:FRAMe<fr>:ENABle ... 1744
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1744
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1745
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1745
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1746
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1746
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1746
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1747
SBUS<sb>:SENT:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1747
SBUS<sb>:SENT:FILTer:FRENable ... 1744
SBUS<sb>:SENT:FILTer:IMAX ... 1746
SBUS<sb>:SENT:FILTer:IMIN ... 1747
SBUS<sb>:SENT:FILTer:INVert ... 1744
SBUS<sb>:SENT:FILTer:IOPerator ... 1747
SBUS<sb>:SENT:FILTer:RST ... 1744
SBUS<sb>:SENT:FORMat:ADDFrame ... 1749
SBUS<sb>:SENT:FORMat:CLR ... 1750
SBUS<sb>:SENT:FORMat:FCOunt? ... 1750
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ADDField ... 1750
SBUS<sb>:SENT:FORMat:FRAMe<fr>:COLor ... 1751
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ... 1755
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:BITCount ... 1753
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:BITorder ... 1754
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:COLor ... 1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ... 1756
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ... 1756
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FORMat ... 1754
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ... 1753
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:NAME ... 1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:PARA ... 1757
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:PARB ... 1757
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:REPetition ... 1754
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLD<fl>:UNIT ... 1756
SBUS<sb>:SENT:FORMat:FRAMe<fr>:FLDCount? ... 1750
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID1Field ... 1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID1Value ... 1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID2Field ... 1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID2Value ... 1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID3Field ... 1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:ID3Value ... 1752
SBUS<sb>:SENT:FORMat:FRAMe<fr>:NAME ... 1751
SBUS<sb>:SENT:FORMat:FRAMe<fr>:TYPE ... 1751
SBUS<sb>:SENT:FORMat:LOAD ... 1749
SBUS<sb>:SENT:FORMat:SAVE ... 1749
SBUS<sb>:SENT:FORMat:SELect ... 1749
SBUS<sb>:SENT:FRAMe<fr>:BITRate? ... 1764
SBUS<sb>:SENT:FRAMe<fr>:CSValue? ... 1764

SBUS<sb>:SENT:FRAMe<fr>:DATA? 1764
SBUS<sb>:SENT:FRAMe<fr>:FDValues? 1765
SBUS<sb>:SENT:FRAMe<fr>:FLD<fl>:FVALue? 1767
SBUS<sb>:SENT:FRAMe<fr>:FLD<fl>:LABel? 1768
SBUS<sb>:SENT:FRAMe<fr>:FLD<fl>:STATe? 1767
SBUS<sb>:SENT:FRAMe<fr>:FLD<fl>:VALue? 1767
SBUS<sb>:SENT:FRAMe<fr>:FLDCount? 1766
SBUS<sb>:SENT:FRAMe<fr>:FSCom? 1765
SBUS<sb>:SENT:FRAMe<fr>:FSDTa? 1765
SBUS<sb>:SENT:FRAMe<fr>:IDTYpe? 1765
SBUS<sb>:SENT:FRAMe<fr>:IDValue? 1766
SBUS<sb>:SENT:FRAMe<fr>:MTPDuration? 1766
SBUS<sb>:SENT:FRAMe<fr>:PAPTicks? 1768
SBUS<sb>:SENT:FRAMe<fr>:SCOM? 1768
SBUS<sb>:SENT:FRAMe<fr>:SDATa? 1769
SBUS<sb>:SENT:FRAMe<fr>:SENSor? 1769
SBUS<sb>:SENT:FRAMe<fr>:STARt? 1769
SBUS<sb>:SENT:FRAMe<fr>:STATus? 1770
SBUS<sb>:SENT:FRAMe<fr>:STOP? 1770
SBUS<sb>:SENT:FRAMe<fr>:SYMBol? 1771
SBUS<sb>:SENT:FRAMe<fr>:SYNCduration? 1771
SBUS<sb>:SENT:FRAMe<fr>:TYPE? 1771
SBUS<sb>:SENT:MODE 1739
SBUS<sb>:SENT:NEWList 1739
SBUS<sb>:SENT:POSition 1740
SBUS<sb>:SENT:PPFLength 1740
SBUS<sb>:SENT:PPULse 1740
SBUS<sb>:SENT:RDSL 1741
SBUS<sb>:SENT:SCALe 1741
SBUS<sb>:SENT:SFORmat 1741
SBUS<sb>:SENT:SWTindex? 1741
SBUS<sb>:SENT:SWTindex? 1762
SBUS<sb>:SENT:SWTTime? 1742
SBUS<sb>:SENT:SWTTime? 1763
SBUS<sb>:SENT:SYMBols 1742
SBUS<sb>:SPI:BORDer 1368
SBUS<sb>:SPI:CSELect:HYSTeresis 1368
SBUS<sb>:SPI:CSELect:POLarity 1369
SBUS<sb>:SPI:CSELect:SOURce 1369
SBUS<sb>:SPI:CSELect:THReshold 1369
SBUS<sb>:SPI:FCOunt? 1397
SBUS<sb>:SPI:FILTer:BIT 1376
SBUS<sb>:SPI:FILTer:CHKall 1375
SBUS<sb>:SPI:FILTer:CLR 1375
SBUS<sb>:SPI:FILTer:DMAX 1376
SBUS<sb>:SPI:FILTer:DMIN 1377
SBUS<sb>:SPI:FILTer:DOPerator 1377
SBUS<sb>:SPI:FILTer:ERENable 1377
SBUS<sb>:SPI:FILTer:ERRor<n>:ENABle 1377
SBUS<sb>:SPI:FILTer:FIENable 1378

SBUS<sb>:SPI:FILTer:FRAMe<fr>:ENABle ... 1378
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1376
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1376
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1377
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1377
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1378
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1378
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1379
SBUS<sb>:SPI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1379
SBUS<sb>:SPI:FILTer:FRENable ... 1378
SBUS<sb>:SPI:FILTer:IMAX ... 1378
SBUS<sb>:SPI:FILTer:IMIN ... 1379
SBUS<sb>:SPI:FILTer:INVert ... 1375
SBUS<sb>:SPI:FILTer:IOPerator ... 1379
SBUS<sb>:SPI:FILTer:RST ... 1376
SBUS<sb>:SPI:FORMat:ADDFrame ... 1381
SBUS<sb>:SPI:FORMat:CLR ... 1381
SBUS<sb>:SPI:FORMat:FCOunt? ... 1382
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ADDField ... 1382
SBUS<sb>:SPI:FORMat:FRAMe<fr>:COLor ... 1383
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ... 1387
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:BITCount ... 1385
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:BITorder ... 1386
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:COLor ... 1384
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ... 1388
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ... 1387
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FORMat ... 1386
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ... 1385
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:NAME ... 1384
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:PARA ... 1388
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:PARB ... 1388
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:REPetition ... 1385
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLD<fl>:UNIT ... 1388
SBUS<sb>:SPI:FORMat:FRAMe<fr>:FLDCount? ... 1382
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID1Field ... 1383
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID1Value ... 1384
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID2Field ... 1383
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID2Value ... 1384
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID3Field ... 1383
SBUS<sb>:SPI:FORMat:FRAMe<fr>:ID3Value ... 1384
SBUS<sb>:SPI:FORMat:FRAMe<fr>:NAME ... 1382
SBUS<sb>:SPI:FORMat:FRAMe<fr>:TYPE ... 1383
SBUS<sb>:SPI:FORMat:LOAD ... 1381
SBUS<sb>:SPI:FORMat:SAVE ... 1381
SBUS<sb>:SPI:FORMat:SELect ... 1380
SBUS<sb>:SPI:FRAMe<fr>:BITRate? ... 1397
SBUS<sb>:SPI:FRAMe<fr>:DATA? ... 1397
SBUS<sb>:SPI:FRAMe<fr>:STARt? ... 1398
SBUS<sb>:SPI:FRAMe<fr>:STATus? ... 1398
SBUS<sb>:SPI:FRAMe<fr>:STOP? ... 1398

SBUS<sb>:SPI:FRAMe<fr>:WCOunt? 1399
SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:FMISo? 1399
SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:FMOSi? 1399
SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:MISO? 1400
SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:MOSI? 1400
SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:STARt? 1400
SBUS<sb>:SPI:FRAMe<fr>:WORD<w>:STOP? 1401
SBUS<sb>:SPI:FRCondition 1368
SBUS<sb>:SPI:MISO:HYSTeresis 1369
SBUS<sb>:SPI:MISO:POLarity 1370
SBUS<sb>:SPI:MISO:POSition 1373
SBUS<sb>:SPI:MISO:SCALe 1373
SBUS<sb>:SPI:MISO:SOURce 1370
SBUS<sb>:SPI:MISO:THReshold 1370
SBUS<sb>:SPI:MOSI:HYSTeresis 1371
SBUS<sb>:SPI:MOSI:POLarity 1371
SBUS<sb>:SPI:MOSI:POSition 1374
SBUS<sb>:SPI:MOSI:SCALe 1374
SBUS<sb>:SPI:MOSI:SOURce 1371
SBUS<sb>:SPI:MOSI:THReshold 1371
SBUS<sb>:SPI:SCLK:HYSTeresis 1372
SBUS<sb>:SPI:SCLK:SOURce 1372
SBUS<sb>:SPI:SCLK:THReshold 1372
SBUS<sb>:SPI:SWTindex? 1396
SBUS<sb>:SPI:SWTTime? 1396
SBUS<sb>:SPI:TIMeout 1373
SBUS<sb>:SPI:WSIZe 1373
SBUS<sb>:SPMI:FCOunt? 1882
SBUS<sb>:SPMI:FILTer:BIT 1862
SBUS<sb>:SPMI:FILTer:CHKall 1862
SBUS<sb>:SPMI:FILTer:CLR 1862
SBUS<sb>:SPMI:FILTer:DMAX 1863
SBUS<sb>:SPMI:FILTer:DMIN 1863
SBUS<sb>:SPMI:FILTer:DOPerator 1864
SBUS<sb>:SPMI:FILTer:ERENable 1864
SBUS<sb>:SPMI:FILTer:ERRor<n>:ENABle 1864
SBUS<sb>:SPMI:FILTer:FIENable 1866
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:ENABle 1866
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:BIT 1862
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DMAX 1863
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DMIN 1863
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:DOPerator 1864
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:ENABle 1866
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IMAX 1865
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IMIN 1865
SBUS<sb>:SPMI:FILTer:FRAMe<fr>:FLD<fl>:IOPerator 1864
SBUS<sb>:SPMI:FILTer:FRENable 1866
SBUS<sb>:SPMI:FILTer:IMAX 1865
SBUS<sb>:SPMI:FILTer:IMIN 1865
SBUS<sb>:SPMI:FILTer:INVert 1862

