Digital Storage Oscilloscope

GDS-2000E Series

PROGRAMMING MANUAL

GW INSTEK PART NO. Version 1.03





December 2018

This manual contains proprietary information which is protected by copyright. All rights are reserved. No part of this manual may be photocopied, reproduced or translated to another language without prior written consent of Good Will Corporation.

The information in this manual was correct at the time of printing. However, Good Will continues to improve products and reserves the right to change specifications, equipment, and maintenance procedures at any time without notice.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Good Will Instrument Co., Ltd. No. 7-1, Jhongsing Rd., Tucheng Dist., New Taipei City 236, Taiwan.

Table of Contents

| INTERFACE O | VERVIEW | 5 |
|-------------|--------------------------------------|-----|
| | Interface Configuration | 5 |
| COMMAND O | VERVIEW | 15 |
| | Command Syntax | |
| | List of Commands in Functional Order | |
| COMMAND D | ETAILS | 30 |
| | Common Commands | 32 |
| | Acquisition Commands | |
| | Autoscale Commands | |
| | Vertical Commands | 47 |
| | Math Commands | 53 |
| | Cursor Commands | 62 |
| | Display Commands | 73 |
| | Hardcopy Commands | 77 |
| | Measure Commands | |
| | Measurement Commands | 105 |
| | Reference Commands | 112 |
| | Run Command | 115 |
| | Stop Command | |
| | Single Command | |
| | Force Command | |
| | Timebase Commands | |
| | Trigger Commands | |
| | System Commands | |
| | Save/Recall Commands | |
| | Ethernet Commands | |
| | Time Commands | |
| | Bus Decode Commands | |
| | Mark Commands | 180 |



| | Search Commands | . 182 |
|---------|-----------------------|-------|
| | Label Commands | . 215 |
| | Segment Commands | . 221 |
| | DVM Commands | . 230 |
| | Go_NoGo Commands | . 232 |
| | Data Logging Commands | . 238 |
| | Remote Disk Commands | . 241 |
| APPENDX | | 244 |
| | Error messages | . 244 |
| INDEX | | 248 |

NTERFACE OVERVIEW

This manual describes how to use the GDS-2000E's remote command functionality and lists the command details. The Overview chapter describes how to configure the GDS-2000E USB and Ethernet remote control interface.

Interface Configuration

Configure USB Interface

USB Configuration PC side connector Type A, host

GDS-2000E side Type B, device

connector

Speed 1.1/2.0

USB Class CDC (communications device

class)

Panel Operation

1. Press the Utility key.

Utility

2. Press I/O from the bottom menu.

1/0

3. Press *USB Device Port* from the side menu and select *Computer*.



4. Connect the USB cable to the rear panel device port.



5. When the PC asks for the USB driver, select the USB driver included on the accompanying User Manual CD or download the driver from the GW Instek website, www.gwinstek.com, in the GDS-2000E Download section. The driver automatically sets the GDS-2000E as a serial COM port (Shown as VPO in the PORTS node).

USB Functionality Check

Terminal Application

Invoke a terminal application such as RealTerm.

Set the COM port, baud rate, stop bit, data bit, and parity accordingly.

To check the COM port number and associated port settings, see the Device Manager in the PC. For Windows 7:

Control panel \rightarrow Hardware and Sound \rightarrow Device Manager

Example: Configuring RealTerm:



Functionality Check

Key in this query command via the terminal application.

*idn?

This should return the Manufacturer, Model number, Serial number, and Firmware version in the following format.

GW,GDS-2202E,PXXXXXX,V1.00



Configure the Ethernet Interface

| Ethernet | MAC Address | Domain Name | |
|-----------------|---|--|----------------------|
| Configuration | Instrument Name | DNS IP Address | |
| | User Password | Gateway IP Address | |
| | Instrument IP | Subnet Mask | |
| | Address | HTTP Port 80 (fixed) | |
| Background | using a socket ser | rface is used for remover connection. For o ket Server section on | letails, |
| Panel Operation | Connect the Et LAN port on the LAN port of the LAN port o | hernet cable to the he rear panel. | LAN |
| | 2. Press the <i>Utilit</i> | y key. | Utility |
| | 3. Press I/O from | the bottom menu. | 1/0 |
| | 4. Press Ethernet | from the side menu. | Ethernet |
| | 5. Set <i>DHCP/BOO</i> from the side r | | DHCP/BOOTP On Off |
| Note | | utomatically be assign to on. For Static IP Ad ould be set to off. | |



```
MAC Address:
                        00:08:21:21:72:73
Instrument Name:
User Password:
                        dso
Instrument IP Address:
Domain Name:
DNS IP Address:
Gateway IP Address:
                        172.16.0.254
Subnet Mask:
                        255.255.0.0
HTTP Port:
 ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 .0123456789-_
1. Use the variable knob to select a character.
2. Press Select to enter the character.
```

6. Use the *Up* and *Down* arrows on the side menu to navigate to each Ethernet configuration item.



Items

MAC Address, Instrument Name, User Password, Instrument IP Address, Domain Name, DNS IP Address, Gateway IP Address, Subnet Mask

Note: HTTP Port is fixed at 80.

7. Use the *Variable* knob to highlight a character and use the *Select* key to choose a character.





Press *Backspace* to delete a character.



Press Save Now to save the configuration. Complete will be displayed when successful.



Configure Socket Server

The GDS-2000E supports socket server functionality for direct twoway communication with a client PC or device over LAN. By default, the Socket Server is off.

Server

Configure Socket 1. Configure the IP address for the GDS-2000E.

Page 7

2. Press the *Utility* key.

Utility

3. Press *I/O* from the bottom menu.

1/0

4. Press Socket Server from the side menu.



5. Press Select Port and choose the port number with the Variable knob.



Range 1024~65535

6. Press Set Port to confirm the port number.



7. The Current Port icon will update to the new port number.





8. Press *Server* and turn the socket server On.



Socket Server Functionality Check

| NI Measurement |
|----------------|
| and Automation |
| Explorer |

To test the socket server functionality, National Instruments Measurement and Automation Explorer can be used. This program is available on the NI website, www.ni.com.

Operation

1. Configure the IP address for the GDS-2000E.

Page 7

- 2. Configure the socket port.
- Page 9
- 3. Start the NI Measurement and Automation Explorer (MAX) program. Using Windows, press:



Start>All Programs>National
Instruments>Measurement & Automation

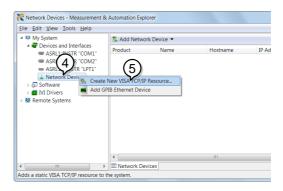


4. From the Configuration panel access;

My System>Devices and Interfaces>Network Devices



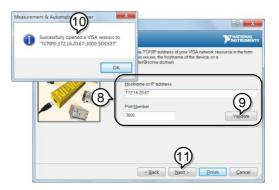
5. Right click *Network Devices* and select *Create New Visa TCP/IP Resource...*



- 6. Select *Manual Entry of Raw Socket* from the popup window.
- 7. Click Next.



- 8. Enter the GDS-2000E's IP address and socket port number.
- 9. Click Validate.
- 10. A popup will appear to tell you if a VISA socket session was successfully created.
- 11. Click Next.



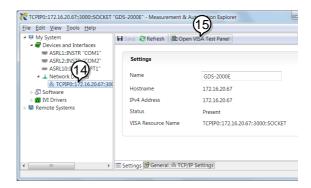
- 12. Choose an alias for the socket connection if you like.
- 13. Click Finish to finish the configuration.



14. The GDS-2000E will now appear under Network Devices in the Configuration Panel.

Functionality Check

15. Click the *Open Visa Test Panel* to send a remote command to the GDS-2000E.



- 16. Click on the Configuration icon.
- 17. Select the I/O Settings tab.
- 18. Mark the *Enable Termination Character* checkbox. Make sure the termination character is a line feed (/n, value: xA).
- 19. Click Apply Changes.



- 20. Click the *Input/Output* icon.
- 21. Make sure *IDN? query is selected in the *Select or Enter Command* drop box.
- 22. Click on Query.
- 23. The manufacturer, model number, serial number and firmware version will be displayed in the buffer. For example:

 GW,GDS-2202E,PXXXXXX,V1.00



COMMAND OVERVIEW

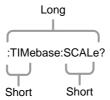
The Command overview chapter lists all GDS-2000E commands in functional order as well as alphabetical order. The command syntax section shows you the basic syntax rules you have to apply when using commands.

Command Syntax

Compatible standard

- USB CDC_ACM compatible
- SCPI, 1994 (partially compatible)

Command forms Commands and queries have two different forms, long and short. The command syntax is written with the short form of the command in capitals and the remainder (long form) in lower case.



The commands can be written in capitals or lowercase, just so long as the short or long forms are complete. An incomplete command will not be recognized.

Below are examples of correctly written commands.

LONG :TIMebase:SCALe? :TIMEBASE:SCALE?

:timebase:scale?



| | SHORT :TIM:SC | AL? :TIM:SO | CAL? |
|-----------------------|---------------------|---|-----------------|
| Command format | :TIMebase:SCAL | e <nr3>LF 1: comm 2: single 2 3 4 3: param 4: messa</nr3> | space |
| Parameter | Туре | Description | Example |
| | <boolean></boolean> | boolean logic | 0, 1 |
| | <nr1></nr1> | Integers | 0, 1, 2, 3 |
| | <nr2></nr2> | floating point | 0.1, 3.14, 8.5 |
| | <nr3></nr3> | floating point with an exponent | 4.5e-1, 8.25e+1 |
| | <nrf></nrf> | any of NR1, 2, 3 | 1, 1.5, 4.5e-1 |
| Message terminator | LF | line feed code | |
| Note | Commands are | non-case sensitive. | |



List of Commands in Functional Order

| Common | *IDN? | 32 |
|-------------|---------------------------------|----|
| | *LRN? | 32 |
| | *SAV | 33 |
| | *RCL | 33 |
| | *RST | 34 |
| | *CLS | 34 |
| | *ESE | 34 |
| | *ESR | 35 |
| | *OPC | 35 |
| | *SRE | 36 |
| | *STB | 36 |
| | ACO. i.e. AVED. e. | 20 |
| Acquisition | :ACQuire:AVERage | |
| | :ACQuire:MODe | |
| | :ACQuire <x>:MEMory?</x> | |
| | :ACQuire:FILTer:SOURce | |
| | :ACQuire:FILTer | |
| | :ACQuire:FILTer:FREQuency | |
| | :ACQuire:FILTer:FREQuency:UPPER | |
| | :ACQuire:FILTer:FREQuency:LOWER | |
| | :ACQuire:FILTer:TYPe | |
| | :ACQuire:FILTer:TRACking | |
| | :ACQuire <x>:STATe?</x> | 44 |
| | :ACQuire:INTERpolation | 44 |
| | :ACQuire:RECOrdlength | 45 |
| | :HEADer | 45 |
| Autoscale | :AUTOSet | 46 |
| | :AUTORSET:MODe | |
| | | |