SBUS<sb>:SPMI:FILTer:IOPerator....1864
SBUS<sb>:SPMI:FILTer:RST....1862
SBUS<sb>:SPMI:FORMat:ADDFrame....1867
SBUS<sb>:SPMI:FORMat:CLR....1868
SBUS<sb>:SPMI:FORMat:FCOunt?....1868
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ADDField....1869
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:COLor....1869
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1874
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1872
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1872
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:COLor....1871
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1874
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1874
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1873
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1871
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:NAME....1871
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:PARA....1875
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:PARB....1875
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1872
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1875
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:FLDCount?....1869
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID1Field....1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID1Value....1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID2Field....1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID2Value....1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID3Field....1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:ID3Value....1870
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:NAME....1869
SBUS<sb>:SPMI:FORMat:FRAMe<fr>:TYPE....1870
SBUS<sb>:SPMI:FORMat:LOAD....1867
SBUS<sb>:SPMI:FORMat:SAVE....1868
SBUS<sb>:SPMI:FORMat:SELect....1867
SBUS<sb>:SPMI:FRAMe<fr>:ADDRess?....1882
SBUS<sb>:SPMI:FRAMe<fr>:BC?....1882
SBUS<sb>:SPMI:FRAMe<fr>:DATA?....1883
SBUS<sb>:SPMI:FRAMe<fr>:FLD<fl>:FVALue?....1883
SBUS<sb>:SPMI:FRAMe<fr>:FLD<fl>:LABel?....1884
SBUS<sb>:SPMI:FRAMe<fr>:FLD<fl>:STATe?....1884
SBUS<sb>:SPMI:FRAMe<fr>:FLD<fl>:VALue?....1884
SBUS<sb>:SPMI:FRAMe<fr>:FLDCount?....1883
SBUS<sb>:SPMI:FRAMe<fr>:RBRate?....1885
SBUS<sb>:SPMI:FRAMe<fr>:SA?....1885
SBUS<sb>:SPMI:FRAMe<fr>:STARt?....1885
SBUS<sb>:SPMI:FRAMe<fr>:STATus?....1885
SBUS<sb>:SPMI:FRAMe<fr>:STOP?....1886
SBUS<sb>:SPMI:FRAMe<fr>:SYMBol?....1887
SBUS<sb>:SPMI:FRAMe<fr>:TYPE?....1886
SBUS<sb>:SPMI:FRAMe<fr>:WBRate?....1887
SBUS<sb>:SPMI:GIDValue....1857
SBUS<sb>:SPMI:GSIDenable....1857

SBUS<sb>:SPMI:GTCHenable ... 1858
SBUS<sb>:SPMI:GTWDith ... 1858
SBUS<sb>:SPMI:NEWList ... 1861
SBUS<sb>:SPMI:POSition ... 1860
SBUS<sb>:SPMI:SCALe ... 1860
SBUS<sb>:SPMI:SCLK:HYSTeresis ... 1858
SBUS<sb>:SPMI:SCLK:SOURce ... 1858
SBUS<sb>:SPMI:SCLK:THReshold ... 1859
SBUS<sb>:SPMI:SDATa:HYSTeresis ... 1859
SBUS<sb>:SPMI:SDATa:SOURce ... 1859
SBUS<sb>:SPMI:SDATa:THReshold ... 1860
SBUS<sb>:SPMI:SWTindex? ... 1881
SBUS<sb>:SPMI:SWTTime? ... 1881
SBUS<sb>:SPMI:SYMBols ... 1860
SBUS<sb>:SWIRe:DATA:HYSTeresis ... 1830
SBUS<sb>:SWIRe:DATA:SOURce ... 1829
SBUS<sb>:SWIRe:DATA:THReshold ... 1830
SBUS<sb>:SWIRe:FCOunt? ... 1850
SBUS<sb>:SWIRe:FILTer:BIT ... 1833
SBUS<sb>:SWIRe:FILTer:CHKall ... 1832
SBUS<sb>:SWIRe:FILTer:CLR ... 1833
SBUS<sb>:SWIRe:FILTer:DMAX ... 1834
SBUS<sb>:SWIRe:FILTer:DMIN ... 1834
SBUS<sb>:SWIRe:FILTer:DOPerator ... 1835
SBUS<sb>:SWIRe:FILTer:ERENable ... 1834
SBUS<sb>:SWIRe:FILTer:ERRor<n>:ENABle ... 1834
SBUS<sb>:SWIRe:FILTer:FIENable ... 1835
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:ENABle ... 1836
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1833
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1834
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1834
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1835
SBUS<sb>:SWIRe:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1835
SBUS<sb>:SWIRe:FILTer:FRENable ... 1836
SBUS<sb>:SWIRe:FILTer:INVert ... 1833
SBUS<sb>:SWIRe:FILTer:RST ... 1833
SBUS<sb>:SWIRe:FORMat:ADDFrame ... 1837
SBUS<sb>:SWIRe:FORMat:CLR ... 1838
SBUS<sb>:SWIRe:FORMat:FCOunt? ... 1838
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ADDField ... 1838
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:COLor ... 1839
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ... 1843
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:BITCount ... 1841
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:BITorder ... 1842
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:COLor ... 1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ... 1844
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ... 1844
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FORMat ... 1842
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ... 1841
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:NAME ... 1840

SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:PARA ... 1845
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:PARB ... 1845
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:REPetition ... 1842
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLD<fl>:UNIT ... 1844
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:FLDCount? ... 1838
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID1Field ... 1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID1Value ... 1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID2Field ... 1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID2Value ... 1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID3Field ... 1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:ID3Value ... 1840
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:NAME ... 1839
SBUS<sb>:SWIRe:FORMat:FRAMe<fr>:TYPE ... 1839
SBUS<sb>:SWIRe:FORMat:LOAD ... 1837
SBUS<sb>:SWIRe:FORMat:SAVE ... 1837
SBUS<sb>:SWIRe:FORMat:SELect ... 1837
SBUS<sb>:SWIRe:FRAMe<fr>:BITRate? ... 1850
SBUS<sb>:SWIRe:FRAMe<fr>:CODParity? ... 1851
SBUS<sb>:SWIRe:FRAMe<fr>:DATA? ... 1851
SBUS<sb>:SWIRe:FRAMe<fr>:DATParity? ... 1852
SBUS<sb>:SWIRe:FRAMe<fr>:ESCParity? ... 1852
SBUS<sb>:SWIRe:FRAMe<fr>:FLD<fl>:FVALue? ... 1854
SBUS<sb>:SWIRe:FRAMe<fr>:FLD<fl>:STATus? ... 1853
SBUS<sb>:SWIRe:FRAMe<fr>:FLD<fl>:TYPE? ... 1853
SBUS<sb>:SWIRe:FRAMe<fr>:FLD<fl>:VALue? ... 1854
SBUS<sb>:SWIRe:FRAMe<fr>:STARt? ... 1855
SBUS<sb>:SWIRe:FRAMe<fr>:STATe? ... 1854
SBUS<sb>:SWIRe:FRAMe<fr>:STOP? ... 1855
SBUS<sb>:SWIRe:FRAMe<fr>:TYPE? ... 1856
SBUS<sb>:SWIRe:MGAP ... 1831
SBUS<sb>:SWIRe:MINGap ... 1831
SBUS<sb>:SWIRe:POSition ... 1831
SBUS<sb>:SWIRe:SCALe ... 1831
SBUS<sb>:SWIRe:STRBe:HYSTeresis ... 1831
SBUS<sb>:SWIRe:STRBe:SOURce ... 1830
SBUS<sb>:SWIRe:STRBe:THReshold ... 1830
SBUS<sb>:SWIRe:SWTindex? ... 1849
SBUS<sb>:SWIRe:SWTTime? ... 1849
SBUS<sb>:THReshold ... 1364
SBUS<sb>:TNOS:FCOunt? ... 1943
SBUS<sb>:TNOS:FILTer:BIT ... 1924
SBUS<sb>:TNOS:FILTer:CHKall ... 1923
SBUS<sb>:TNOS:FILTer:CLR ... 1923
SBUS<sb>:TNOS:FILTer:DMAX ... 1924
SBUS<sb>:TNOS:FILTer:DMIN ... 1925
SBUS<sb>:TNOS:FILTer:DOPerator ... 1925
SBUS<sb>:TNOS:FILTer:ERENable ... 1925
SBUS<sb>:TNOS:FILTer:ERRor<n>:ENABle ... 1925
SBUS<sb>:TNOS:FILTer:FIENable ... 1926
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:ENABle ... 1927

SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1924
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1924
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1925
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1925
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1926
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1926
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1926
SBUS<sb>:TNOS:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1927
SBUS<sb>:TNOS:FILTer:FRENable ... 1927
SBUS<sb>:TNOS:FILTer:IMAX ... 1926
SBUS<sb>:TNOS:FILTer:IMIN ... 1926
SBUS<sb>:TNOS:FILTer:INVert ... 1924
SBUS<sb>:TNOS:FILTer:IOPerator ... 1927
SBUS<sb>:TNOS:FILTer:RST ... 1923
SBUS<sb>:TNOS:FORMat:ADDFrame ... 1929
SBUS<sb>:TNOS:FORMat:CLR ... 1929
SBUS<sb>:TNOS:FORMat:FCOunt? ... 1930
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ADDField ... 1930
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:COLor ... 1931
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ... 1935
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:BITCount ... 1933
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:BITorder ... 1934
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:COLor ... 1932
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ... 1936
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ... 1935
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FORMat ... 1934
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:FSTBit ... 1933
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:NAME ... 1932
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:PARA ... 1936
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:PARB ... 1936
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:REPetition ... 1933
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLD<fl>:UNIT ... 1936
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:FLDCount? ... 1930
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID1Field ... 1931
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID1Value ... 1932
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID2Field ... 1931
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID2Value ... 1932
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID3Field ... 1931
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:ID3Value ... 1932
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:NAME ... 1930
SBUS<sb>:TNOS:FORMat:FRAMe<fr>:TYPE ... 1931
SBUS<sb>:TNOS:FORMat:LOAD ... 1929
SBUS<sb>:TNOS:FORMat:SAVE ... 1929
SBUS<sb>:TNOS:FORMat:SELect ... 1928
SBUS<sb>:TNOS:FRAMe<fr>:BITRate? ... 1944
SBUS<sb>:TNOS:FRAMe<fr>:CRC? ... 1946
SBUS<sb>:TNOS:FRAMe<fr>:DATA? ... 1946
SBUS<sb>:TNOS:FRAMe<fr>:DTADdress? ... 1946
SBUS<sb>:TNOS:FRAMe<fr>:DTSYmbol? ... 1947
SBUS<sb>:TNOS:FRAMe<fr>:FCRC? ... 1947