| Vertical Scale | :CHANnel <x>:BWLimit</x> | 47 |
|----------------|-------------------------------|----|
| | :CHANnel <x>:COUPling</x> | 48 |
| | :CHANnel <x>:DESKew</x> | 48 |
| | :CHANnel <x>:DISPlay</x> | 48 |
| | :CHANnel <x>:EXPand</x> | 49 |
| | :CHANnel <x>:IMPedance?</x> | 49 |
| | :CHANnel <x>:INVert</x> | 50 |
| | :CHANnel <x>:POSition</x> | 50 |
| | :CHANnel <x>:PROBe:RATio</x> | 51 |
| | :CHANnel <x>:PROBe:TYPe</x> | 51 |
| | :CHANnel <x>:SCALe</x> | 51 |
| Math | :MATH:DISP | 53 |
| | :MATH:TYPe | 54 |
| | :MATH:DUAL:SOURce <x></x> | 54 |
| | :MATH:DUAL:OPERator | 54 |
| | :MATH:DUAL:POSition | 55 |
| | :MATH:DUAL:SCALe | 55 |
| | :MATH:FFT:SOURce | 56 |
| | :MATH:FFT:MAG | 56 |
| | :MATH:FFT:WINDow | |
| | :MATH:FFT:POSition | 57 |
| | :MATH:FFT:SCALe | |
| | :MATH:FFT:HORizontal:SCALe | 58 |
| | :MATH:FFT:HORizontal:POSition | 58 |
| | :MATH:DEFine | 58 |
| | :MATHVAR? | 59 |
| | :MATHVAR:VAR <x></x> | 60 |
| | :MATH:ADVanced:POSition | 60 |
| | :MATH:ADVanced:SCALe | 60 |
| Cursor | :CURSor:MODe | |
| | :CURSor:SOURce | |
| | :CURSor:HUNI | |
| | :CURSor:HUSE | 64 |
| | :CURSor:VUNI | 65 |



| | :CURSor:VUSE | 65 |
|---------|---|-------|
| | :CURSor:DDT | 65 |
| | :CURSor:H1Position | 66 |
| | :CURSor:H2Position | 66 |
| | :CURSor:HDELta | 67 |
| | :CURSor:V1Position | 67 |
| | :CURSor:V2Position | 67 |
| | :CURSor:VDELta | 68 |
| | :CURSor:XY:RECTangular:X:POSition <x></x> | 68 |
| | :CURSor:XY:RECTangular:X:DELta | 68 |
| | :CURSor:XY:RECTangular:Y:POSition <x></x> | 69 |
| | :CURSor:XY:RECTangular:Y:DELta | |
| | :CURSor:XY:POLar:RADIUS:POSition <x></x> | 69 |
| | :CURSor:XY:POLar:RADIUS:DELta | 70 |
| | :CURSor:XY:POLar:THETA:POSition <x></x> | 70 |
| | :CURSor:XY:POLar:THETA:DELta | 70 |
| | :CURSor:XY:PRODuct:POSition <x></x> | 71 |
| | :CURSor:XY:PRODuct:DELta | 71 |
| | :CURSor:XY:RATio:POSition <x></x> | 71 |
| | :CURSor:XY:RATio:DELta | 72 |
| Display | :DISPlay:INTensity:WAVEform | 73 |
| | :DISPlay:INTensity:GRATicule | |
| | :DISPlay:INTensity:BACKLight | 74 |
| | :DISPlay:INTensity:BACKLight:AUTODim:ENA | ble74 |
| | :DISPlay:INTENSITy:BACKLight:AUTODim:TIN | Ле74 |
| | :DISPlay:PERSistence | 75 |
| | :DISPlay:GRATicule | 75 |
| | :DISPlay:WAVEform | 76 |
| | :DISPlay:OUTPut | 76 |



| Hardcopy | :HARDcopy:START | 77 |
|----------|------------------------------------|----|
| ., | :HARDcopy:MODe | 77 |
| | :HARDcopy:PRINTINKSaver | 78 |
| | :HARDcopy:SAVEINKSaver | 78 |
| | :HARDcopy:SAVEFORMat | 78 |
| | :HARDcopy:ASSIGN | |
| Measure | :MEASure:GATing | 81 |
| | :MEASure:SOURce | 81 |
| | :MEASure:METHod | |
| | :MEASUrement:REFLevel:PERCent:HIGH | 82 |
| | :MEASUrement:REFLevel:PERCent:LOW | 83 |
| | :MEASUrement:REFLevel:PERCent:MID | 83 |
| | :MEASUrement:REFLevel:PERCent:MID2 | 83 |
| | :MEASure:FALL | 84 |
| | :MEASure:FOVShoot | 84 |
| | :MEASure:FPReshoot | 85 |
| | :MEASure:FREQuency | 85 |
| | :MEASure:NWIDth | 86 |
| | :MEASure:PDUTy | 86 |
| | :MEASure:PERiod | 87 |
| | :MEASure:PWIDth | 87 |
| | :MEASure:RISe | 88 |
| | :MEASure:ROVShoot | 88 |
| | :MEASure:RPReshoot | 89 |
| | :MEASure:PPULSE | 90 |
| | :MEASure:NPULSE | 90 |
| | :MEASure:PEDGE | 91 |
| | :MEASure:NEDGE | 91 |
| | :MEASure:AMPlitude | 92 |
| | :MEASure:MEAN | 92 |
| | :MEASure:CMEan | 93 |
| | :MEASure:HIGH | 93 |
| | :MEASure:LOW | 94 |
| | :MEASure:MAX | 95 |



| | :MEASure:MIN | 95 |
|-------------|---|-----|
| | :MEASure:PK2PK | 96 |
| | :MEASure:RMS | 96 |
| | :MEASure:CRMS | 97 |
| | :MEASure:AREa | 97 |
| | :MEASure:CARea | 98 |
| | :MEASure:FRRDelay | 99 |
| | :MEASure:FRFDelay | 99 |
| | :MEASure:FFRDelay | 100 |
| | :MEASure:FFFDelay | 100 |
| | :MEASure:LRRDelay | 101 |
| | :MEASure:LRFDelay | 102 |
| | :MEASure:LFRDelay | 102 |
| | :MEASure:LFFDelay | 103 |
| | :MEASure:PHAse | 103 |
| Measurement | :MEASUrement:MEAS <x>:SOURCE<x></x></x> | 105 |
| | :MEASUrement:MEAS <x>:TYPe</x> | 106 |
| | :MEASUrement:MEAS <x>:STATE</x> | 106 |
| | :MEASUrement:MEAS <x>:VALue</x> | 107 |
| | :MEASUrement:MEAS <x>:MAXimum</x> | 108 |
| | :MEASUrement:MEAS <x>:MEAN</x> | 109 |
| | :MEASUrement:MEAS <x>:MINImum</x> | 109 |
| | :MEASUrement:MEAS <x>:STDdev</x> | 110 |
| | :MEASUrement:STATIstics:MODe | 111 |
| | :MEASUrement:STATIstics:WEIghting | 111 |
| | :MEASUrement:STATIstics | 111 |
| Reference | :REF <x>:DISPlay</x> | 112 |
| | :REF <x>:TIMebase:POSition</x> | 112 |
| | :REF <x>:TIMebase:SCALe</x> | 113 |
| | :REF <x>:OFFSet</x> | 113 |
| | :REF <x>:SCALe</x> | 114 |
| Run | :RUN | 115 |

GW INSTEK

| Stop | :STOP | |
|----------|------------------------------|-----|
| Single | :SINGle | 115 |
| Force | :FORCe | |
| Timebase | :TIMebase:EXPand | 117 |
| | :TIMebase:POSition | 117 |
| | :TIMebase:SCALe | 117 |
| | :TIMebase:MODe | 118 |
| | :TIMebase:WINDow:POSition | 118 |
| | :TIMebase:WINDow:SCALe | 119 |
| Trigger | :TRIGger:FREQuency | 122 |
| | :TRIGger:TYPe | 122 |
| | :TRIGger:SOURce | 122 |
| | :TRIGger:COUPle | 123 |
| | :TRIGger:NREJ | 123 |
| | :TRIGger:MODe | 124 |
| | :TRIGger:HOLDoff | 124 |
| | :TRIGger:LEVel | 124 |
| | :TRIGger:HLEVel | 125 |
| | :TRIGger:LLEVel | 125 |
| | :TRIGger:EDGe:SLOP | 126 |
| | :TRIGger:DELay:SLOP | 126 |
| | :TRIGger:DELay:TYPe | 127 |
| | :TRIGger:DELay:TIMe | 127 |
| | :TRIGger:DELay:EVENt | 127 |
| | :TRIGger:DELay:LEVel | 128 |
| | :TRIGger:PULSEWidth:POLarity | |
| | :TRIGger:RUNT:POLarity | 128 |
| | :TRIGger:RUNT:WHEn | |
| | :TRIGger:RUNT:TIMe | |
| | :TRIGger:RISEFall:SLOP | |
| | :TRIGger:RISEFall:WHEn | |



| :TRIGger:RISEFall:TIMe | 131 |
|---------------------------------------|-----|
| :TRIGger:VIDeo:TYPe | 131 |
| :TRIGger:VIDeo:FIELd | |
| :TRIGger:VIDeo:LINe | 132 |
| :TRIGger:VIDeo:POLarity | 133 |
| :TRIGger:PULSe:WHEn | 133 |
| :TRIGger:PULSe:TIMe | 134 |
| :TRIGger:TIMEOut:WHEn | 134 |
| :TRIGger:TIMEOut:TIMER | 134 |
| :TRIGger:ALTernate | 135 |
| :TRIGger:STATe | 135 |
| :TRIGger:EXTERnal:PROBe:TYPe | 136 |
| :TRIGger:EXTERnal:PROBe:RATio | 136 |
| :TRIGger:BUS:TYPe | 137 |
| :TRIGger:BUS:THReshold:CH <x></x> | 137 |
| :TRIGger:BUS:B1:I2C:CONDition | 137 |
| :TRIGger:BUS:B1:I2C:ADDRess:MODe | 138 |
| :TRIGger:BUS:B1:I2C:ADDRess:TYPe | 139 |
| :TRIGger:BUS:B1:I2C:ADDRess:VALue | 139 |
| :TRIGger:BUS:B1:I2C:ADDRess:DIRection | 140 |
| :TRIGger:BUS:B1:I2C:DATa:SIZe | 140 |
| :TRIGger:BUS:B1:I2C:DATa:VALue | 141 |
| :TRIGger:BUS:B1:UART:CONDition | 142 |
| :TRIGger:BUS:B1:UART:RX:DATa:SIZe | 142 |
| :TRIGger:BUS:B1:UART:RX:DATa:VALue | 143 |
| :TRIGger:BUS:B1:UART:TX:DATa:SIZe | 143 |
| :TRIGger:BUS:B1:UART:TX:DATa:VALue | 144 |
| :TRIGger:BUS:B1:SPI:CONDition | 145 |
| :TRIGger:BUS:B1:SPI:DATa:SIZe | 145 |
| :TRIGger:BUS:B1:SPI:DATa:MISO:VALue | 146 |
| :TRIGger:BUS:B1:SPI:DATa:MOSI:VALue | 147 |
| :TRIGger:BUS:B1:CAN:CONDition | 147 |
| :TRIGger:BUS:B1:CAN:FRAMEtype | 148 |
| :TRIGger:BUS:B1:CAN:IDentifier:MODe | 148 |
| ·TRICger·RLIS·R1·CAN·IDentifier·VALue | 149 |



| | :TRIGger:BUS:B1:CAN:IDentifier:DIRection | 150 |
|----------------------|--|-----|
| | :TRIGger:BUS:B1:CAN:DATa:QUALifier | 150 |
| | :TRIGger:BUS:B1:CAN:DATa:SIZe | 151 |
| | :TRIGger:BUS:B1:CAN:DATa:VALue | 151 |
| | :TRIGger:BUS:B1:LIN:CONDition | 152 |
| | :TRIGger:BUS:B1:LIN:DATa:QUALifier | 154 |
| | :TRIGger:BUS:B1:LIN:DATa:SIZe | 154 |
| | :TRIGger:BUS:B1:LIN:DATa:VALue | 155 |
| | :TRIGger:BUS:B1:LIN:ERRTYPE | 155 |
| | :TRIGger:BUS:B1:LIN:IDentifier:VALue | 156 |
| System | :SYSTem:LOCK | 157 |
| commands | :SYSTem:ERRor | 157 |
| Save/Recall | :RECAll:SETUp | 158 |
| • | :RECAll:WAVEform | 158 |
| | :SAVe:IMAGe | 159 |
| | :SAVe:IMAGe:FILEFormat | 159 |
| | :SAVe:IMAGe:INKSaver | 160 |
| | :SAVe:SETUp | 160 |
| | :SAVe:WAVEform | 161 |
| | :SAVe:WAVEform:FILEFormat | 162 |
| Ethernet Commands | :ETHERnet:DHCP | 163 |
| Time Commands | :DATe | 164 |
| Bus Decode | :BUS1 | |
| Commands | :BUS1:STATE | |
| | :BUS1:TYPe | |
| | :BUS1:INPut | |
| | :BUS1:I2C:ADDRess:RWINClude | 167 |
| | :BUS1:I2C:SCLK:SOURce | 168 |
| | :BUS1:I2C:SDA:SOURce | 168 |
| | :BUS1:UART:BITRate | 168 |
| | :BUS1:UART:DATABits | 169 |



| :BUS1:UART:POLARity 17 :BUS1:UART:EOFPAcket 17 :BUS1:UART:TX:SOURce 17 :BUS1:UART:RX:SOURce 17 :BUS1:SPI:SCLK:POLARity 17 :BUS1:SPI:SS:POLARity 17 :BUS1:SPI:WORDSize 17 :BUS1:SPI:BITORder 17 :BUS1:SPI:SCLK:SOURce 17 :BUS1:SPI:SS:SOURce 17 :BUS1:SPI:MISO:SOURce 17 :BUS1:DISPlay:FORMAt 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:BOFORmat 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 18 :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:UART:PARIty169 |
|--|---------------|----------------------------|
| :BUS1:UART:EOFPAcket 17 :BUS1:UART:TX:SOURce 17 :BUS1:UART:RX:SOURce 17 :BUS1:SPI:SCLK:POLARity 17 :BUS1:SPI:SS:POLARity 17 :BUS1:SPI:WORDSize 17 :BUS1:SPI:BITORder 17 :BUS1:SPI:SOURce 17 :BUS1:SPI:MOSI:SOURce 17 :BUS1:SPI:MISO:SOURce 17 :BUS1:SPI:MISO:SOURce 17 :BUS1:CISPIQUECE 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:BOFORmat 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 18 :MARK:CREATE 18 :MARK:DELEte 18 :SEARCH:COPY 18 :SEARCH:TOTAL | | :BUS1:UART:PACKEt170 |
| :BUS1:UART:TX:SOURce 17 :BUS1:UART:RX:SOURce 17 :BUS1:SPI:SCLK:POLARity 17 :BUS1:SPI:SS:POLARity 17 :BUS1:SPI:WORDSize 17 :BUS1:SPI:WORDSize 17 :BUS1:SPI:BITORder 17 :BUS1:SPI:SSISOURce 17 :BUS1:SPI:MOSI:SOURce 17 :BUS1:SPI:MISO:SOURce 17 :BUS1:SPI:MISO:SOURce 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:PROBe 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:CAN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 18 :MARK:CREATE 18 :MARK:DELEte 18 :MARK:DELEte 18 :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:UART:POLARity170 |
| :BUS1:UART:RX:SOURce 17 :BUS1:SPI:SCLK:POLARity 17 :BUS1:SPI:WORDSize 17 :BUS1:SPI:WORDSize 17 :BUS1:SPI:BITORder 17 :BUS1:SPI:SCLK:SOURce 17 :BUS1:SPI:SS:SOURce 17 :BUS1:SPI:MOSI:SOURce 17 :BUS1:SPI:MISO:SOURce 17 :BUS1:DISPlay:FORMAt 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:PROBe 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:DFORmat 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 18 :MARK:CREATE 18 :MARK:DELEte 18 :MARK:DELEte 18 :SEARCH:COPY 18 :SEARCH:TOTAL 18 | | :BUS1:UART:EOFPAcket170 |
| :BUS1:SPI:SCLK:POLARity 17 :BUS1:SPI:WORDSize 17 :BUS1:SPI:BITORder 17 :BUS1:SPI:SCLK:SOURce 17 :BUS1:SPI:SS:SOURce 17 :BUS1:SPI:MOSI:SOURce 17 :BUS1:SPI:MISO:SOURce 17 :BUS1:DISPlay:FORMAt 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:SPROBe 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:IDFORmat 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 18 :MARK:CREATE 18 :MARK:DELEte 18 :MARK:DELEte 18 :SEARCH:COPY 18 :SEARCH:TOTAL 18 | | :BUS1:UART:TX:SOURce171 |
| :BUS1:SPI:SS:POLARity 17 :BUS1:SPI:WORDSize 17 :BUS1:SPI:BITORder 17 :BUS1:SPI:SCLK:SOURce 17 :BUS1:SPI:SS:SOURce 17 :BUS1:SPI:MOSI:SOURce 17 :BUS1:SPI:MISO:SOURce 17 :BUS1:DISPlay:FORMAt 17 :BUS1:DISPlay:FORMAt 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:PROBe 17 :BUS1:CAN:BITRate 17 :BUS1:CAN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:BOPOLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 18 :MARK:CREATE 18 :MARK:DELEte 18 :MARK:DELEte 18 :SEARCH:COPY 18 :SEARCH:TOTAL 18 | | :BUS1:UART:RX:SOURce171 |
| :BUS1:SPI:WORDSize 17 :BUS1:SPI:BITORder 17 :BUS1:SPI:SCLK:SOURce 17 :BUS1:SPI:SS:SOURce 17 :BUS1:SPI:MOS1:SOURce 17 :BUS1:SPI:MISO:SOURce 17 :BUS1:DISPlay:FORMAt 17 :LISTer:DATA 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:PROBe 17 :BUS1:CAN:BITRate 17 :BUS1:CAN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 18 :MARK:CREATE 18 :MARK:DELEte 18 :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:SPI:SCLK:POLARity172 |
| :BUS1:SPI:BITORder .17 :BUS1:SPI:SCLK:SOURce .17 :BUS1:SPI:SS:SOURce .17 :BUS1:SPI:MISO:SOURce .17 :BUS1:DISPlay:FORMAt .17 :BUS1:CAN:SOURce .17 :BUS1:CAN:SOURce .17 :BUS1:CAN:PROBe .17 :BUS1:CAN:PROBe .17 :BUS1:CAN:BITRate .17 :BUS1:LIN:BITRate .17 :BUS1:LIN:BITRate .17 :BUS1:LIN:POLARity .17 :BUS1:LIN:SAMPLEpoint .17 :BUS1:LIN:SOURce .17 :BUS1:LIN:STANDard .17 :BUS1:LIN:STANDard .17 :BUS1:LIN:STANDard .18 :MARK:CREATE .18 :MARK:DELEte .18 :MARK:DELEte .18 :SEARCH:COPY .18 Commands :SEARCH:STATE .18 :SEARCH:TOTAL .18 | | :BUS1:SPI:SS:POLARity172 |
| :BUS1:SPI:SCLK:SOURce 17 :BUS1:SPI:SS:SOURce 17 :BUS1:SPI:MOSI:SOURce 17 :BUS1:SPI:MISO:SOURce 17 :BUS1:DISPlay:FORMAt 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:PROBe 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:CAN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:IDFORmat 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 18 :MARK:CREATE 18 :MARK:DELEte 18 :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:SPI:WORDSize172 |
| :BUS1:SPI:SS:SOURce 17 :BUS1:SPI:MOSI:SOURce 17 :BUS1:SPI:MISO:SOURce 17 :BUS1:DISPlay:FORMAt 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:PROBe 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:CAN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:IDFORmat 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 18 :MARK:CREATE 18 :MARK:DELEte 18 :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:SPI:BITORder173 |
| :BUS1:SPI:MOSI:SOURce 17 :BUS1:SPI:MISO:SOURce 17 :BUS1:DISPlay:FORMAt 17 :LISTer:DATA 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:PROBe 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:IDFORmat 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 17 :MARK:CREATE 18 :MARK:DELEte 18 :Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:SPI:SCLK:SOURce173 |
| :BUS1:SPI:MISO:SOURce 17 :BUS1:DISPlay:FORMAt 17 :LISTer:DATA 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:PROBe 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:CAN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:IDFORmat 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:STANDard 17 :BUS1:LIN:STANDard 18 :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:SPI:SS:SOURce173 |
| :BUS1:DISPlay:FORMAt 17 :LISTer:DATA 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:PROBe 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:CAN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:IDFORmat 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 Mark Commands :MARK :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:SPI:MOSI:SOURce174 |
| :LISTer:DATA 17 :BUS1:CAN:SOURce 17 :BUS1:CAN:PROBe 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:CAN:BITRate 17 :BUS1:LIN:IDFORmat 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 Mark Commands :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:SPI:MISO:SOURce174 |
| :BUS1:CAN:SOURce 17 :BUS1:CAN:PROBe 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:CAN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:IDFORmat 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 Mark Commands :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:DISPlay:FORMAt174 |
| :BUS1:CAN:PROBe 17 :BUS1:CAN:SAMPLEpoint 17 :BUS1:CAN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:IDFORmat 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 Mark Commands :MARK :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :LISTer:DATA175 |
| :BUS1:CAN:SAMPLEpoint 17 :BUS1:CAN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:IDFORmat 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 Mark Commands :MARK 18 :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:CAN:SOURce175 |
| :BUS1:CAN:BITRate 17 :BUS1:LIN:BITRate 17 :BUS1:LIN:IDFORmat 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 Mark Commands :MARK :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:CAN:PROBe175 |
| :BUS1:LIN:BITRate 17 :BUS1:LIN:IDFORmat 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 Mark Commands :MARK 18 :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:CAN:SAMPLEpoint176 |
| :BUS1:LIN:IDFORmat 17 :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 Mark Commands :MARK :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:CAN:BITRate176 |
| :BUS1:LIN:POLARity 17 :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 Mark Commands :MARK :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY Commands :SEARCH:STATE :SEARCH:TOTAL 18 | | :BUS1:LIN:BITRate177 |
| :BUS1:LIN:SAMPLEpoint 17 :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 Mark Commands :MARK :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:LIN:IDFORmat177 |
| :BUS1:LIN:SOURce 17 :BUS1:LIN:STANDard 17 Mark Commands :MARK 18 :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:LIN:POLARity177 |
| :BUS1:LIN:STANDard 17 Mark Commands :MARK 18 :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:LIN:SAMPLEpoint178 |
| Mark Commands :MARK 18 :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:LIN:SOURce178 |
| :MARK:CREATE 18 :MARK:DELEte 18 Search :SEARCH:COPY 18 Commands :SEARCH:STATE 18 :SEARCH:TOTAL 18 | | :BUS1:LIN:STANDard178 |
| :MARK:DELEte | Mark Commands | :MARK180 |
| Search :SEARCH:COPY | | :MARK:CREATE180 |
| Search :SEARCH:COPY | | :MARK:DELEte181 |
| :SEARCH:TOTAL18 | Search | :SEARCH:COPY183 |
| | Commands | :SEARCH:STATE184 |
| :SEARCH:TRIGger:TYPe18 | | :SEARCH:TOTAL184 |
| | | :SEARCH:TRIGger:TYPe184 |