SBUS<sb>:TNOS:FRAMe<fr>:FDATa? 1947
SBUS<sb>:TNOS:FRAMe<fr>:FDTaddress? 1947
SBUS<sb>:TNOS:FRAMe<fr>:FLD<fl>:FVALue? 1944
SBUS<sb>:TNOS:FRAMe<fr>:FLD<fl>:LABel? 1944
SBUS<sb>:TNOS:FRAMe<fr>:FLD<fl>:STATe? 1945
SBUS<sb>:TNOS:FRAMe<fr>:FLD<fl>:VALue? 1945
SBUS<sb>:TNOS:FRAMe<fr>:FLDCount? 1943
SBUS<sb>:TNOS:FRAMe<fr>:FSRaddress? 1948
SBUS<sb>:TNOS:FRAMe<fr>:FTPLenth? 1948
SBUS<sb>:TNOS:FRAMe<fr>:SRADdress? 1948
SBUS<sb>:TNOS:FRAMe<fr>:SRSYmbol? 1949
SBUS<sb>:TNOS:FRAMe<fr>:STARt? 1949
SBUS<sb>:TNOS:FRAMe<fr>:STATus? 1949
SBUS<sb>:TNOS:FRAMe<fr>:STOP? 1950
SBUS<sb>:TNOS:FRAMe<fr>:TPLenth? 1950
SBUS<sb>:TNOS:FRAMe<fr>:TYPE? 1951
SBUS<sb>:TNOS:NEWList 1922
SBUS<sb>:TNOS:POSition 1920
SBUS<sb>:TNOS:SCALe 1920
SBUS<sb>:TNOS:SOURce 1921
SBUS<sb>:TNOS:SWTindex? 1942
SBUS<sb>:TNOS:SWTTime? 1942
SBUS<sb>:TNOS:SYMBols 1921
SBUS<sb>:TNOS:THReshold:HYSTeresis 1921
SBUS<sb>:TNOS:THReshold:LOWer 1921
SBUS<sb>:TNOS:THReshold:UPPer 1922
SBUS<sb>:TYPE 1362
SBUS<sb>:UART:BITRate 1533
SBUS<sb>:UART:BORDer 1534
SBUS<sb>:UART:FILTer:BIT 1543
SBUS<sb>:UART:FILTer:CHKall 1539
SBUS<sb>:UART:FILTer:CLR 1540
SBUS<sb>:UART:FILTer:DMAX 1543
SBUS<sb>:UART:FILTer:DMIN 1542
SBUS<sb>:UART:FILTer:DOPerator 1542
SBUS<sb>:UART:FILTer:ERENable 1542
SBUS<sb>:UART:FILTer:ERRor<n>:ENABle 1542
SBUS<sb>:UART:FILTer:FIENable 1544
SBUS<sb>:UART:FILTer:FRAMe<fr>:ENABle 1543
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:BIT 1543
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:DMAX 1543
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:DMIN 1542
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:DOPerator 1542
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:ENABle 1544
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:IMAX 1541
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:IMIN 1540
SBUS<sb>:UART:FILTer:FRAMe<fr>:FLD<fl>:IOPerator 1541
SBUS<sb>:UART:FILTer:FRENable 1543
SBUS<sb>:UART:FILTer:IMAX 1541
SBUS<sb>:UART:FILTer:IMIN 1540

SBUS<sb>:UART:FILTer:INVert....1540
SBUS<sb>:UART:FILTer:IOPerator....1541
SBUS<sb>:UART:FILTer:RST....1540
SBUS<sb>:UART:FORMat:ADDFrame....1545
SBUS<sb>:UART:FORMat:CLR....1546
SBUS<sb>:UART:FORMat:FCOunt?....1546
SBUS<sb>:UART:FORMat:FRAMe<fr>:ADDField....1546
SBUS<sb>:UART:FORMat:FRAMe<fr>:COLor....1547
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1551
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1550
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1550
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:COLor....1549
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1552
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1552
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1551
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1549
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:NAME....1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:PARA....1553
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:PARB....1553
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1550
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1552
SBUS<sb>:UART:FORMat:FRAMe<fr>:FLDCount?....1547
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID1Field....1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID1Value....1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID2Field....1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID2Value....1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID3Field....1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:ID3Value....1548
SBUS<sb>:UART:FORMat:FRAMe<fr>:NAME....1547
SBUS<sb>:UART:FORMat:FRAMe<fr>:TYPE....1548
SBUS<sb>:UART:FORMat:LOAD....1545
SBUS<sb>:UART:FORMat:SAVE....1546
SBUS<sb>:UART:FORMat:SELect....1545
SBUS<sb>:UART:PACKets....1534
SBUS<sb>:UART:PARity....1534
SBUS<sb>:UART:POLarity....1535
SBUS<sb>:UART:RX:HYSTeresis....1535
SBUS<sb>:UART:RX:POSition....1538
SBUS<sb>:UART:RX:SCALe....1538
SBUS<sb>:UART:RX:SOURce....1535
SBUS<sb>:UART:RX:THReshold....1536
SBUS<sb>:UART:SBIT....1536
SBUS<sb>:UART:SSIZe....1536
SBUS<sb>:UART:SWTindex?....1560
SBUS<sb>:UART:SWTTime?....1560
SBUS<sb>:UART:TOUT....1537
SBUS<sb>:UART:TX:HYSTeresis....1537
SBUS<sb>:UART:TX:POSition....1538
SBUS<sb>:UART:TX:SCALe....1538
SBUS<sb>:UART:TX:SOURce....1537

SBUS<sb>:UART:TX:THReshold ... 1538
SBUS<sb>:UART:WCOunt? ... 1561
SBUS<sb>:UART:WORD<w>:BITRate? ... 1562
SBUS<sb>:UART:WORD<w>:FLD<fl>:FVALue? ... 1565
SBUS<sb>:UART:WORD<w>:FLD<fl>:LABel? ... 1565
SBUS<sb>:UART:WORD<w>:FLD<fl>:STATe? ... 1562
SBUS<sb>:UART:WORD<w>:FLD<fl>:VALue? ... 1565
SBUS<sb>:UART:WORD<w>:FLDCount? ... 1561
SBUS<sb>:UART:WORD<w>:RXValue? ... 1562
SBUS<sb>:UART:WORD<w>:SOURce? ... 1563
SBUS<sb>:UART:WORD<w>:STARt? ... 1563
SBUS<sb>:UART:WORD<w>:STATe? ... 1564
SBUS<sb>:UART:WORD<w>:STOP? ... 1564
SBUS<sb>:UART:WORD<w>:TXValue? ... 1563
SBUS<sb>:USBHspeed:FCOunt? ... 2005
SBUS<sb>:USBHspeed:FILTer:BIT ... 1989
SBUS<sb>:USBHspeed:FILTer:CHKall ... 1985
SBUS<sb>:USBHspeed:FILTer:CLR ... 1985
SBUS<sb>:USBHspeed:FILTer:DMAX ... 1986
SBUS<sb>:USBHspeed:FILTer:DMIN ... 1986
SBUS<sb>:USBHspeed:FILTer:DOPerator ... 1987
SBUS<sb>:USBHspeed:FILTer:ERENable ... 1987
SBUS<sb>:USBHspeed:FILTer:ERRor<n>:ENABle ... 1987
SBUS<sb>:USBHspeed:FILTer:FIENable ... 1989
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:ENABle ... 1985
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:BIT ... 1989
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DMAX ... 1986
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DMIN ... 1986
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:DOPerator ... 1987
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:ENABle ... 1989
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IMAX ... 1987
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IMIN ... 1988
SBUS<sb>:USBHspeed:FILTer:FRAMe<fr>:FLD<fl>:IOPerator ... 1988
SBUS<sb>:USBHspeed:FILTer:FRENable ... 1985
SBUS<sb>:USBHspeed:FILTer:IMAX ... 1987
SBUS<sb>:USBHspeed:FILTer:IMIN ... 1988
SBUS<sb>:USBHspeed:FILTer:INVert ... 1985
SBUS<sb>:USBHspeed:FILTer:IOPerator ... 1988
SBUS<sb>:USBHspeed:FILTer:RST ... 1985
SBUS<sb>:USBHspeed:FORMat:ADDFrame ... 1990
SBUS<sb>:USBHspeed:FORMat:CLR ... 1991
SBUS<sb>:USBHspeed:FORMat:FCOunt? ... 1991
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ADDField ... 1992
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:COLor ... 1992
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ADDValue ... 1997
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:BITCount ... 1995
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:BITorder ... 1995
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:COLor ... 1994
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel ... 1997
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue ... 1997

SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1996
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1994
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:NAME....1994
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:PARA....1998
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:PARB....1998
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1995
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1998
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:FLDCount?....1992
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID1Field....1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID1Value....1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID2Field....1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID2Value....1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID3Field....1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:ID3Value....1993
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:NAME....1992
SBUS<sb>:USBHspeed:FORMat:FRAMe<fr>:TYPE....1993
SBUS<sb>:USBHspeed:FORMat:LOAD....1991
SBUS<sb>:USBHspeed:FORMat:SAVE....1991
SBUS<sb>:USBHspeed:FORMat:SELect....1990
SBUS<sb>:USBHspeed:FRAMe<fr>:ADDRess?....2005
SBUS<sb>:USBHspeed:FRAMe<fr>:BITRate?....2006
SBUS<sb>:USBHspeed:FRAMe<fr>:CRC?....2010
SBUS<sb>:USBHspeed:FRAMe<fr>:DATA?....2006
SBUS<sb>:USBHspeed:FRAMe<fr>:ENDPoint?....2011
SBUS<sb>:USBHspeed:FRAMe<fr>:FADDress?....2010
SBUS<sb>:USBHspeed:FRAMe<fr>:FCRC?....2010
SBUS<sb>:USBHspeed:FRAMe<fr>:FLD<fl>:FVALue?....2008
SBUS<sb>:USBHspeed:FRAMe<fr>:FLD<fl>:LABel?....2008
SBUS<sb>:USBHspeed:FRAMe<fr>:FLD<fl>:STATe?....2009
SBUS<sb>:USBHspeed:FRAMe<fr>:FLD<fl>:VALue?....2009
SBUS<sb>:USBHspeed:FRAMe<fr>:FLDCount?....2006
SBUS<sb>:USBHspeed:FRAMe<fr>:FRNumber?....2011
SBUS<sb>:USBHspeed:FRAMe<fr>:PID?....2011
SBUS<sb>:USBHspeed:FRAMe<fr>:PIDName?....2011
SBUS<sb>:USBHspeed:FRAMe<fr>:SPET?....2012
SBUS<sb>:USBHspeed:FRAMe<fr>:SPEU?....2013
SBUS<sb>:USBHspeed:FRAMe<fr>:SPPort?....2013
SBUS<sb>:USBHspeed:FRAMe<fr>:SPS?....2013
SBUS<sb>:USBHspeed:FRAMe<fr>:SPSC?....2014
SBUS<sb>:USBHspeed:FRAMe<fr>:STARt?....2007
SBUS<sb>:USBHspeed:FRAMe<fr>:STATus?....2007
SBUS<sb>:USBHspeed:FRAMe<fr>:STOP?....2008
SBUS<sb>:USBHspeed:FRAMe<fr>:TYPE?....2012
SBUS<sb>:USBHspeed:POSition....1983
SBUS<sb>:USBHspeed:SCALe....1984
SBUS<sb>:USBHspeed:SOURce....1982
SBUS<sb>:USBHspeed:SWTindex?....2004
SBUS<sb>:USBHspeed:SWTTime?....2004
SBUS<sb>:USBHspeed:THReshold:HYSTeresis....1983
SBUS<sb>:USBHspeed:THReshold:LOWer....1983

SBUS<sb>:USBHspeed:THReshold:UPPer....1983
SBUS<sb>:USBone:DMINus:HYSTeresis....1953
SBUS<sb>:USBone:DMINus:SOURce....1952
SBUS<sb>:USBone:DMINus:THReshold....1953
SBUS<sb>:USBone:DPLus:HYSTeresis....1953
SBUS<sb>:USBone:DPLus:SOURce....1951
SBUS<sb>:USBone:DPLus:THReshold....1952
SBUS<sb>:USBone:FCOunt?....1975
SBUS<sb>:USBone:FILTer:BIT....1959
SBUS<sb>:USBone:FILTer:CHKall....1955
SBUS<sb>:USBone:FILTer:CLR....1955
SBUS<sb>:USBone:FILTer:DMAX....1956
SBUS<sb>:USBone:FILTer:DMIN....1956
SBUS<sb>:USBone:FILTer:DOPerator....1957
SBUS<sb>:USBone:FILTer:ERENable....1957
SBUS<sb>:USBone:FILTer:ERRor<n>:ENABle....1957
SBUS<sb>:USBone:FILTer:FIENable....1959
SBUS<sb>:USBone:FILTer:FRAMe<fr>:ENABle....1955
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:BIT....1959
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DMAX....1956
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DMIN....1956
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:DOPerator....1957
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:ENABle....1959
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:IMAX....1957
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:IMIN....1958
SBUS<sb>:USBone:FILTer:FRAMe<fr>:FLD<fl>:IOPerator....1958
SBUS<sb>:USBone:FILTer:FRENable....1955
SBUS<sb>:USBone:FILTer:IMAX....1957
SBUS<sb>:USBone:FILTer:IMIN....1958
SBUS<sb>:USBone:FILTer:INVert....1955
SBUS<sb>:USBone:FILTer:IOPerator....1958
SBUS<sb>:USBone:FILTer:RST....1955
SBUS<sb>:USBone:FORMat:ADDFrame....1960
SBUS<sb>:USBone:FORMat:CLR....1961
SBUS<sb>:USBone:FORMat:FCOunt?....1961
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ADDField....1962
SBUS<sb>:USBone:FORMat:FRAMe<fr>:COLor....1962
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ADDValue....1967
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:BITCount....1965
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:BITorder....1965
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:COLor....1964
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:LABel....1967
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:ENUM<p>:VALue....1967
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FORMat....1966
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:FSTBit....1964
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:NAME....1964
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:PARA....1968
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:PARB....1968
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:REPetition....1965
SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLD<fl>:UNIT....1968

SBUS<sb>:USBone:FORMat:FRAMe<fr>:FLDCount? 1962
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID1Field 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID1Value 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID2Field 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID2Value 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID3Field 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:ID3Value 1963
SBUS<sb>:USBone:FORMat:FRAMe<fr>:NAME 1962
SBUS<sb>:USBone:FORMat:FRAMe<fr>:TYPE 1963
SBUS<sb>:USBone:FORMat:LOAD 1961
SBUS<sb>:USBone:FORMat:SAVE 1961
SBUS<sb>:USBone:FORMat:SELect 1960
SBUS<sb>:USBone:FRAMe<fr>:ADDRess? 1975
SBUS<sb>:USBone:FRAMe<fr>:BITRate? 1976
SBUS<sb>:USBone:FRAMe<fr>:CRC? 1979
SBUS<sb>:USBone:FRAMe<fr>:DATA? 1976
SBUS<sb>:USBone:FRAMe<fr>:ENDPoint? 1980
SBUS<sb>:USBone:FRAMe<fr>:FADDress? 1980
SBUS<sb>:USBone:FRAMe<fr>:FCRC? 1980
SBUS<sb>:USBone:FRAMe<fr>:FLD<fl>:FVALue? 1978
SBUS<sb>:USBone:FRAMe<fr>:FLD<fl>:LABel? 1978
SBUS<sb>:USBone:FRAMe<fr>:FLD<fl>:STATe? 1978
SBUS<sb>:USBone:FRAMe<fr>:FLD<fl>:VALue? 1979
SBUS<sb>:USBone:FRAMe<fr>:FLDCount? 1975
SBUS<sb>:USBone:FRAMe<fr>:FRNumber? 1981
SBUS<sb>:USBone:FRAMe<fr>:PID? 1981
SBUS<sb>:USBone:FRAMe<fr>:PIDName? 1981
SBUS<sb>:USBone:FRAMe<fr>:STARt? 1976
SBUS<sb>:USBone:FRAMe<fr>:STATus? 1977
SBUS<sb>:USBone:FRAMe<fr>:STOP? 1977
SBUS<sb>:USBone:FRAMe<fr>:TYPE? 1981
SBUS<sb>:USBone:MODE 1952
SBUS<sb>:USBone:POSition 1953
SBUS<sb>:USBone:SCALe 1954
SBUS<sb>:USBone:SWTindex? 1974
SBUS<sb>:USBone:SWTTime? 1974
SBUS<sb>:ZCOupling 1365
SBUS<sb>[:STATe] 1362
SERVice:REPort 989
SESSion:LOAD[:EXECute] 1134
SESSion:SAVE:ABORt 1134
SESSion:SAVE:CHANnel 1134
SESSion:SAVE:REFerence 1134
SESSion:SAVE[:EXECute] 1135
SESSion:USERpref 1135
SINGle 990
STATus:OPERation:CONDition? 2046
STATus:OPERation:ENABle 2047
STATus:OPERation:NTRansition 2047
STATus:OPERation:PTRansition 2048

STATus:OPERation[:EVENt]?....2047
STATus:PRESet....2046
STATus:QUEStionable:ADCState:CONDition?....2050
STATus:QUEStionable:ADCState:ENABle....2051
STATus:QUEStionable:ADCState:NTRansition....2052
STATus:QUEStionable:ADCState:PTRansition....2053
STATus:QUEStionable:ADCState[:EVENt]?....2050
STATus:QUEStionable:COVerload:CONDition?....2050
STATus:QUEStionable:COVerload:ENABle....2051
STATus:QUEStionable:COVerload:NTRansition....2052
STATus:QUEStionable:COVerload:PTRansition....2053
STATus:QUEStionable:COVerload[:EVENt]?....2050
STATus:QUEStionable:GOVerload:CONDition?....2050
STATus:QUEStionable:GOVerload:ENABle....2051
STATus:QUEStionable:GOVerload:NTRansition....2052
STATus:QUEStionable:GOVerload:PTRansition....2053
STATus:QUEStionable:GOVerload[:EVENt]?....2051
STATus:QUEStionable:IMPRecise:CONDition?....2050
STATus:QUEStionable:IMPRecise:ENABle....2051
STATus:QUEStionable:IMPRecise:NTRansition....2052
STATus:QUEStionable:IMPRecise:PTRansition....2053
STATus:QUEStionable:IMPRecise[:EVENt]?....2051
STATus:QUEStionable:MASK:CONDition?....2050
STATus:QUEStionable:MASK:ENABle....2051
STATus:QUEStionable:MASK:NTRansition....2052
STATus:QUEStionable:MASK:PTRansition....2052
STATus:QUEStionable:MASK[:EVENt]?....2050
STATus:QUEStionable:PLL:CONDition?....2050
STATus:QUEStionable:PLL:ENABle....2051
STATus:QUEStionable:PLL:NTRansition....2052
STATus:QUEStionable:PLL:PTRansition....2053
STATus:QUEStionable:PLL:RCONdition....2053
STATus:QUEStionable:PLL[:EVENt]?....2051
STATus:QUEStionable:PPSupply:CONDition?....2050
STATus:QUEStionable:PPSupply:ENABle....2051
STATus:QUEStionable:PPSupply:NTRansition....2052
STATus:QUEStionable:PPSupply:PTRansition....2053
STATus:QUEStionable:PPSupply[:EVENt]?....2051
STATus:QUEStionable:TEMPerature:CONDition?....2050
STATus:QUEStionable:TEMPerature:ENABle....2051
STATus:QUEStionable:TEMPerature:NTRansition....2052
STATus:QUEStionable:TEMPerature:PTRansition....2053
STATus:QUEStionable:TEMPerature[:EVENt]?....2051
STOP....990
SYSTem:APUP....989
SYSTem:COMMunicate:NET[:HOSTname]....953
SYSTem:DATE....953
SYSTem:DISPlay:MESSage:STATe....952
SYSTem:DISPlay:MESSage[:TEXT]....952
SYSTem:DISPlay:UPDate....952

SYSTem:EXIT ..... 954
SYSTem:FW:FILepath ..... 954
SYSTem:FW:STARt ..... 955
SYSTem:PRESet ..... 954
SYSTem:SHUTdown ..... 954
SYSTem:TIME ..... 955
TIMebase:DIVisions? ..... 991
TIMebase:HORizontal:POSition ..... 991
TIMebase:RANGe ..... 991
TIMebase:REFerence ..... 992
TIMebase:ROLL:ENABle ..... 992
TIMebase:ROLL:MTIMe ..... 992
TIMebase:ROLL:STATe? ..... 993
TIMebase:SCALe ..... 990
TREFerence<m>:CDR:SOFTware:ALGorithm ..... 1081
TREFerence<m>:CDR:SOFTware:BWIDth ..... 1082
TREFerence<m>:CDR:SOFTware:CFRequency:CRSYnc ..... 1082
TREFerence<m>:CDR:SOFTware:PLL:DAMPing ..... 1082
TREFerence<m>:CDR:SOFTware:PLL:ORDer ..... 1083
TREFerence<m>:CDR:SOFTware:PLL:SYNC ..... 1083
TREFerence<m>:CDR:SOFTware:RELBwidth ..... 1084
TREFerence<m>:CDR:SOFTware:SELResults ..... 1084
TREFerence<m>:CDR:SOFTware:SREStimate ..... 1083
TREFerence<m>:CLK:FACTor ..... 1084
TREFerence<m>:CLK:FREQuency ..... 1085
TREFerence<m>:CLK:OFFSet ..... 1085
TREFerence<m>:CLK:REFLevel ..... 1085
TREFerence<m>:EDGE ..... 1086
TREFerence<m>:ENABle ..... 1086
TREFerence<m>:GATE ..... 1086
TREFerence<m>:REFLevel ..... 1087
TREFerence<m>:RFLSet ..... 1087
TREFerence<m>:SOURce ..... 1087
TREFerence<m>:STATe? ..... 1087
TREFerence<m>:SYMRate ..... 1088
TREFerence<m>:TYPE ..... 1088
TRFS:COUNt? ..... 1088
TRIGger:ACTions:BEEP ..... 1063
TRIGger:ACTions:OUT:DELay ..... 1063
TRIGger:ACTions:OUT:OVERload:STATus:CONDition? ..... 2050
TRIGger:ACTions:OUT:OVERload:STATus:ENABle ..... 2051
TRIGger:ACTions:OUT:OVERload:STATus:NTRansition ..... 2052
TRIGger:ACTions:OUT:OVERload:STATus:PTRansition ..... 2052
TRIGger:ACTions:OUT:OVERload:STATus:PTRansition ..... 2052
TRIGger:ACTions:OUT:OVERload:STATus[:EVENt] ..... 2050
TRIGger:ACTions:OUT:PLENgth ..... 1064
TRIGger:ACTions:OUT:POLarity ..... 1064
TRIGger:ACTions:OUT:SOURce ..... 1063
TRIGger:ACTions:OUT:STATe ..... 1063
TRIGger:ACTions:SCReenshot ..... 1064