| :SEARCH:TRIGger:SOURce | .185 |
|---|------|
| :SEARCH:TRIGger:EDGE:SLOP | |
| :SEARCH:TRIGger:LEVel | |
| :SEARCH:TRIGger:HLEVel | |
| :SEARCH:TRIGger:LLEVel | |
| :SEARCH:TRIGger:PULSEWidth:POLarity | |
| :SEARCH:TRIGger:RUNT:POLarity | |
| :SEARCH:TRIGger:RISEFall:SLOP | .188 |
| :SEARCH:TRIGger:PULSe:WHEn | |
| :SEARCH:TRIGger:PULSe:TIMe | .189 |
| :SEARCH:TRIGger:RUNT:WHEn | |
| :SEARCH:TRIGger:RUNT:TIMe | |
| :SEARCH:TRIGger:RISEFall:WHEn | |
| :SEARCH:TRIGger:RISEFall:TIMe | .191 |
| :SEARCH:TRIGger:BUS:TYPe | |
| :SEARCH:TRIGger:BUS:B1:I2C:CONDition | |
| :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:MODe | .193 |
| :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:TYPe | .193 |
| : SEARCH: TRIGger: BUS: B1: I2C: ADDRess: VALue | .194 |
| :SEARCH:TRIGger:BUS:B1:I2C:ADDRess | |
| :DIRection | |
| :SEARCH:TRIGger:BUS:B1:I2C:DATa:SIZe | |
| :SEARCH:TRIGger:BUS:B1:I2C:DATa:VALue | |
| : SEARCH: TRIGger: BUS: B1: UART: CONDition | |
| : SEARCH: TRIGger: BUS: B1: UART: RX: DATa: SIZe | .197 |
| : SEARCH: TRIGger: BUS: B1: UART: RX: DATa: VALue . | .198 |
| $: SEARCH: TRIGger: BUS: B1: UART: TX: DATa: SIZe \dots \\$ | |
| : SEARCH: TRIGger: BUS: B1: UART: TX: DATa: VALue . | |
| : SEARCH: TRIGger: BUS: B1: SPI: CONDition | .200 |
| :SEARCH:TRIGger:BUS:B1:SPI:DATa:SIZe | .200 |
| :SEARCH:TRIGger:BUS:B1:SPI:DATa:MISO | |
| :VALue | .201 |
| :SEARCH:TRIGger:BUS:B1:SPI:DATa:MOSI | |
| :VALue | |
| :SEARCH:TRIGger:BUS:B1:CAN:CONDition | .203 |



| | :SEARCH:TRIGger:BUS:B1:CAN:FRAMEtype | 204 |
|----------------|---|-----|
| | :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODe | 204 |
| | :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:VALue | 205 |
| | :SEARCH:TRIGger:BUS:B1:CAN:IDentifier | |
| | :DIRection | 205 |
| | : SEARCH: TRIGger: BUS: B1: CAN: DATa: QUALifier. | 206 |
| | :SEARCH:TRIGger:BUS:B1:CAN:DATa:SIZe | 207 |
| | :SEARCH:TRIGger:BUS:B1:CAN:DATa:VALue | 207 |
| | :SEARCH:TRIGger:BUS:B1:LIN:CONDition | 208 |
| | $: SEARCH: TRIGger: BUS: B1: LIN: DATa: QUALifier \dots \\$ | 209 |
| | :SEARCH:TRIGger:BUS:B1:LIN:DATa:SIZe | 210 |
| | :SEARCH:TRIGger:BUS:B1:LIN:DATa:VALue | 210 |
| | :SEARCH:TRIGger:BUS:B1:LIN:ERRTYPE | 211 |
| | : SEARCH: TRIGger: BUS: B1: LIN: IDentifier: VALue | 212 |
| | :SEARCH:FFTPeak:METHod | 212 |
| | :SEARCH:FFTPeak:METHod:MPEak | 213 |
| | :SEARCH:FFTPeak:SINFo | |
| Label Commands | :CHANnel <x>:LABel</x> | 215 |
| | :CHANnel <x>:LABel:DISPlay</x> | 216 |
| | :REF <x>:LABel</x> | |
| | :REF <x>:LABel:DISPlay</x> | 217 |
| | :BUS1:LABel | 218 |
| | :BUS1:LABel:DISPlay | 218 |
| | :SET <x>:LABel</x> | 219 |
| Segment | :SEGMents:STATE | 221 |
| Commands | :SEGMents:CURRent | 222 |
| | :SEGMents:TOTalnum | 222 |
| | :SEGMents:TIMe | 223 |
| | :SEGMents:DISPALL | 223 |
| | :SEGMents:MEASure:MODe | 223 |
| | :SEGMents:MEASure:PLOT:SOURce | 224 |
| | : SEGMents: MEASure: PLOT: DIVide | 224 |
| | :SEGMents:MEASure:PLOT:SELect | 225 |



| | :SEGMents:MEASure:PLOT:RESults | 225 |
|--------------|--------------------------------|-----|
| | :SEGMents:MEASure:TABle:SOURce | 226 |
| | :SEGMents:MEASure:TABle:SELect | 226 |
| | :SEGMents:MEASure:TABle:LIST | 227 |
| | :SEGMents:MEASure:TABle:SAVe | 227 |
| | :SEGMents:SAVe | 228 |
| | :SEGMents:SAVe:SOURce | 228 |
| | :SEGMents:SAVe:SELect:STARt | 228 |
| | :SEGMents:SAVe:SELect:END | 229 |
| DVM Commands | :DVM:STATE | 230 |
| | :DVM:SOURce | 230 |
| | :DVM:MODe | 231 |
| | :DVM:VALue | 231 |
| Go-NoGo | :GONogo:CLEar | 232 |
| Commands | :GONogo:EXECute | 232 |
| | :GONogo:FUNCtion | 233 |
| | :GONogo:NGCount | 233 |
| | :GONogo:NGDefine | 233 |
| | :GONogo:SOURce | 234 |
| | :GONogo:VIOLation | 234 |
| | :GONogo:SCRipt | 234 |
| | :TEMPlate:MODe | 235 |
| | :TEMPlate:MAXimum | 235 |
| | :TEMPlate:MINimum | 235 |
| | :TEMPlate:POSition:MAXimum | 236 |
| | :TEMPlate:POSition:MINimum | 236 |
| | :TEMPlate:SAVe:MAXimum | 236 |
| | :TEMPlate:SAVe:MINimum | 237 |
| | :TEMPlate:TOLerance | 237 |
| | :TEMPlate:SAVe:AUTo | 237 |
| Data Logging | :DATALOG:STATE | 238 |
| Commands | :DATALOG:SOURce | 238 |

COMMAND OVERVIEW



| | :DATALOG:SAVe | 239 |
|-------------|-----------------------|-----|
| | :DATALOG:INTerval | 239 |
| | :DATALOG:DURation | 240 |
| | | |
| Remote Disk | :REMOTEDisk:IPADDress | 241 |
| Commands | :REMOTEDisk:PATHName | 241 |
| | :REMOTEDisk:USERName | 241 |
| | :REMOTEDisk:PASSWord | 242 |
| | :REMOTEDisk:MOUNT | 242 |
| | :RFMOTFDisk:AUTOMount | 243 |

COMMAND DETAILS

The Command details chapter shows the detailed syntax, equivalent panel operation, and example for each command. For the list of all commands, see page17.

| Common Commands 32 |
|--------------------------|
| Acquisition Commands 38 |
| Autoscale Commands |
| Vertical Commands 47 |
| Math Commands 53 |
| Cursor Commands 62 |
| Display Commands 73 |
| Hardcopy Commands77 |
| Measure Commands 80 |
| Measurement Commands 105 |
| Reference Commands 112 |
| Run Command 115 |
| Stop Command |
| Single Command |
| Force Command |
| Timebase Commands |
| Trigger Commands 120 |
| System Commands 157 |
| Save/Recall Commands |
| Ethernet Commands |
| Time Commands |
| Bus Decode Commands 165 |
| Mark Commands |
| Search Commands |

COMMAND DETAILS



| Label Commands | 215 |
|-----------------------|-----|
| Segment Commands | 221 |
| DVM Commands | 230 |
| Go_NoGo Commands | 232 |
| Data Logging Commands | 238 |
| Remote Disk Commands | 241 |



Common Commands

| *IDN? | 32 |
|-------|----|
| *LRN? | 32 |
| *SAV | 33 |
| *RCL | 33 |
| *RST | 34 |
| *CLS | 34 |
| *ESE | 34 |
| *ESR | 35 |
| *OPC | 35 |
| *SRE | 36 |
| *STB | 36 |
| | |

*IDN;



| Description | Returns the manufacturer, model, serial number and version number of the unit. |
|-------------|--|
| Syntax | *IDN? |
| Example | *IDN? |
| | GW,GDS-2074E,PXXXXXX,V1.XX |

*LRN?



| Description | Returns the oscilloscope settings as a data string. | |
|-------------|---|--|
| Syntax | *LRN? | |
| Example | *LRN? | |

:DISPlay:WAVEform VECTOR;PERSistence 2.400E-01; INTensity:WAVEform 50;INTensity:GRATicule 50;GRATicule FULL;:CHANnel CH1:DISPlay ON;BWLimit FULL;COUPling DC;INVert OFF;POSition -1.600E+00;PROBe:RATio



**∜C \ \ **

1.000e+01;PROBe:TYPe VOLTAGE;SCALe 2.000E+ 01;IMPedance 1E+6;EXPand GROUND::CHANnel CH2:DISPlay ON;BWLimit FULL;COUPling DC;INVert OFF;POSition 0.000E+00;PROBe:RATio 1.000e+01;PROBe:TYPe VOLTAGE;SCALe 2.000E+00;IMPedance 1E+6;EXPand GROUND;:CHANnel CH3:DISPlay OFF;BWLimit FULL;COUPling DC;INVert OFF;POSition 0.000E+00;PROBe:RATio 1.000e+01;PROBe:TYPe VOLTAGE; SCALe 1.000E+00; IMPedance 1E+6; EXPand GROUND::CHANnel CH4:DISPlay OFF;BWLimit FULL;COUPling DC;INVert OFF;POSition 0.000E+00;PROBe:RATio 1.000e+01;PROBe:TYPe VOLTAGE;SCALe 1.000E+00;IMPedance 1E+6;EXPand GROUND;:MATH:TYPe FFT;DISP OFF:DUAL:SOURce1 CH1:SOURce2 CH2:OPERator MUL;POSition 0.000E+00;SCALe ?;FFT:SOURce CH1;MAG DB;WINDow HANNING;POSition 2.800E-01;SCALe 2.000E+01;MATH:ADVanced:OPERator DIFF;ADVanced:SOURce CH1;ADVanced:EDIT: SOURce1 CH1:ADVanced:EDIT:S