TRIGger:ACTions:SRESults....1064
TRIGger:ACTions:STOP....1065
TRIGger:ACTions:WFMSave....1065
TRIGger:ANEDge:COUPling....1036
TRIGger:ANEDge:CUToff:HIGHpass....1036
TRIGger:ANEDge:CUToff:LOWPass....1037
TRIGger:ANEDge:FILTer....1036
TRIGger:ANEDge:LEVel....1036
TRIGger:ANEDge:NREJect....1037
TRIGger:ANEDge:OVERload:STATus:CONDition?....2050
TRIGger:ANEDge:OVERload:STATus:ENABle....2051
TRIGger:ANEDge:OVERload:STATus:NTRansition....2052
TRIGger:ANEDge:OVERload:STATus:PTRansition....2052
TRIGger:ANEDge:OVERload:STATus[:EVENt]....2050
TRIGger:EVENt<ev>:EDGE:SLOPe....1035
TRIGger:EVENt<ev>:GLITch:POLarity....1037
TRIGger:EVENt<ev>:GLITch:RANGe....1038
TRIGger:EVENt<ev>:GLITch:WIDTh....1038
TRIGger:EVENt<ev>:INTerval:DELTa....1046
TRIGger:EVENt<ev>:INTerval:RANGe....1047
TRIGger:EVENt<ev>:INTerval:SLOPe....1047
TRIGger:EVENt<ev>:INTerval:WIDTh....1047
TRIGger:EVENt<ev>:LEVel<n>:RUNT:LOWer....1040
TRIGger:EVENt<ev>:LEVel<n>:RUNT:UPPer....1040
TRIGger:EVENt<ev>:LEVel<n>:SLEW:LOWer....1048
TRIGger:EVENt<ev>:LEVel<n>:SLEW:UPPer....1048
TRIGger:EVENt<ev>:LEVel<n>:WINDow:LOWer....1043
TRIGger:EVENt<ev>:LEVel<n>:WINDow:UPPer....1043
TRIGger:EVENt<ev>:LEVel<n>[:VALue]....1032
TRIGger:EVENt<ev>:PATTern:QUALify:ANALog:CHAN<n>:HLX....1054
TRIGger:EVENt<ev>:PATTern:QUALify:DIGital:CHAN<n>:HLX....1054
TRIGger:EVENt<ev>:PATTern:QUALify:DIGital:LOGic....1055
TRIGger:EVENt<ev>:PATTern:QUALify:LOGic....1055
TRIGger:EVENt<ev>:PATTern:QUALify:SOURces....1055
TRIGger:EVENt<ev>:RUNT:DELTa....1041
TRIGger:EVENt<ev>:RUNT:POLarity....1041
TRIGger:EVENt<ev>:RUNT:RANGe....1041
TRIGger:EVENt<ev>:RUNT:WIDTh....1042
TRIGger:EVENt<ev>:SETHold:CSOurce:EDGE....1050
TRIGger:EVENt<ev>:SETHold:CSOurce:LEVel....1051
TRIGger:EVENt<ev>:SETHold:CSOurce[:VALue]....1051
TRIGger:EVENt<ev>:SETHold:HTIMe....1051
TRIGger:EVENt<ev>:SETHold:STIMe....1052
TRIGger:EVENt<ev>:SLEW:DELTa....1049
TRIGger:EVENt<ev>:SLEW:RANGe....1049
TRIGger:EVENt<ev>:SLEW:RATE....1050
TRIGger:EVENt<ev>:SLEW:SLOPe....1050
TRIGger:EVENt<ev>:SOURce....1032
TRIGger:EVENt<ev>:STATe:QUALify:ANALog:CHAN<n>:HLX....1052
TRIGger:EVENt<ev>:STATe:QUALify:DIGital:CHAN<n>:HLX....1054

TRIGger:EVENt<ev>:STATe:QUALify:DIGital:LOGic ... 1053
TRIGger:EVENt<ev>:STATe:QUALify:LOGic ... 1053
TRIGger:EVENt<ev>:STATe:SLOPe ... 1053
TRIGger:EVENt<ev>:TIMeout:RANGe ... 1045
TRIGger:EVENt<ev>:TIMeout:TIME ... 1046
TRIGger:EVENt<ev>:TYPE ... 1031
TRIGger:EVENt<ev>:WIDTh:DELTa ... 1039
TRIGger:EVENt<ev>:WIDTh:POLarity ... 1039
TRIGger:EVENt<ev>:WIDTh:RANGe ... 1039
TRIGger:EVENt<ev>:WIDTh:WIDTh ... 1039
TRIGger:EVENt<ev>:WINDow:DELTa ... 1043
TRIGger:EVENt<ev>:WINDow:RANGe ... 1044
TRIGger:EVENt<ev>:WINDow:TIME ... 1044
TRIGger:EVENt<ev>:WINDow:WIDTh ... 1045
TRIGger:FILTermode ... 1061
TRIGger:FINDlevel ... 1032
TRIGger:FORCe ... 1056
TRIGger:HOLDoff:AUTotime? ... 1057
TRIGger:HOLDoff:EVENts ... 1058
TRIGger:HOLDoff:MAX ... 1058
TRIGger:HOLDoff:MIN ... 1059
TRIGger:HOLDoff:MODE ... 1057
TRIGger:HOLDoff:SCALing ... 1058
TRIGger:HOLDoff:TIME ... 1059
TRIGger:LFReject ... 1062
TRIGger:MEVents:AEVents ... 1033
TRIGger:MEVents:MODE ... 1031
TRIGger:MEVents:SEQuence<se>:COUNt ... 1033
TRIGger:MEVents:SEQuence<se>:DELay ... 1034
TRIGger:MEVents:SEQuence<se>:RESet:EVENt ... 1034
TRIGger:MEVents:SEQuence<se>:RESet:TIMeout:TIME ... 1034
TRIGger:MEVents:SEQuence<se>:RESet:TIMeout[:ENABle] ... 1035
TRIGger:MODE ... 1056
TRIGger:NOISe<m>:ABSolute ... 1059
TRIGger:NOISe<m>:EFFective? ... 1060
TRIGger:NOISe<m>:MODE ... 1060
TRIGger:NOISe<m>:PERDivision ... 1060
TRIGger:NOISe<m>:RELative ... 1061
TRIGger:NOISe<m>[:STATe] ... 1061
TRIGger:RFReject ... 1062
TRIGger:SBHW:CAN:ACKerror ... 1676
TRIGger:SBHW:CAN:BITSterror ... 1676
TRIGger:SBHW:CAN:BORDer ... 1676
TRIGger:SBHW:CAN:CRCerror ... 1677
TRIGger:SBHW:CAN:DCONdition ... 1677
TRIGger:SBHW:CAN:DLC ... 1677
TRIGger:SBHW:CAN:DLCCondition ... 1677
TRIGger:SBHW:CAN:DMIN ... 1678
TRIGger:SBHW:CAN:FDATa:BRS ... 1678
TRIGger:SBHW:CAN:FDATa:DPOSition ... 1678

TRIGger:SBHW:CAN:FDATa:ESI ... 1678
TRIGger:SBHW:CAN:FDATa:SCERror ... 1679
TRIGger:SBHW:CAN:FORMerror ... 1679
TRIGger:SBHW:CAN:FTYPe ... 1679
TRIGger:SBHW:CAN:ICONdition ... 1680
TRIGger:SBHW:CAN:IMAX ... 1680
TRIGger:SBHW:CAN:IMIN ... 1680
TRIGger:SBHW:CAN:ITYPe ... 1681
TRIGger:SBHW:CAN:TYPE ... 1681
TRIGger:SBHW:CAN:XDATa:AF:CONDition ... 1682
TRIGger:SBHW:CAN:XDATa:AF:MAX ... 1682
TRIGger:SBHW:CAN:XDATa:AF:MIN ... 1682
TRIGger:SBHW:CAN:XDATa:SDT:CONDition ... 1683
TRIGger:SBHW:CAN:XDATa:SDT:MAX ... 1683
TRIGger:SBHW:CAN:XDATa:SDT:MIN ... 1683
TRIGger:SBHW:CAN:XDATa:SEC ... 1684
TRIGger:SBHW:CAN:XDATa:VCID:CONDition ... 1684
TRIGger:SBHW:CAN:XDATa:VCID:MAX ... 1684
TRIGger:SBHW:CAN:XDATa:VCID:MIN ... 1684
TRIGger:SBHW:I2C:ACCess ... 1483
TRIGger:SBHW:I2C:ACONdition ... 1483
TRIGger:SBHW:I2C:ADDRess ... 1483
TRIGger:SBHW:I2C:ADDTo ... 1484
TRIGger:SBHW:I2C:ADNack ... 1484
TRIGger:SBHW:I2C:AMODe ... 1484
TRIGger:SBHW:I2C:DCONdition ... 1484
TRIGger:SBHW:I2C:DMIN ... 1485
TRIGger:SBHW:I2C:DPOSition ... 1485
TRIGger:SBHW:I2C:DRNack ... 1485
TRIGger:SBHW:I2C:DWNack ... 1485
TRIGger:SBHW:I2C:TYPE ... 1482
TRIGger:SBHW:LIN:CHKSerror ... 1724
TRIGger:SBHW:LIN:DCONdition ... 1724
TRIGger:SBHW:LIN:DMIN ... 1723
TRIGger:SBHW:LIN:DPOSition ... 1723
TRIGger:SBHW:LIN:ICONdition ... 1723
TRIGger:SBHW:LIN:IMAX ... 1723
TRIGger:SBHW:LIN:IMIN ... 1722
TRIGger:SBHW:LIN:IPERror ... 1722
TRIGger:SBHW:LIN:SYERror ... 1722
TRIGger:SBHW:LIN:TYPE ... 1721
TRIGger:SBHW:SPI:DMINpattern ... 1390
TRIGger:SBHW:SPI:DPOSition ... 1390
TRIGger:SBHW:SPI:FCONdition ... 1390
TRIGger:SBHW:SPI:PALignment ... 1390
TRIGger:SBHW:SPI:TYPE ... 1389
TRIGger:SBHW:UART:DATA ... 1553
TRIGger:SBHW:UART:DPOSition ... 1554
TRIGger:SBHW:UART:FCONdition ... 1554
TRIGger:SBHW:UART:OPERator ... 1554