Sot

| *SAV | <u>Set</u> → |
|-------------|--|
| Description | Saves the current panel settings to the selected memory number (setup $1 \sim 20$). |
| Syntax | *SAV {1 2 3 20} |
| Example | *SAV 1 |
| | Saves the current panel settings to Set 1. |
| *RCL | Set → |
| Description | Recalls a set of panel settings. |
| Syntax | *RCL {1 2 3 20} |
| Example | *RCL 1 |
| | Recalls the selected setup from Set 1. |



| *RST | | | | Set → | | |
|------------------|--|--------|-------|------------------------|--|--|
| Description | Resets the GDS-2000E (recalls the default panel settings). | | | | | |
| Syntax | *RST | | | | | |
| *CLS | | | | (Set)→ | | |
| Description | Clears the error queue. | | | | | |
| Syntax | *CLS | | | | | |
| *ESE | | | | Set → Query | | |
| Description | Sets or queries the Standard Event Status Enable register. | | | | | |
| Syntax | *ESE <nr1></nr1> | | | | | |
| Query Syntax | *ESE? | | | | | |
| Return parameter | <nr1> 0~255</nr1> | | | | | |
| Bit Weight | Bit# | Weight | Event | Description | | |
| | 0 | 1 | OPC | Operation Complete Bit | | |
| | 1 | 2 | RQC | Not used | | |
| | 2 | 4 | QYE | Query Error | | |
| | 3 | 8 | DDE | Device Error | | |
| | 4 | 16 | EXE | Execution Error | | |
| | 5 | 32 | CME | Command Error | | |
| | 6 | 64 | URQ | User Request | | |
| | 7 | 128 | PON | Power On | | |
| Example | *ESE | þ | | | | |
| | >4 | | | | | |
| | Indicates that there is a query error. | | | | | |



| *ESR | | | | → Query | | | |
|------------------|---|--------|-------|------------------------|--|--|--|
| Description | Queries the Standard Event Status (Event) register. The Event Status register is cleared after it is read. | | | | | | |
| Query Syntax | *ESR? | | | | | | |
| Return parameter | · <nr1> 0~255</nr1> | | | | | | |
| Bit Weight | Bit# | Weight | Event | Description | | | |
| | 0 | 1 | OPC | Operation Complete Bit | | | |
| | 1 | 2 | RQC | Not used | | | |
| | 2 | 4 | QYE | Query Error | | | |
| | 3 | 8 | DDE | Device Error | | | |
| | 4 | 16 | EXE | Execution Error | | | |
| | 5 | 32 | CME | Command Error | | | |
| | 6 | 64 | URQ | User Request | | | |
| | 7 | 128 | PON | Power On | | | |
| Example | *ESR? >4 | | | | | | |
| | | | | | | | |
| | Indicates that there is a query error. | | | | | | |
| | Set → | | | | | | |
| *OPC | | | | → Query | | | |
| Description | The *OPC command sets the OPC bit (bit0) of the Standard Event Status Register when all current commands have been processed. | | | | | | |
| | The *OPC? Query returns 1 when all the outstanding commands have completed. | | | | | | |
| Syntax | *OPC | • | | | | | |
| Query Syntax | *OPC | :? | | | | | |
| Return parameter | Returns 1 when all the outstanding commands have completed. | | | | | | |



| *SRE | | | | Set → Query | | | |
|--------------------------------|---|----------|-------|-----------------------|--|--|--|
| Description | Sets or queries the Service Request Enable register. The Service Request Enable register determines which registers of the Status Byte register are able to generate service requests. | | | | | | |
| Syntax | *SRE <nr1></nr1> | | | | | | |
| Query Syntax | *SRE? | | | | | | |
| Parameter/ Return parameter | <nr1< td=""><td>> 0~255</td><td></td><td></td></nr1<> | > 0~255 | | | | | |
| Bit Weight | Bit# | Weight | Event | Description | | | |
| | 0 | 1 | | Not used | | | |
| | 1 | 2 | | Not used | | | |
| | 2 | 4 | | Not used | | | |
| | 3 | 8 | | Not used | | | |
| | 4 | 16 | MAV | Message Available Bit | | | |
| | 5 | 32 | ESB | Event Status Bit | | | |
| | 6 | 64 | MSS | Master Summary Bit | | | |
| | 6 | 64 | RQS | Request Service Bit | | | |
| | 7 | 128 | | Not used | | | |
| Example | *SRE? | | | | | | |
| | >48 | | | | | | |
| | Indicates that the MAVB and ESB bit are both set. | | | | | | |
| *STB | | | | → Query | | | |
| Description | Queries the bit sum of the Status Byte register with MSS (Master summary Status) replacing the RQS bit (bit 6). | | | | | | |
| Query Syntax | *STB |) | | | | | |
| Return parameter | <nr1< td=""><td>> 0 ~ 25</td><td>5</td><td></td></nr1<> | > 0 ~ 25 | 5 | | | | |



| Bit Weight | Bit# | Weight | Event | Description |
|------------|------|----------|-------|-----------------------|
| | 0 | 1 | | Not used |
| | 1 | 2 | | Not used |
| | 2 | 4 | | Not used |
| | 3 | 8 | | Not used |
| | 4 | 16 | MAV | Message Available Bit |
| | 5 | 32 | ESB | Event Status Bit |
| | 6 | 64 | MSS | Master Summary Bit |
| | 6 | 64 | RQS | Request Service Bit |
| | 7 | 128 | | Not used |
| Example | *STB | , | | |
| | >16 | | | |

>16

Indicates that the MAV bit is set.



Acquisition Commands

| | :ACQuire:AVERage | 38 | | |
|-------------|---|---------|--|--|
| | :ACQuire:MODe | 39 | | |
| | :ACQuire <x>:MEMory?</x> | 39 | | |
| | :ACQuire:FILTer:SOURce | 41 | | |
| | :ACQuire:FILTer | 42 | | |
| | :ACQuire:FILTer:FREQuency | 42 | | |
| | :ACQuire:FILTer:FREQuency:UPPER | 42 | | |
| | :ACQuire:FILTer:FREQuency:LOWER | 43 | | |
| | :ACQuire:FILTer:TYPe | 43 | | |
| | :ACQuire:FILTer:TRACking | 44 | | |
| | :ACQuire <x>:STATe?</x> | 44 | | |
| | :ACQuire:INTERpolation | 44 | | |
| | :ACQuire:RECOrdlength | 45 | | |
| | :HEADer | 45 | | |
| | | | | |
| | Set → | | | |
| :ACQuire:AV | /ERage →Query | | | |
| Description | Selects or returns the number of waveform | | | |
| | acquisitions that are averaged in the average | | | |
| | acquisition mode. | | | |
| Syntax | :ACQuire:AVERage { <nr1> ?}</nr1> | | | |
| Related | :ACQuire:MODe | | | |
| Commands | | | | |
| Parameter | <nr1> 2, 4, 8, 16, 32, 64, 128, 256</nr1> | | | |
| Note | Before using this command, select the avera | ge | | |
| | acquisition mode. See the example below. | | | |
| Example | :ACQuire:MODe AVERage | | | |
| | :ACQuire:AVERage 2 | | | |
| | Selects the average acquisition mode, and se | ets the | | |
| | average number to 2. | | | |



| :ACQuire:MOD |)e | Set → Query | | |
|---|---|---------------------------------------|--|--|
| Description | Selects or returns the acquisition mode. | | | |
| Syntax | :ACQuire:MODe {SAMPle PDETect AVERage ?} | | | |
| Related Commands | :ACQuire:AVERage | | | |
| Parameter | SAMPle | Sample mode sampling | | |
| | PDETect | Peak detect sampling | | |
| | AVERage | Average sampling mode | | |
| Example :ACQuire:MODe PDETect | | MODe PDETect | | |
| Sets the sampling mode to peak detection. | | | | |
| :ACQuire <x>:N</x> | 1EMory? | — Query | | |
| Description | Returns the data in acquisition memory for the selected channel as a header + raw data. | | | |
| Syntax | :ACQuire <x>:MEMory?</x> | | | |
| Related | :ACQuire:RECOrdlength | | | |
| Commands | :HEADer | | | |
| Parameter | <x></x> | Channel number (1 to 4) | | |
| Return parameter | <string></string> | Returns acquisition settings followed | | |

<string>

Format:

data.

eform Data:

selected channel.

<waveform block data>

by raw waveform block data.

Returns the acquisition settings for the

parameter(1),setting(1);parameter(2),se
tting(2)...parameter(n),setting(n);Wav

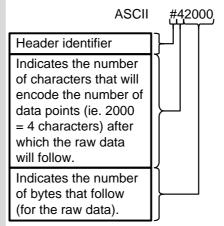
Header followed by the raw waveform

<waveform

block data>

Format:

Header: The header (in ASCII) encodes the number of bytes for the header followed by the number of data points in bytes for the raw data.



Raw Data:

Each two bytes (in hex) encodes the vertical data of a data point. The data is signed hex data (2's complement, $-32768 \sim 32767$).

Waveform Raw Data Example:

Header raw data.....

Hex:

23 34 32 30 30 30 00 1C 00 1B 00 1A 00 1A 00 1B

ASCII/Decimal:

#42000 28 27 26 26 27......

The actual value of a data point can be calculated with the following formula: (Decimal value of hex data/AD Factor) * vertical scale.

Note: AD Factor is fixed as 25. The vertical scale is returned with the

| | acquisition settings that precede the raw data. For example if the raw data for a point is $001C$ (=28 decimal) then, $(28/25) \times 0.5 = 0.56V$ |
|---------|---|
| Example | :ACQuire1:MEMory? |
| | Format,2.0E;Memory Length,10000;IntpDistance,0; Trigger Address,4999;Trigger Level,1.160E+01; Source,CH1;Vertical Units,V;Vertical Units Div,0;Vertical Units Extend Div,15;Label,ACK;Probe Type,0;Probe Ratio,1.000e+01;Vertical Scale,5.000e+00;Vertical Position,-1.100e+01;Horizontal Units,S;Horizontal Scale,5.000E-04;Horizontal Position,0.000E+00; Horizontal Mode,Main;SincET Mode,Real Time;Sampling Period,5.000e-07;Horizontal Old Scale,5.000E-04;Horizontal Old Position,0.000E+00; Firmware,V0.99b8;Time,02-Oct-14 17:00:43; Waveform Data; #520000 |

:ACQuire:FILTer:SOURce



| Description | Returns the | source of the filter. |
|--------------------------------|----------------|--------------------------------|
| Syntax | :ACQuire:FIL | Ter:SOURce {CH1 CH2 CH3 CH4 ?} |
| Parameter/ Return parameter | CH1 ~ CH4 | Source channel |
| Example | :ACQuire:FIL | Ter:SOURce? |
| | CH1 | |
| | Sets the filte | er source to CH1. |



Set)→ :ACQuire:FILTer (Query Turns the filter on/off or queries its status. Description :ACQuire:FILTer {ON|OFF|?} Syntax Parameter/ ON Filter on. Return parameter OFF Filter off. Example :ACQuire:FILTer? **OFF** Indicates that the filter is turned off. Set) :ACQuire:FILTer:FREQuency (Query Description Sets or queries the filter frequency. :ACQuire:FILTer:FREQuency {DEFault|<NRf>|?} Syntax Parameter/ **DEFault** Sets the filter frequency to the default. Return parameter <NRf> Manually sets the filter frequency. $(1Hz \sim 500MHz)$ Example :ACQuire:FILTer:FREQuency 1 Sets the filter frequency to 1Hz. Set) :ACQuire:FILTer:FREQuency:UPPER (Query Sets or returns the filter upper frequency. Description Syntax :ACQuire:FILTer:FREQuency:UPPER {DEFault} :ACQuire:FILTer:FREQuency:UPPER <NRf> :ACQuire:FILTer:FREQuency:UPPER? Parameter **DEFault** Sets the frequency to default.

Sets the frequency to user.(Range:1Hz~500MHz)

<NRf>



| Example | :ACQuire:FILTer:FREQuency:UPPER 4.95e+07 |
|---------|--|
| | |

:ACQuire:FILTer:FREQuency:UPPER?

4.950000e+07

:ACQuire:FILTer:FREQuency:LOWER



| Description | Sets or returns the filter lower frequency. | | |
|--------------|---|---|--|
| Syntax | :ACQuire:FILTer:FREQuency:LOWER {DEFault} | | |
| | :ACQuire:FILTer:FREQuency:LOWER <nrf></nrf> | | |
| | :ACQuire:FI | LTer:FREQuency:LOWER? | |
| Parameter | DEFault | Sets the frequency to default. | |
| | <nrf></nrf> | Sets the frequency to user.(Range:1Hz~500MHz) | |
| Example | :ACQuire:FILTer:FREQuency:LOWER 1.25e+07 | | |
| | :ACQuire:FILTer:FREQuency:LOWER? | | |
| 1.250000e+07 | | 07 | |

:ACQuire:FILTer:TYPe



| Description | Sets or returns the filter type. | | |
|-------------|--|----------------|--|
| Syntax | :ACQuire:FILTer:TRACking {LOWPass HIGHPass BANDPass} | | |
| | :ACQuire:FII | LTer:TYPe? | |
| Parameter | LOWPass | Lowpass Type. | |
| | HIGHPass | Highpass Type. | |
| | BANDPass | bandpass Type. | |
| Example | :ACQuire:FILTer:TYPe? | | |
| | >LOWPass | | |
| | Returns low pass type as present filter type | | |



Return parameter

:ACQuire:FILTer:TRACking Description Turns filter tracking on/off or queries its state. Syntax :ACQuire:FILTer:TRACking {ON|OFF|?} Parameter/ OFF Tracking off

Tracking on

Example :ACQuire:FILTer:TRACking ON
Turns filter tracking on.