TRIGger:SBHW:UART:SOURce....1554
TRIGger:SBHW:UART:TYPE....1555
TRIGger:SBSW:ARINc:BIT....1792
TRIGger:SBSW:ARINc:CHKall....1791
TRIGger:SBSW:ARINc:CLR....1791
TRIGger:SBSW:ARINc:DMAX....1792
TRIGger:SBSW:ARINc:DMIN....1792
TRIGger:SBSW:ARINc:DOPerator....1793
TRIGger:SBSW:ARINc:ERENable....1794
TRIGger:SBSW:ARINc:ERRor<m>:ENABle....1794
TRIGger:SBSW:ARINc:FIENable....1793
TRIGger:SBSW:ARINc:FRAMe<fr>:ENABle....1791
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:BIT....1792
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DMAX....1792
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DMIN....1792
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:DOPerator....1793
TRIGger:SBSW:ARINc:FRAMe<fr>:FLD<fl>:ENABle....1793
TRIGger:SBSW:ARINc:FRENable....1791
TRIGger:SBSW:ARINc:INVert....1791
TRIGger:SBSW:ARINc:RST....1791
TRIGger:SBSW:CAN:BIT....1686
TRIGger:SBSW:CAN:CHKall....1685
TRIGger:SBSW:CAN:CLR....1686
TRIGger:SBSW:CAN:DMAX....1687
TRIGger:SBSW:CAN:DMIN....1687
TRIGger:SBSW:CAN:DOPerator....1688
TRIGger:SBSW:CAN:ERENable....1689
TRIGger:SBSW:CAN:ERRor<m>:ENABle....1689
TRIGger:SBSW:CAN:FIENable....1688
TRIGger:SBSW:CAN:FRAMe<fr>:ENABle....1686
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:BIT....1686
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DMAX....1687
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DMIN....1687
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:DOPerator....1688
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:ENABle....1688
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IMAX....1688
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IMIN....1689
TRIGger:SBSW:CAN:FRAMe<fr>:FLD<fl>:IOPerator....1689
TRIGger:SBSW:CAN:FRENable....1686
TRIGger:SBSW:CAN:IMAX....1688
TRIGger:SBSW:CAN:IMIN....1689
TRIGger:SBSW:CAN:INVert....1686
TRIGger:SBSW:CAN:IOPerator....1689
TRIGger:SBSW:CAN:RST....1686
TRIGger:SBSW:I2C:BIT....1487
TRIGger:SBSW:I2C:CHKall....1486
TRIGger:SBSW:I2C:CLR....1487
TRIGger:SBSW:I2C:DMAX....1488
TRIGger:SBSW:I2C:DMIN....1488
TRIGger:SBSW:I2C:DOPerator....1489

TRIGger:SBSW:I2C:ERENable ... 1490
TRIGger:SBSW:I2C:ERRor<m>:ENABle ... 1490
TRIGger:SBSW:I2C:FIENable ... 1489
TRIGger:SBSW:I2C:FRAMe<fr>:ENABle ... 1487
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:BIT ... 1487
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DMAX ... 1488
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DMIN ... 1488
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:DOPerator ... 1489
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:ENABle ... 1489
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IMAX ... 1489
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IMIN ... 1490
TRIGger:SBSW:I2C:FRAMe<fr>:FLD<fl>:IOPerator ... 1490
TRIGger:SBSW:I2C:FRENable ... 1487
TRIGger:SBSW:I2C:IMAX ... 1489
TRIGger:SBSW:I2C:IMIN ... 1490
TRIGger:SBSW:I2C:INVert ... 1487
TRIGger:SBSW:I2C:IOPerator ... 1490
TRIGger:SBSW:I2C:RST ... 1487
TRIGger:SBSW:I3C:BIT ... 1520
TRIGger:SBSW:I3C:CHKall ... 1519
TRIGger:SBSW:I3C:CLR ... 1519
TRIGger:SBSW:I3C:DMAX ... 1520
TRIGger:SBSW:I3C:DMIN ... 1521
TRIGger:SBSW:I3C:DOPerator ... 1521
TRIGger:SBSW:I3C:ERENable ... 1523
TRIGger:SBSW:I3C:ERRor<m>:ENABle ... 1523
TRIGger:SBSW:I3C:FIENable ... 1522
TRIGger:SBSW:I3C:FRAMe<fr>:ENABle ... 1520
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:BIT ... 1520
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DMAX ... 1520
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DMIN ... 1521
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:DOPerator ... 1521
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:ENABle ... 1522
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IMAX ... 1522
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IMIN ... 1522
TRIGger:SBSW:I3C:FRAMe<fr>:FLD<fl>:IOPerator ... 1523
TRIGger:SBSW:I3C:FRENable ... 1520
TRIGger:SBSW:I3C:IMAX ... 1522
TRIGger:SBSW:I3C:IMIN ... 1522
TRIGger:SBSW:I3C:INVert ... 1519
TRIGger:SBSW:I3C:IOPerator ... 1523
TRIGger:SBSW:I3C:RST ... 1520
TRIGger:SBSW:LIN:BIT ... 1726
TRIGger:SBSW:LIN:CHKall ... 1725
TRIGger:SBSW:LIN:CLR ... 1725
TRIGger:SBSW:LIN:DMAX ... 1726
TRIGger:SBSW:LIN:DMIN ... 1727
TRIGger:SBSW:LIN:DOPerator ... 1727
TRIGger:SBSW:LIN:ERENable ... 1729
TRIGger:SBSW:LIN:ERRor<m>:ENABle ... 1729

TRIGger:SBSW:LIN:FIENable....1727
TRIGger:SBSW:LIN:FRAMe<fr>:ENABle....1726
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:BIT....1726
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DMAX....1726
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DMIN....1727
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:DOPerator....1727
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:ENABle....1727
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IMAX....1728
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IMIN....1728
TRIGger:SBSW:LIN:FRAMe<fr>:FLD<fl>:IOPerator....1729
TRIGger:SBSW:LIN:FRENable....1726
TRIGger:SBSW:LIN:IMAX....1728
TRIGger:SBSW:LIN:IMIN....1728
TRIGger:SBSW:LIN:INVert....1725
TRIGger:SBSW:LIN:IOPerator....1729
TRIGger:SBSW:LIN:RST....1725
TRIGger:SBSW:MANCh:BIT....1646
TRIGger:SBSW:MANCh:CHKall....1645
TRIGger:SBSW:MANCh:CLR....1645
TRIGger:SBSW:MANCh:DMAX....1646
TRIGger:SBSW:MANCh:DMIN....1646
TRIGger:SBSW:MANCh:DOPerator....1647
TRIGger:SBSW:MANCh:ERENable....1649
TRIGger:SBSW:MANCh:ERRor<m>:ENABle....1649
TRIGger:SBSW:MANCh:FIENable....1647
TRIGger:SBSW:MANCh:FRAMe<fr>:ENABle....1645
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:BIT....1646
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DMAX....1646
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DMIN....1646
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:DOPerator....1647
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:ENABle....1647
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IMAX....1648
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IMIN....1648
TRIGger:SBSW:MANCh:FRAMe<fr>:FLD<fl>:IOPerator....1648
TRIGger:SBSW:MANCh:FRENable....1645
TRIGger:SBSW:MANCh:IMAX....1648
TRIGger:SBSW:MANCh:IMIN....1648
TRIGger:SBSW:MANCh:INVert....1645
TRIGger:SBSW:MANCh:IOPerator....1648
TRIGger:SBSW:MANCh:RST....1645
TRIGger:SBSW:MILStd:BIT....1819
TRIGger:SBSW:MILStd:CHKall....1818
TRIGger:SBSW:MILStd:CLR....1818
TRIGger:SBSW:MILStd:DMAX....1820
TRIGger:SBSW:MILStd:DMIN....1820
TRIGger:SBSW:MILStd:DOPerator....1820
TRIGger:SBSW:MILStd:ERENable....1822
TRIGger:SBSW:MILStd:ERRor<m>:ENABle....1822
TRIGger:SBSW:MILStd:FIENable....1821
TRIGger:SBSW:MILStd:FRAMe<fr>:ENABle....1819

TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:BIT ... 1819
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DMAX ... 1820
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DMIN ... 1820
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:DOPerator ... 1820
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:ENABle ... 1821
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IMAX ... 1821
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IMIN ... 1822
TRIGger:SBSW:MILStd:FRAMe<fr>:FLD<fl>:IOPerator ... 1822
TRIGger:SBSW:MILStd:FRENable ... 1819
TRIGger:SBSW:MILStd:IMAX ... 1821
TRIGger:SBSW:MILStd:IMIN ... 1822
TRIGger:SBSW:MILStd:INVert ... 1819
TRIGger:SBSW:MILStd:IOPerator ... 1822
TRIGger:SBSW:MILStd:RST ... 1819
TRIGger:SBSW:NRZC:BIT ... 1586
TRIGger:SBSW:NRZC:CHKall ... 1585
TRIGger:SBSW:NRZC:CLR ... 1586
TRIGger:SBSW:NRZC:DMAX ... 1587
TRIGger:SBSW:NRZC:DMIN ... 1587
TRIGger:SBSW:NRZC:DOPerator ... 1588
TRIGger:SBSW:NRZC:ERENable ... 1589
TRIGger:SBSW:NRZC:ERRor<m>:ENABle ... 1589
TRIGger:SBSW:NRZC:FIENable ... 1588
TRIGger:SBSW:NRZC:FRAMe<fr>:ENABle ... 1586
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:BIT ... 1586
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DMAX ... 1587
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DMIN ... 1587
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:DOPerator ... 1588
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:ENABle ... 1588
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IMAX ... 1588
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IMIN ... 1589
TRIGger:SBSW:NRZC:FRAMe<fr>:FLD<fl>:IOPerator ... 1589
TRIGger:SBSW:NRZC:FRENable ... 1586
TRIGger:SBSW:NRZC:IMAX ... 1588
TRIGger:SBSW:NRZC:IMIN ... 1589
TRIGger:SBSW:NRZC:INVert ... 1586
TRIGger:SBSW:NRZC:IOPerator ... 1589
TRIGger:SBSW:NRZC:RST ... 1586
TRIGger:SBSW:NRZU:BIT ... 1616
TRIGger:SBSW:NRZU:CHKall ... 1615
TRIGger:SBSW:NRZU:CLR ... 1615
TRIGger:SBSW:NRZU:DMAX ... 1616
TRIGger:SBSW:NRZU:DMIN ... 1616
TRIGger:SBSW:NRZU:DOPerator ... 1617
TRIGger:SBSW:NRZU:ERENable ... 1619
TRIGger:SBSW:NRZU:ERRor<m>:ENABle ... 1619
TRIGger:SBSW:NRZU:FIENable ... 1617
TRIGger:SBSW:NRZU:FRAMe<fr>:ENABle ... 1615
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:BIT ... 1616
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DMAX ... 1616

TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DMIN ... 1616
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:DOPerator ... 1617
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:ENABle ... 1617
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IMAX ... 1618
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IMIN ... 1618
TRIGger:SBSW:NRZU:FRAMe<fr>:FLD<fl>:IOPerator ... 1618
TRIGger:SBSW:NRZU:FRENable ... 1615
TRIGger:SBSW:NRZU:IMAX ... 1618
TRIGger:SBSW:NRZU:IMIN ... 1618
TRIGger:SBSW:NRZU:INVert ... 1615
TRIGger:SBSW:NRZU:IOPerator ... 1618
TRIGger:SBSW:NRZU:RST ... 1615
TRIGger:SBSW:OSPI:BIT ... 1453
TRIGger:SBSW:OSPI:CHKall ... 1452
TRIGger:SBSW:OSPI:CLR ... 1452
TRIGger:SBSW:OSPI:DMAX ... 1453
TRIGger:SBSW:OSPI:DMIN ... 1454
TRIGger:SBSW:OSPI:DOPerator ... 1454
TRIGger:SBSW:OSPI:ERENable ... 1456
TRIGger:SBSW:OSPI:ERRor<m>:ENABle ... 1456
TRIGger:SBSW:OSPI:FIENable ... 1454
TRIGger:SBSW:OSPI:FRAMe<fr>:ENABle ... 1453
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:BIT ... 1453
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:DMAX ... 1453
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:DMIN ... 1454
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:DOPerator ... 1454
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:ENABle ... 1454
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:IMAX ... 1455
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:IMIN ... 1455
TRIGger:SBSW:OSPI:FRAMe<fr>:FLD<fl>:IOPerator ... 1456
TRIGger:SBSW:OSPI:FRENable ... 1453
TRIGger:SBSW:OSPI:IMAX ... 1455
TRIGger:SBSW:OSPI:IMIN ... 1455
TRIGger:SBSW:OSPI:INVert ... 1452
TRIGger:SBSW:OSPI:IOPerator ... 1456
TRIGger:SBSW:OSPI:RST ... 1452
TRIGger:SBSW:QSPI:BIT ... 1423
TRIGger:SBSW:QSPI:CHKall ... 1422
TRIGger:SBSW:QSPI:CLR ... 1422
TRIGger:SBSW:QSPI:DMAX ... 1424
TRIGger:SBSW:QSPI:DMIN ... 1424
TRIGger:SBSW:QSPI:DOPerator ... 1424
TRIGger:SBSW:QSPI:ERENable ... 1426
TRIGger:SBSW:QSPI:ERRor<m>:ENABle ... 1426
TRIGger:SBSW:QSPI:FIENable ... 1425
TRIGger:SBSW:QSPI:FRAMe<fr>:ENABle ... 1423
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:BIT ... 1423
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DMAX ... 1424
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DMIN ... 1424
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:DOPerator ... 1424

TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:ENABle....1425
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IMAX....1425
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IMIN....1426
TRIGger:SBSW:QSPI:FRAMe<fr>:FLD<fl>:IOPerator....1426
TRIGger:SBSW:QSPI:FRENable....1423
TRIGger:SBSW:QSPI:IMAX....1425
TRIGger:SBSW:QSPI:IMIN....1426
TRIGger:SBSW:QSPI:INVert....1422
TRIGger:SBSW:QSPI:IOPerator....1426
TRIGger:SBSW:QSPI:RST....1423
TRIGger:SBSW:RFFE:BIT....1908
TRIGger:SBSW:RFFE:CHKall....1907
TRIGger:SBSW:RFFE:CLR....1907
TRIGger:SBSW:RFFE:DMAX....1908
TRIGger:SBSW:RFFE:DMIN....1908
TRIGger:SBSW:RFFE:DOPerator....1909
TRIGger:SBSW:RFFE:ERENable....1911
TRIGger:SBSW:RFFE:ERRor<m>:ENABle....1911
TRIGger:SBSW:RFFE:FIENable....1909
TRIGger:SBSW:RFFE:FRAMe<fr>:ENABle....1907
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:BIT....1908
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DMAX....1908
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DMIN....1908
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:DOPerator....1909
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:ENABle....1909
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IMAX....1910
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IMIN....1910
TRIGger:SBSW:RFFE:FRAMe<fr>:FLD<fl>:IOPerator....1910
TRIGger:SBSW:RFFE:FRENable....1907
TRIGger:SBSW:RFFE:IMAX....1910
TRIGger:SBSW:RFFE:IMIN....1910
TRIGger:SBSW:RFFE:INVert....1907
TRIGger:SBSW:RFFE:IOPerator....1910
TRIGger:SBSW:RFFE:RST....1907
TRIGger:SBSW:SENT:BIT....1759
TRIGger:SBSW:SENT:CHKall....1758
TRIGger:SBSW:SENT:CLR....1758
TRIGger:SBSW:SENT:DMAX....1760
TRIGger:SBSW:SENT:DMIN....1760
TRIGger:SBSW:SENT:DOPerator....1760
TRIGger:SBSW:SENT:ERENable....1762
TRIGger:SBSW:SENT:ERRor<m>:ENABle....1762
TRIGger:SBSW:SENT:FIENable....1759
TRIGger:SBSW:SENT:FRAMe<fr>:ENABle....1759
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:BIT....1759
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DMAX....1760
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DMIN....1760
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:DOPerator....1760
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:ENABle....1759
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IMAX....1761

TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IMIN ..... 1761
TRIGger:SBSW:SENT:FRAMe<fr>:FLD<fl>:IOPerator ..... 1762
TRIGger:SBSW:SENT:FRENable ..... 1759
TRIGger:SBSW:SENT:IMAX ..... 1761
TRIGger:SBSW:SENT:IMIN ..... 1761
TRIGger:SBSW:SENT:INVert ..... 1758
TRIGger:SBSW:SENT:IOPerator ..... 1762
TRIGger:SBSW:SENT:RST ..... 1758
TRIGger:SBSW:SPI:BIT ..... 1392
TRIGger:SBSW:SPI:CHKall ..... 1391
TRIGger:SBSW:SPI:CLR ..... 1392
TRIGger:SBSW:SPI:DMAX ..... 1393
TRIGger:SBSW:SPI:DMIN ..... 1393
TRIGger:SBSW:SPI:DOPerator ..... 1394
TRIGger:SBSW:SPI:ERENable ..... 1395
TRIGger:SBSW:SPI:ERRor<m>:ENABle ..... 1395
TRIGger:SBSW:SPI:FIENable ..... 1394
TRIGger:SBSW:SPI:FRAMe<fr>:ENABle ..... 1392
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:BIT ..... 1392
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DMAX ..... 1393
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DMIN ..... 1393
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:DOPerator ..... 1394
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:ENABle ..... 1394
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IMAX ..... 1394
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IMIN ..... 1395
TRIGger:SBSW:SPI:FRAMe<fr>:FLD<fl>:IOPerator ..... 1395
TRIGger:SBSW:SPI:FRENable ..... 1392
TRIGger:SBSW:SPI:IMAX ..... 1394
TRIGger:SBSW:SPI:IMIN ..... 1395
TRIGger:SBSW:SPI:INVert ..... 1392
TRIGger:SBSW:SPI:IOPerator ..... 1395
TRIGger:SBSW:SPI:RST ..... 1392
TRIGger:SBSW:SPMI:BIT ..... 1877
TRIGger:SBSW:SPMI:CHKall ..... 1876
TRIGger:SBSW:SPMI:CLR ..... 1877
TRIGger:SBSW:SPMI:DMAX ..... 1878
TRIGger:SBSW:SPMI:DMIN ..... 1878
TRIGger:SBSW:SPMI:DOPerator ..... 1879
TRIGger:SBSW:SPMI:ERENable ..... 1880
TRIGger:SBSW:SPMI:ERRor<m>:ENABle ..... 1880
TRIGger:SBSW:SPMI:FIENable ..... 1879
TRIGger:SBSW:SPMI:FRAMe<fr>:ENABle ..... 1877
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:BIT ..... 1877
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DMAX ..... 1878
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DMIN ..... 1878
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:DOPerator ..... 1879
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:ENABle ..... 1879
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IMAX ..... 1879
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IMIN ..... 1880
TRIGger:SBSW:SPMI:FRAMe<fr>:FLD<fl>:IOPerator ..... 1880

TRIGger:SBSW:SPMI:FRENable ... 1877
TRIGger:SBSW:SPMI:IMAX ... 1879
TRIGger:SBSW:SPMI:IMIN ... 1880
TRIGger:SBSW:SPMI:INVert ... 1877
TRIGger:SBSW:SPMI:IOPerator ... 1880
TRIGger:SBSW:SPMI:RST ... 1877
TRIGger:SBSW:SWIRe:BIT ... 1847
TRIGger:SBSW:SWIRe:CHKall ... 1846
TRIGger:SBSW:SWIRe:CLR ... 1846
TRIGger:SBSW:SWIRe:DMAX ... 1847
TRIGger:SBSW:SWIRe:DMIN ... 1848
TRIGger:SBSW:SWIRe:DOPerator ... 1848
TRIGger:SBSW:SWIRe:ERENable ... 1849
TRIGger:SBSW:SWIRe:ERRor<m>:ENABle ... 1849
TRIGger:SBSW:SWIRe:FIENable ... 1848
TRIGger:SBSW:SWIRe:FRAMe<fr>:ENABle ... 1847
TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:BIT ... 1847
TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DMAX ... 1847
TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DMIN ... 1848
TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:DOPerator ... 1848
TRIGger:SBSW:SWIRe:FRAMe<fr>:FLD<fl>:ENABle ... 1848
TRIGger:SBSW:SWIRe:FRENable ... 1847
TRIGger:SBSW:SWIRe:INVert ... 1846
TRIGger:SBSW:SWIRe:RST ... 1846
TRIGger:SBSW:TNOS:BIT ... 1939
TRIGger:SBSW:TNOS:CHKall ... 1938
TRIGger:SBSW:TNOS:CLR ... 1938
TRIGger:SBSW:TNOS:DMAX ... 1939
TRIGger:SBSW:TNOS:DMIN ... 1939
TRIGger:SBSW:TNOS:DOPerator ... 1940
TRIGger:SBSW:TNOS:ERENable ... 1942
TRIGger:SBSW:TNOS:ERRor<m>:ENABle ... 1942
TRIGger:SBSW:TNOS:FIENable ... 1940
TRIGger:SBSW:TNOS:FRAMe<fr>:ENABle ... 1938
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:BIT ... 1939
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DMAX ... 1939
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DMIN ... 1939
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:DOPerator ... 1940
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:ENABle ... 1940
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IMAX ... 1941
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IMIN ... 1941
TRIGger:SBSW:TNOS:FRAMe<fr>:FLD<fl>:IOPerator ... 1941
TRIGger:SBSW:TNOS:FRENable ... 1938
TRIGger:SBSW:TNOS:IMAX ... 1941
TRIGger:SBSW:TNOS:IMIN ... 1941
TRIGger:SBSW:TNOS:INVert ... 1938
TRIGger:SBSW:TNOS:IOPerator ... 1941
TRIGger:SBSW:TNOS:RST ... 1938
TRIGger:SBSW:UART:BIT ... 1557
TRIGger:SBSW:UART:CHKall ... 1556