ON

:ACQuire<X>:STATe?



| * | | | |
|------------------|---|-------------------------|--|
| Description | Returns the status of waveform data. | | |
| Syntax | :ACQuire <x>:STATe?</x> | | |
| Parameter | <x></x> | Channel number (1 to 4) | |
| Return parameter | 0 | Raw data is not ready | |
| | 1 | Raw data is ready | |
| Example | :ACQuire1:STATe? | | |
| | 0 | | |
| | Returns 0. Channel 1's raw data is not ready. | | |

Note: If the oscilloscope changes the acquisition status from STOP to RUN, the status will be reset as zero.

Description Selects or returns the interpolation mode. Syntax :ACQuire:INTERpolation {ET | SINC | ?} Parameter/Return parameter ET Equivalent Time interpolation. The GDS-2000E doesn't support ET. SINC Sets to SIN(X)/X interpolation



Example :ACQuire:INTERpolation?

>SINC

Returns SINC as the interpolation mode.

:ACQuire:RECOrdlength

| Set |) |
|------|----------|
| → Qı | uery) |

| Syntax :ACQuire:RECOrdlength { <ni <nrf="" parameter="" return=""> Record length. Set</ni> | |
|--|---|
| Parameter/Return < NRf> Record length. Set | Rf> ?} |
| parameter (1e+3 1e+4 1e- | table record length: +5 1e+6 1e+7) |

Example :ACQuire:RECOrdlength 1e+3

:HEADer ON

Sets the record length to 1000 points.

:HEADer

Example



| Description | Configures whether the returned data of the :ACQuire:MEM query will contain header information or not. It is set to ON by default. | | |
|---------------------|--|-------------------------------|--|
| Syntax | :HEADer {OFF ON ?} | | |
| Related Commands | :ACQuire< | <x>:MEMory?</x> | |
| Parameter | ON | Add header information. | |
| | OFF | Don't add header information. | |
| Return parameter | Returns the configuration (ON, OFF) for the selected channel. | | |



Autoscale Commands

:AUTOSet



Description Runs the Autoset function to automatically configure the horizontal scale, vertical scale, and trigger according to the input signal.

Syntax :AUTOSet

:AUTORSET:MODe



| Description | Sets the Autoset mode or queries its state. | | |
|---------------------|---|------------------|--|
| Syntax | :AUTORSET:MODe {FITScreen ACPriority ?} | | |
| Related Commands | :AUTOSet | | |
| Parameter/Return | FITScreen | Fit Screen mode | |
| parameter | ACPriority | AC priority mode | |

FITSCREEN

:AUTORSET?

Example

Vertical Commands

| | :CHANnel <x>:BWLimit47</x> | | |
|------------------|-----------------------------|--|--|
| | :CHANnel <x>:COUPling48</x> | | |
| | :CHANnel <x>:DESKew48</x> | | |
| | | l <x>:DISPlay48</x> | |
| | | l <x>:EXPand49</x> | |
| | :CHANne | I <x>:IMPedance?49</x> | |
| | | l <x>:INVert50</x> | |
| | | I <x>:POSition50</x> | |
| | | I <x>:PROBe:RATio51</x> | |
| | | l <x>:PROBe:TYPe51</x> | |
| | :CHANne | l <x>:SCALe51</x> | |
| | | | |
| CHANnel <x>:</x> | BWLimit | (Set)→ Query | |
| Description | Sets or re | turns the bandwidth limit on/off. | |
| Syntax | :CHANne | <x>:BWLimit {FULL <nr3> ?}</nr3></x> | |
| Parameter | <x></x> | Channel 1,2,3,4 | |
| | FULL | Full bandwidth | |
| | <nr3></nr3> | Sets the bandwidth limit to a pre-defined bandwidth. | |
| | | 100E+6: 100MHz | |
| | | 20E+6: 20MHz | |
| Return Parameter | <nr3></nr3> | Returns the bandwidth. | |
| | Full | Full bandwidth | |
| Example | :CHANne | 11:BWLimit 2.000E+07 | |
| | Sets the c | hannel 1 bandwidth to 20MHz. | |



| :CHANnel <x>:</x> | COUPlin | g Set \longrightarrow Query |
|--|---|---|
| Description | Selects or | r returns the coupling mode. |
| Syntax | CHANnel | <x>:COUPling {AC DC GND ?}</x> |
| Parameter | <x></x> | Channel 1,2,3,4 |
| | AC | AC coupling |
| | DC | DC coupling |
| | GND | Ground coupling |
| Return parameter | Returns t | he coupling mode. |
| Example | :CHANne | l1:COUPling DC |
| | Sets the d | coupling to DC for Channel 1. |
| | | <u>Set</u> → |
| :CHANnel <x>:</x> | DESKew | → Query |
| Description | Sets the deskew time in seconds. | |
| Syntax | :CHANne | el <x>:DESKew { <nr3> ?}</nr3></x> |
| Parameter | <x></x> | Channel 1,2,3,4 |
| | <nr3></nr3> | Deskew time: -5.00E -11 to 5.00E-11 -50ns to 50 ns. (10 ps /step) |
| Return parameter | <nr3></nr3> | Returns the deskew time. |
| Example | :CHANne | l1:DESKew 1.300E-9 |
| | Sets the d | deskew time to 1.3 nano seconds. |
| $\begin{array}{ccc} & & & & & \\ & & & \\ \text{:CHANnel} < \text{X} > : \text{DISPlay} & & & & \\ & & & & \\ \end{array}$ | | |
| Description | Turns a c | channel on/off or returns its status. |
| Syntax | :CHANnel <x>:DISPlay {OFF ON ?}</x> | |
| Parameter | <x></x> | Channel 1,2,3,4 |
| | OFF | Channel off |
| | ON | Channel on |



| Return Parameter | ON | N Channel is on | |
|---|---|--|--|
| | OFF | Channel is off | |
| Example | :CHANne | l1:DISPlay ON | |
| | Turns on | Channel 1 | |
| | | (Set)→ | |
| :CHANnel <x>:</x> | EXPand | Query | |
| Description | | and By Ground or Expand By Center for a or queries its status. | |
| Syntax | :CHANne | <pre>l<x>:EXPand {GND CENTer ?}</x></pre> | |
| Parameter | <x></x> | Channel 1,2,3,4 | |
| | GND | Ground | |
| | CENTer | Center | |
| Return parameter | GND | Expand By Ground | |
| | CENTER | Expand By Center | |
| Example | :CHANnel1:EXPand GND | | |
| | Sets Channel 1 to Expand By Ground. | | |
| | | | |
| :CHANnel <x>:</x> | IMPedan | ce? —Query | |
| Description | Returns the impedance of the oscilloscope. (The impedance of the GDS-2000E is fixed at $1M\Omega$) | | |
| Syntax | :CHANnel <x>:IMPedance?</x> | | |
| Parameter | <x></x> | Channel | |
| | 1/2/3/4 | CH1/2/3/4 | |
| Return parameter | | Returns the impedance value. | |
| Example :CHANnel1:IMPedance? 1.000000E+06 | | | |
| | The impe | edance is 1M ohms. | |
| | | | |



| :CHANnel <x>:</x> | INVert | | Set ———————————————————————————————————— |
|-------------------|--|-------------------------------|--|
| Description | Inverts a channel or returns its st | | tatus. |
| Syntax | :CHANr | el <x>:INVert {OFF ON </x> | ?} |
| Parameter | <x></x> | Channel 1, 2, 3, 4 | |
| | OFF | Invert off | |
| | ON | Invert on | |
| Return parameter | ON | Invert on | |
| | OFF | Invert off | |
| Example | :CHANr | el1:INVert ON | |
| | Inverts | Channel 1 | |
| | | | Set → |
| :CHANnel <x>:</x> | POSitio | n | → Query |
| Description | Sets or returns the position level for a channel. | | |
| Note | The vertical position will only be set to closest allowed value. The position level range depends on the vertical scale. | | |
| | The scale | le must first be set before | e the position can |
| Syntax | :CHANr | $el:POSition { < NRf> $ | ?} |
| Parameter | <x></x> | Channel 1, 2, 3, 4 | |
| | <nrf></nrf> | Position. Range depend scale. | s on the vertical |
| Return parameter | <nr3></nr3> | Returns the position val | ue. |
| Example 1 | :CHANr | el1:POSition 2.4E–3 | |
| | Sets the | Channel 1 position to 2. | 4mV/mA |
| Example 2 | :CHANr | el1:POSition? | |
| | 2.4E-3 | | |
| | Returns | 2.4mV as the vertical po | sition. |



| :CHANnel <x>:</x> | PROBe: | RATio | Set → Query | |
|---------------------|--|--|--------------------|--|
| Description | Sets or r | Sets or returns the probe attenuation factor. | | |
| Syntax | :CHANn | el <x>:PROBe:RATio { <ni< td=""><td>Rf> ?}</td></ni<></x> | Rf> ?} | |
| Related Commands | :CHANn | el <x>:PROBe:TYPe</x> | | |
| Parameter | <x></x> | Channel 1, 2, 3, 4 | | |
| | <nrf></nrf> | Probe attenuation factor | | |
| Return parameter | <nr3></nr3> | Returns the probe factor | | |
| Example | :CHANn | el1:PROBe:RATio 1.00E+0 | | |
| | Sets the | Channel 1 probe attenua | ation factor to 1x | |
| | | | Set → | |
| :CHANnel <x>:</x> | PROBe: | TYPe | → Query | |
| Description | Sets or returns the probe type (voltage/current). | | | |
| Syntax | :CHANnel <x>:PROBe:TYPe { VOLTage CURRent ?}</x> | | | |
| Related Commands | :CHANnel <x>:PROBe:RATio</x> | | | |
| Parameter | <x></x> | Channel 1, 2, 3, 4 | | |
| | VOLTage | Voltage | | |
| | CURRen | t Current | | |
| Return parameter | Returns | the probe type. | | |
| Example | :CHANn | el1:PROBe:TYPe VOLTage | | |
| | Sets the Channel 1 probe type to voltage. | | | |
| | | | Set → | |
| :CHANnel <x>:</x> | SCALe | | → Query | |
| Description | | eturns the vertical scale. robe attenuation factor. | The scale depends | |
| | Note the before the | e probe attenuation factone scale. | r should be set | |



| Syntax | :CHANnel <x>:SCALe { <nrf> ?}</nrf></x> | | |
|------------------|---|--|--|
| Parameter | <x></x> | Channel 1, 2, 3, 4 | |
| | <nrf></nrf> | Vertical scale: 2e–3 to 1e+1 | |
| | | 2mV to 10V (Probe x1) | |
| Return parameter | <nr3></nr3> | Returns the vertical scale in volts or amps. | |
| Example | :CHANnel1:SCAle 2.00E–2 | | |
| | Sets the Channel 1 vertical scale to 20mV/div | | |

Math Commands

| :MATH:DISP | | → Query | |
|------------|--|---------|---|
| | | Set | |
| | .WATTIADValiceu.3CALE | 0 | J |
| | :MATH:ADVanced:POSition | | |
| | :MATH:ADVanced:POSition | | |
| | :MATHVAR:: | | |
| | :MATHVAR? | | |
| | :MATH:DEFine | | |
| | :MATH:FFT:HORizontal:SCALe | | |
| | :MATH:FFT:SCALe | | |
| | :MATH:FFT:POSition: :MATH:FFT:SCALe | | |
| | :MATH:FFT:WINDow | | |
| | :MATH:FFT:MAG | | |
| | :MATH:FFT:SOURce | | |
| | :MATH:DUAL:SCALe | | |
| | :MATH:DUAL:POSition | | |
| | :MATH:DUAL:OPERator | | |
| | :MATH:DUAL:SOURce <x></x> | | |
| | :MATH:TYPe | | |
| | :MATH:DISP | 5 | 3 |
| | | | |