TRIGger:SBSW:UART:CLR....1556
TRIGger:SBSW:UART:DMAX....1557
TRIGger:SBSW:UART:DMIN....1557
TRIGger:SBSW:UART:DOPerator....1558
TRIGger:SBSW:UART:ERENable....1560
TRIGger:SBSW:UART:ERRor<m>:ENABle....1560
TRIGger:SBSW:UART:FIENable....1558
TRIGger:SBSW:UART:FRAMe<fr>:ENABle....1556
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:BIT....1557
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DMAX....1557
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DMIN....1557
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:DOPerator....1558
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:ENABle....1558
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IMAX....1559
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IMIN....1559
TRIGger:SBSW:UART:FRAMe<fr>:FLD<fl>:IOPerator....1559
TRIGger:SBSW:UART:FRENable....1556
TRIGger:SBSW:UART:IMAX....1559
TRIGger:SBSW:UART:IMIN....1559
TRIGger:SBSW:UART:INVert....1556
TRIGger:SBSW:UART:IOPerator....1559
TRIGger:SBSW:UART:RST....1556
TRIGger:SBSW:USBHspeed:BIT....2001
TRIGger:SBSW:USBHspeed:CHKall....2000
TRIGger:SBSW:USBHspeed:CLR....2000
TRIGger:SBSW:USBHspeed:DMAX....2001
TRIGger:SBSW:USBHspeed:DMIN....2001
TRIGger:SBSW:USBHspeed:DOPerator....2002
TRIGger:SBSW:USBHspeed:ERENable....2004
TRIGger:SBSW:USBHspeed:ERRor<m>:ENABle....2004
TRIGger:SBSW:USBHspeed:FIENable....2002
TRIGger:SBSW:USBHspeed:FRAMe<fr>:ENABle....2000
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:BIT....2001
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DMAX....2001
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DMIN....2001
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:DOPerator....2002
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:ENABle....2002
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:IMAX....2002
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:IMIN....2003
TRIGger:SBSW:USBHspeed:FRAMe<fr>:FLD<fl>:IOPerator....2003
TRIGger:SBSW:USBHspeed:FRENable....2000
TRIGger:SBSW:USBHspeed:IMAX....2002
TRIGger:SBSW:USBHspeed:IMIN....2003
TRIGger:SBSW:USBHspeed:INVert....2000
TRIGger:SBSW:USBHspeed:IOPerator....2003
TRIGger:SBSW:USBHspeed:RST....2000
TRIGger:SBSW:USBone:BIT....1970
TRIGger:SBSW:USBone:CHKall....1969
TRIGger:SBSW:USBone:CLR....1970
TRIGger:SBSW:USBone:DMAX....1971

TRIGger:SBSW:USBone:DMIN....1971
TRIGger:SBSW:USBone:DOPerator....1972
TRIGger:SBSW:USBone:ERENable....1973
TRIGger:SBSW:USBone:ERRor<m>:ENABle....1973
TRIGger:SBSW:USBone:FIENable....1972
TRIGger:SBSW:USBone:FRAMe<fr>:ENABle....1970
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:BIT....1970
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DMAX....1971
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DMIN....1971
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:DOPerator....1972
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:ENABle....1972
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IMAX....1972
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IMIN....1973
TRIGger:SBSW:USBone:FRAMe<fr>:FLD<fl>:IOPerator....1973
TRIGger:SBSW:USBone:FRENable....1970
TRIGger:SBSW:USBone:IMAX....1972
TRIGger:SBSW:USBone:IMIN....1973
TRIGger:SBSW:USBone:INVert....1970
TRIGger:SBSW:USBone:IOPerator....1973
TRIGger:SBSW:USBone:RST....1970
TRIGger:ZONE:EXPRession[:DEFine]....1066
TRIGger:ZONE:HISTory....1071
TRPRobe:SETup:ATTenuation:UNIT?....1027
TRPRobe:SETup:ATTenuation[:AUTO]?....1027
TRPRobe:SETup:NAME?....1027
TRPRobe:SETup:STATe?....1028
TRPRobe:SETup:TYPE?....1028
USERdefined:PRESet:NAME....987
USERdefined:PRESet:OPEN....987
USERdefined:PRESet:SAVE....987
USERdefined:PRESet[:ENABle]....987
WGENerator<wg>:ARBGen:NAME....2041
WGENerator<wg>:ARBGen:OPEN....2041
WGENerator<wg>:ARBGen:RUNMode....2041
WGENerator<wg>:ARBGen:RUNSingle....2042
WGENerator<wg>:ARBGen:SAMPles?....2042
WGENerator<wg>:ARBGen:SRATe....2042
WGENerator<wg>:ARBGen[:SOURce]....2043
WGENerator<wg>:FREQuency....2028
WGENerator<wg>:FUNCtion:PULSe[:WIDTh]....2028
WGENerator<wg>:FUNCtion:RAMP[:SYMMetry]....2028
WGENerator<wg>:FUNCtion[:SELect]....2029
WGENerator<wg>:FUNCtion[:SQUare]:DCYCle....2029
WGENerator<wg>:GOVerload:STATus:CONDition?....2050
WGENerator<wg>:GOVerload:STATus:ENABle....2051
WGENerator<wg>:GOVerload:STATus:NTRansition....2052
WGENerator<wg>:GOVerload:STATus:PTRansition....2052
WGENerator<wg>:GOVerload:STATus[:EVENt]....2050
WGENerator<wg>:MODulation:AM:CAMPlitude....2030
WGENerator<wg>:MODulation:AM:DCYCle....2034

WGENerator<wg>:MODulation:AM:DEPTh....2034
WGENerator<wg>:MODulation:AM:FREQuency....2034
WGENerator<wg>:MODulation:AM:SYMMetry....2035
WGENerator<wg>:MODulation:AM[:FUNCtion]....2035
WGENerator<wg>:MODulation:CARRier:FREQuency....2035
WGENerator<wg>:MODulation:CARRier:PERiod....2035
WGENerator<wg>:MODulation:FM:DCYCle....2036
WGENerator<wg>:MODulation:FM:DEViation....2036
WGENerator<wg>:MODulation:FM:FREQuency....2036
WGENerator<wg>:MODulation:FM:SYMMetry....2037
WGENerator<wg>:MODulation:FM[:FUNCtion]....2037
WGENerator<wg>:MODulation:FSK:FONE....2037
WGENerator<wg>:MODulation:FSK:FTWO....2037
WGENerator<wg>:MODulation:FSK[:RATE]....2038
WGENerator<wg>:MODulation:NDCLevel....2038
WGENerator<wg>:MODulation:NLPCent....2038
WGENerator<wg>:MODulation:PWM:DCYCle....2039
WGENerator<wg>:MODulation:PWM:DEPTh....2039
WGENerator<wg>:MODulation:PWM:FREQuency....2039
WGENerator<wg>:MODulation:PWM:SYMMetry....2040
WGENerator<wg>:MODulation:PWM[:FUNCtion]....2040
WGENerator<wg>:MODulation:TYPE....2040
WGENerator<wg>:MODulation[:STATe]....2033
WGENerator<wg>:OUTPut[:LOAD]....2029
WGENerator<wg>:PERiod....2030
WGENerator<wg>:PRESet....2030
WGENerator<wg>:SOURce....2031
WGENerator<wg>:SWEep:FEND....2044
WGENerator<wg>:SWEep:FSTart....2044
WGENerator<wg>:SWEep:TIME....2044
WGENerator<wg>:SWEep:TYPE....2043
WGENerator<wg>:SWEep[:STATe]....2043
WGENerator<wg>:TEMPerature:STATus:CONDition?....2050
WGENerator<wg>:TEMPerature:STATus:ENABle....2051
WGENerator<wg>:TEMPerature:STATus:NTRansition....2052
WGENerator<wg>:TEMPerature:STATus:PTRansition....2052
WGENerator<wg>:TEMPerature:STATus[:EVENt]....2050
WGENerator<wg>:VOLTage:DCLevel....2031
WGENerator<wg>:VOLTage:INVersion....2032
WGENerator<wg>:VOLTage:OFFSet....2032
WGENerator<wg>:VOLTage[:VPP]....2032
WGENerator<wg>[:ENABle]....2032
XY<m>:SWAP....1360
XY<m>:XSOurce....1361
XY<m>:YSOurce....1360
XY<m>[:STATe]....1360
ZONE<m>:ACOMbination....1066
ZONE<m>:ADD....1066
ZONE<m>:AREA<n>:ADD....1067
ZONE<m>:AREA<n>:COUNt?....1067

ZONE<m>:AREA<n>:INTersect....1067
ZONE<m>:AREA<n>:LABel....1068
ZONE<m>:AREA<n>:POINt<o>:ADD....1068
ZONE<m>:AREA<n>:POINt<o>:COUNt?....1069
ZONE<m>:AREA<n>:POINt<o>:REMove....1069
ZONE<m>:AREA<n>:POINt<o>:VALid?....1070
ZONE<m>:AREA<n>:POINt<o>:X....1069
ZONE<m>:AREA<n>:POINt<o>:Y....1069
ZONE<m>:AREA<n>:REMove....1070
ZONE<m>:AREA<n>:STATe....1070
ZONE<m>:AREA<n>:VALid?....1068
ZONE<m>:COUNt?....1071
ZONE<m>:DIAGram....1071
ZONE<m>:REMove....1072
ZONE<m>:SOURce....1072
ZONE<m>[:VISible]....1072