Description Turns the math display on or off on the screen. Syntax :MATH:DISP {OFF|ON|?} Parameter/ OFF Math is not displayed on screen Return parameter ON Math is displayed on screen Example :MATH:DISP OFF

Math is off.



| :MATH:TYPe | | Set → Query | | |
|---------------------|---|--|--|--|
| Description | - | Queries or sets the Math type to FFT, Advanced Math or to dual channel math operations | | |
| Syntax | :MATH:TYPe | e { DUAL ADVanced FFT ? } | | |
| Related Commands | :MATH:DISF | D | | |
| Parameter | DUAL | Dual channel operations | | |
| | ADVanced | Advanced math operations | | |
| | FFT | FFT operations | | |
| Return parameter | Returns the | math type. | | |
| Example | :MATH:TYPe | e DUAL | | |
| | Sets the Ma | th type to dual channel math operation. | | |
| | | Set | | |
| :MATH:DUAL: | SOURce <x:< td=""><td>⇒ Query</td></x:<> | ⇒ Query | | |
| Description | Sets the dual math source for source 1 or 2. | | | |
| Syntax | :MATH:DUAL:SOURce <x> { CH1 CH2 CH3 CH4 REF1 REF2 REF3 REF4 ? }</x> | | | |
| Parameter | <x> Sou</x> | arce number 1 or 2 | | |
| | CH1~4 Cha | annel 1 to 4 | | |
| | REF1~4 Ref | ference waveforms 1 to 4 | | |
| Return parameter | Returns the | source for the source 1 or 2. | | |
| Example | :MATH:DUAL:SOURce1 CH1 | | | |
| | Sets source1 | 1 as channel 1. | | |
| :MATH:DUAL: | OPERator | Set → Query | | |
| Description | Sets the mat | th operator for dual math operations. | | |



| Syntax | :MATH:DUAL:OPERator {PLUS MINUS MUL $DIV $?} | | |
|------------------|---|--|--|
| Parameter | PLUS | + operator | |
| | MINUS | - operator | |
| | MUL | × operator | |
| | DIV | ÷ operator | |
| Return parameter | Returns opera | ator type. | |
| Example | :MATH:DUAL | :OPERator PLUS | |
| | Sets the math | n operator as plus (+). | |
| | | <u>Set</u> → | |
| :MATH:DUAL: | POSition | → Query | |
| Description | Sets the vertical position of the displayed math result expressed by unit/division. | | |
| Syntax | :MATH:DUAL:POSition { <nrf> ? }</nrf> | | |
| Parameter | <nrf></nrf> | Vertical position | |
| | | Depends on the vertical scale (Unit/Div) | |
| Return parameter | <nr3></nr3> | Returns the vertical position. | |
| Example | :MATH:DUAL | :POSition 1.0E+0 | |
| | Sets the verti | cal position to 1.00 unit/div. | |
| | | <u>Set</u> → | |
| :MATH:DUAL: | SCALe | → Query | |
| Description | Sets the vertical scale of the displayed math result. | | |
| Syntax | :MATH:DUAL:SCALe { <nrf> ?}</nrf> | | |
| Parameter | <nrf></nrf> | Vertical scale | |
| Return parameter | <nr3></nr3> | Returns the scale. | |
| Example | :MATH:DUAL:SCALe 2.0E-3 | | |
| | Sets the vertical scale to 2mV/2mA. | | |



| | | | Set → |
|------------------|--|-----------------------------------|------------------|
| :MATH:FFT:SC | URce | | → Query |
| Description | Sets and queries the FFT math source. | | |
| Syntax | :MATH:FFT:S REF2 REF3 | OURce { CH1 CH2 REF4 ? } | CH3 CH4 REF1 |
| Related | :MATH:ADVa | nced:EDIT:SOURce<> | <> |
| commands | :MATH:ADVa | nced:EDIT:OPERator | |
| Parameter | CH1~4 | Channel 1 to 4 | |
| | REF1~4 | Reference waveform | 1 to 4 |
| Return parameter | Returns the F | FT source. | |
| Example | :MATH:FFT:S | OURce CH1 | |
| | Sets the FFT | math source as char | nnel 1. |
| | | | Set → |
| :MATH:FFT:M/ | ٩G | | → Query |
| Description | Sets FFT vertical units as linear or decibels. | | |
| Syntax | :MATH:FFT:M | 1AG {LINEAR DB ? | } |
| Parameter | LINEAR | Linear units (Vrms) | |
| | DB | Logarithmic units (d | В) |
| Return parameter | Returns the F | FT vertical units. | |
| Example | :MATH:FFT:M | MAG DB | |
| | Sets FFT ver | tical units to dB. | |
| | | | Set → |
| :MATH:FFT:WI | NDow | | → Query |
| Description | Sets the windowing filter used for the FFT function. | | |
| Syntax | :MATH:FFT:WINDow {RECTangular HAMming HANning BLAckman ?} | | |
| Parameter | RECTangular Rectangular window | | |



Example

| | HAMming | Hamming window |
|------------------|---|--|
| | HANning | Hanning window |
| | BLAckman | Blackman window |
| Return parameter | Returns the F | FT window. |
| Example | :MATH:FFT:W | /INDow HAMming |
| | Sets the FFT | window filter to hamming. |
| | | Set |
| :MATH:FFT:PC | Sition | → Query |
| Description | Sets the vertical position of the displayed FFT result. | |
| Syntax | MATH:FFT:POSition { <nrf> ? }</nrf> | |
| Parameter | <nrf></nrf> | Vertical position: -12e+0 to +12e+0 (12 units/division to +12 units/division.) |
| Return parameter | <nr3></nr3> | Returns the vertical position. |
| Example | :MATH:FFT:POSition -2e-1 | |
| | Sets the FFT position to -0.2 divisions. | |
| | | <u>Set</u> → |
| :MATH:FFT:SC | ALe | → Query |
| Description | Sets the vertical scale of the displayed FFT result. | |
| Syntax | :MATH:FFT:SCALe { <nrf> ?}</nrf> | |
| Parameter | <nrf></nrf> | Vertical scale: |
| | | Linear: 2e-3 to 1e+3 (2mV~1kV) |
| | | dB: 1e+0 to 2e+1 (1~20dB) |
| Return parameter | <nr3></nr3> | Returns vertical scale. |
| | | |

:MATH:FFT:SCAle 1.0e+0 Sets the scale to 1dB.



| | | | Set → | |
|---------------------------------------|---|---|-----------------------------------|--|
| :MATH:FFT:HO | ORizontal:S0 | CALe | → Query | |
| Description | Sets or queries the zoom scale for FFT math. | | | |
| Syntax | :MATH:FFT:H | :MATH:FFT:HORizonatal:SCALe { <nrf> ?}</nrf> | | |
| Parameter | <nrf></nrf> | Zoom scale: 1 to 20 | times | |
| Return parameter | <nr3></nr3> | Returns zoom scale. | | |
| Example | :MATH:FFT:HORizontal:SCALe 5 | | | |
| | Sets the zoor | m scale to 5X. | | |
| | | | <u>Set</u> → | |
| :MATH:FFT:HORizontal:POSition → Query | | | | |
| Description | Sets the horizontal position of the displayed FFT result. | | | |
| Syntax | MATH:FFT:HORizontal:POSition { <nrf> ? }</nrf> | | | |
| Parameter | <nrf></nrf> | Horizontal position: | 0Hz ~ 999.9kHz | |
| Return parameter | <nr3></nr3> | Returns the vertical p | oosition. | |
| Example | :MATH:FFT:HORizontal:POSition 6.0e5 | | | |
| | Sets the FFT horizontal position to 600kHz. | | | |
| | <u>Set</u> → | | Set → | |
| :MATH:DEFine | | | → Query | |
| Description | Sets or queries the advanced math expression as a string. | | | |
| Syntax | :MATH:DEFine { <string> ?}</string> | | | |
| Related | :MATH:DISP :MATH:TYPe | | | |
| Parameter | <string></string> | An expression enclose quotes. Note, ensure used correctly in the expression can contaparts: | e parentheses are expression. The | |



| | Source | CH1~CH4, Ref1~Ref4 |
|--|------------------|--|
| | Function | Intg(, Diff(, log(, ln(, Exp(, Sqrt(, Abs(, Rad(, Deg(, sin(, cos(, tan(, asin(, acos(, atan(|
| | Variable | VAR1, VAR2 |
| | Operator | +, -, *, /, (,), !(, <, >, <=, >=, ==, !=, , && |
| | Figure | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ., E |
| | Measure- ment | Pk-Pk(, Max(, Min(, Amp(, High(, Low(, Mean(, CycleMean(, RMS(, CycleRMS(, Area(, CycleArea(, ROVShoot(, FOVShoot(, Freq(, Period(, Rise(, Fall(, PosWidth(, NegWidth(, Dutycycle(, FRR(, FRF(, FFR(, FFF(, LRR(, LRF(, LFR(, LFF(, Phase(|

Return parameter Returns the expression as a string.

Example :MATH:DISP ON

:MATH:TYPe ADVanced MATH:DEFine "CH1-CH2"

Sets the math expression to CH1-CH2.

:MATHVAR?



| Description | Returns the value of the VAR1 and VAR2 variables. | | |
|---------------------|--|--|--|
| Syntax | MATHVAR? | | |
| Related Commands | MATHVAR:VAR <x> MATH:DEFine</x> | | |
| Return parameter | <string> VAR1 <nr3>: VAR2 <nr3></nr3></nr3></string> | | |



Example MATHVAR?

VAR1 1.000000E+06; VAR2 1.0E+1

Returns the value of both variables.

:MATHVAR:VAR<X>



| Syntax MATHVAR:VAR <x> {<nrf> ?} Related MATH:DEFine Commands</nrf></x> | Description |
|--|------------------|
| | Syntax |
| | |
| Parameter <x> 1, 2 (VAR1 or VAR2)</x> | Parameter |
| <nrf> Value of VAR1/VAR2</nrf> | |
| Return parameter <nr3> Returns the value of VAR1/VAR2</nr3> | Return parameter |

Example :MATH:VAR1 6.0e4

Sets VAR1 to 60000.

:MATH:ADVanced:POSition



| Description | Sets the vertical position of the advanced math result, expressed in unit/div. | |
|------------------|--|---|
| Syntax | MATH:ADVanced:POSition { <nrf> ? }</nrf> | |
| Parameter | <nrf></nrf> | Vertical position: -12e+0 to +12e+0 (12 units/division to +12 units/division.) |
| Return parameter | <nr3></nr3> | Returns the vertical position. |
| Example | :MATH:ADVanced:POSition 1.0e+0 | |

Sets the position as 1.00 unit/div.

:MATH:ADVanced:SCALe



| Description | Sets or queries the vertical scale the advanced math result. |
|-------------|--|
| Syntax | :MATH:ADVanced:SCALe { <nrf> ?}</nrf> |



| Parameter | <nrf></nrf> | Vertical scale |
|------------------|-------------------------------------|-----------------------------|
| Return parameter | <nr3></nr3> | Returns the vertical scale. |
| Example | :MATH:ADVanced:SCALe 2.0E-3 | |
| | Sets the vertical scale to 2mV/Div. | |



Cursor Commands

| :CURSor:MODe | 63 |
|---|----|
| :CURSor:SOURce | 63 |
| :CURSor:HUNI | 64 |
| :CURSor:HUSE | 64 |
| :CURSor:VUNI | 65 |
| :CURSor:VUSE | 65 |
| :CURSor:DDT | 65 |
| :CURSor:H1Position | 66 |
| :CURSor:H2Position | 66 |
| :CURSor:HDELta | 67 |
| :CURSor:V1 Position | 67 |
| :CURSor:V2Position | 67 |
| :CURSor:VDELta | 68 |
| : CURSor: XY: RECT angular: X: POSition < X > | 68 |
| :CURSor:XY:RECTangular:X:DELta | 68 |
| : CURSor: XY: RECT angular: Y: POSition < X > | 69 |
| :CURSor:XY:RECTangular:Y:DELta | 69 |
| : CURSor: XY: POLar: RADIUS: POSition < X > | 69 |
| :CURSor:XY:POLar:RADIUS:DELta | 70 |
| : CURSor: XY: POLar: THETA: POSition < X > | 70 |
| :CURSor:XY:POLar:THETA:DELta | 70 |
| : CURSor: XY: PRODuct: POSition < X > | 71 |
| :CURSor:XY:PRODuct:DELta | 71 |
| :CURSor:XY:RATio:POSition <x></x> | 71 |
| :CURSor:XY:RATio:DELta | 72 |



| :CURSor:MOD | e | Set → Query |
|------------------|---|---|
| Description | Sets cursor mode to horizontal (H) or horizontal and vertical (HV). | |
| | Note: When the cursor source is set to bus, then only the horizontal cursor is available. | |
| Syntax | :CURSor:MODe {OFF H HV ? } | |
| Parameter | OFF | Turns the cursors off. |
| | Н | Turns the horizontal cursors on. |
| | HV | Turns horizontal and vertical cursors on. |
| Return parameter | Returns the state of the cursors (H, HV, OFF). | |
| Example | :CURSor:MODe OFF | |
| | Turns the cursors off. | |
| | | <u>Set</u> → |
| :CURSor:SOUR | ce | → Query |
| Description | Sets or queries the cursor source. | |
| Syntax | :CURSor:SOURce {CH1 CH2 CH3 CH4 REF1 REF2 REF3 REF4 MATH BUS1 ?} | |
| Parameter | CH1~CH4 | Channel 1 to 4 |
| | REF1~4 | Reference waveform 1 to 4 |
| | MATH | Math source |
| | BUS1 | Bus source |
| Return parameter | Returns the cursor source. | |
| Example | :CURSor:SOURce CH1 | |

Turns the cursor source as channel 1.



| :CURSor:HUN | I | Set → Query |
|---------------------|---|--|
| Description | Sets or queries the units for the horizontal bar cursors. | |
| Syntax | :CURSor:HUNI {SEConds HERtz DEGrees PERcent ?} | |
| Related Commands | :CURSor:MODe | |
| Parameter | SEConds | Sets the cursor units to time in seconds. |
| | HERtz | Sets the cursor units to frequency. |
| | DEGrees | Sets the cursor units to degrees. |
| | PERcent | Sets the cursor units to percent. |
| Return parameter | Returns the unit type. | |
| Example | :CURSor:HUNI SEConds | |
| | Sets the units to time in seconds. | |
| | | |
| :CURSor:HUSE | - - - | Set → |
| Description | | ent cursor position as the phase or ce for the Percent or Degrees cursors. |
| Note | This command can only be used when :CURSor:HUNI is set to DEGrees or PERcent. | |

:CURSor:HUSE {CURRent}

:CURSor:HUSE CURRent.

CURRent Uses the current horizontal position

:CURSor:MODe

:CURSor:HUNI

Syntax

Related

Commands

Parameter

Example



| :CURSor:VUNI | | Set → Query | |
|---------------------------------------|--|---|--|
| Description | Sets or queri | Sets or queries the units for the vertical bar cursors. | |
| Syntax | :CURSor:VUI | NI {BASE PERcent ?} | |
| Related Commands | :CURSor:MODe | | |
| Parameter | BASE | Sets the vertical cursor units the same as the scope units (V or A). | |
| | PERcent | Sets the displayed units to percent. | |
| Return parameter | Returns the u | Returns the unit type. | |
| Example | :CURSor:VUI | NI BASE | |
| | Sets the unit | ts to the base units. | |
| :CURSor:VUSE | | Set → | |
| Description | | rent cursor position as the ratio r the Percent (vertical) cursors. | |
| Note | This command can only be used when :CURSor:VUNI is set to PERcent. | | |
| Syntax | :CURSor:VUSE {CURRent} | | |
| Related Commands | :CURSor:MODe :CURSor:VUNI | | |
| Parameter | CURRent | Uses the current vertical position | |
| Example | :CURSor:VUS | | |
| :CURSor:DDT | | → Query | |
| Description | function is o | deltaY/deltaT (dy/dT) readout. This only supported if the source channels Ref1~4 or Math. | |
| Syntax | :CURSor:DD | Τ{?} | |
| · · · · · · · · · · · · · · · · · · · | | | |



| Related Commands | :CURSor:MODe | |
|---------------------------|---|--|
| Return Parameter | <nr3></nr3> | Returns the readout in <nr3> format.</nr3> |
| Example | :CURSor:DDT? | |
| | 4.00E-05 | |
| | | Set → |
| :CURSor:H1Position —Query | | Query |
| Description | Sets or returns the first horizontal cursor (H1) position. | |
| Syntax | :CURSor:H1Position { <nrf> ?}</nrf> | |
| Related Commands | :CURSor:H2Position | |
| Parameter | <nrf></nrf> | Horizontal position |
| Return parameter | Returns the cursor position. | |
| Example | :CURSor:H1Position? | |
| | -1.34E-3 | |
| | Returns the H1 cursor position as -1.34ms. | |
| | | Set → |
| :CURSor:H2Position | | → Query |
| Description | Sets or returns the second horizontal cursor (H2) position. | |
| Syntax | :CURSor:H2Position { <nrf> ?}</nrf> | |
| Related Commands | :CURSor:H1Position | |
| Parameter | <nrf></nrf> | Horizontal Position |
| Return parameter | Returns the cursor position. | |

:CURSor:H2Position 1.5E-3

Sets the H2 cursor position to 1.5ms.

Example



| :CURSor:HDEL | ta | → Query |
|----------------------------|---|--|
| Description | Returns the delta of H1 and H2. | |
| Syntax | :CURSor:HDELta{?} | |
| Return Parameter | <nr3></nr3> | Returns the distance between two horizontal cursors. |
| Example | :CURSor:HDI | ELta? |
| | 5.0E-9 | |
| | Returns the | horizontal delta as 5ns. |
| | | <u>Set</u> → |
| :CURSor:V1Position → Query | | |
| Description | Sets the first vertical cursor (V1) position. | |
| Syntax | :CURSor:V1Position { <nrf> ?}</nrf> | |
| Parameter | <nrf></nrf> | Vertical position. Depends on the vertical scale. |
| Return parameter | <nr3></nr3> | Returns the cursor position. |
| Example | :CURSor:V1P | osition 1.6E -1 |
| | Sets the V1 cursor position to 160mA. | |
| | | Set |
| :CURSor:V2Position —Query | | → Query |
| Description | Sets the first | vertical cursor (V2) position. |
| Syntax | :CURSor:V2Position { <nrf> ?}</nrf> | |
| Parameter | <nrf></nrf> | Vertical position. Depends on the vertical scale. |
| Return parameter | <nr3></nr3> | Returns the cursor position. |
| Example | :CURSor:V2Position 1.1E-1 | |

Sets the V2 cursor position to 110mA.



:CURSor:VDELta (Query Returns the delta of V1 and V2. Description Syntax :CURSor:VDELta{?} Returns the difference between two Return Parameter < NR3> vertical cursors. Example :CURSor:VDELta? 4.00F + 0Returns the vertical delta as 4 volts. Set) (Query) :CURSor:XY:RECTangular:X:POSition<X> Sets or queries the horizontal position in XY mode Description for the X rectangular coordinates for cursor 1 or 2. :CURSor:XY:RECTangular:X:POSition<X> {<NRf>|?} Syntax Parameter <X> Cursor 1, 2 <NRf> Horizontal position co-ordinates Return parameter <NR3> Returns the cursor position. Example :CURSor:XY:RECTangular:X:POSition1 4.0E-3 Sets the X-coordinate cursor 1 position to 40 mV/mV. :CURSor:XY:RECTangular:X:DELta Query Returns the delta value of cursor 1 and 2 on the X Description coordinate. Syntax :CURSor:XY:RECTangular:X:DELta{?} Returns the delta value of cursor 1 and 2 Return Parameter < NR3>

as <NR3>.



| Example | :CURSor:XY:RECTangular:X:DELta? | | |
|--------------------|---|---|--|
| | 80.0E-3 | | |
| | Returns the horizontal delta as 80mV. | | |
| | | Set → | |
| :CURSor:XY:RE | CTangul | ar:Y:POSition <x> → Query</x> | |
| Description | Sets or queries the vertical position in XY mode for the Y rectangular coordinates for cursor 1 or 2. | | |
| Syntax | :CURSor: | (Y:RECTangular:Y:POSition <x> {<nrf> ?}</nrf></x> | |
| Parameter | <x></x> | Cursor 1, 2 | |
| | <nrf></nrf> | Vertical position co-ordinates | |
| Return parameter | <nr3></nr3> | Returns the cursor position. | |
| Example | :CURSor: | (Y:RECTangular:Y:POSition1 4.0E-3 | |
| | Sets the Y 40mV/m | Y-coordinate cursor 1 position to V. | |
| :CURSor:XY:RE | CTangula | ar:Y:DELta → Query | |
| Description | Returns the delta value of cursor 1 and 2 on the Y coordinate. | | |
| Syntax | :CURSor:XY:RECTangular:Y:DELta{?} | | |
| Return Parameter | <nr3></nr3> | Returns the delta value of cursor 1 and 2 as <nr3>.</nr3> | |
| Example :CURSor:XY | | ······································ | |
| | 80.0E-3 | | |
| | Returns t | he horizontal delta as 80mV. | |
| | | | |
| :CURSor:XY:PC | Lar:RAD | IUS:POSition <x> → Query</x> | |
| Description | Queries the polar radius position for the specified cursor in XY mode, where X can be either cursor 1 or 2. | | |



| Syntax | :CURSor:XY:POLar:RADIUS:POSition <x>{?}</x> | |
|------------------|--|------------------------------------|
| Parameter | <x></x> | 1, 2 (cursor 1, cursor 2) |
| Return Parameter | <nr3></nr3> | Returns the polar radius position. |
| Example | :CURSor:XY:POLar:RADIUS:POSition1? | |
| | 80.0E-3 | |
| | Returns the polar radius position as 80.0mV. | |

:CURSor:XY:POLar:RADIUS:DELta



| Description | Returns the radius delta value of cursor 1 and 2. | | |
|------------------|---|---------------------------|--|
| Syntax | :CURSor:XY:POLar:RADIUS:DELta{?} | | |
| Return Parameter | <nr3></nr3> | Returns the radius delta. | |
| Example | :CURSor:XY:POLar:RADIUS:DELta? | | |
| | 31.4E-3 | | |
| | Returns the radius delta as 31.4mV. | | |

:CURSor:XY:POLar:THETA:POSition<X>



| Description | Queries the polar angle for the specified cursor in XY mode, where X can be either 1 or 2. | | |
|------------------|--|---------------------------|--|
| Syntax | :CURSor:XY:POLar:THETA:POSition <x>{?}</x> | | |
| Parameter | <x></x> | 1, 2 (Cursor 1, Cursor 2) | |
| Return parameter | <nr3></nr3> | Returns the polar angle. | |
| Example | :CURSor:XY:POLAR:RADIUS:POSition1? | | |
| | 8.91E+1 | | |
| | Returns the polar angle for cursor1 as 89.1°. | | |

:CURSor:XY:POLar:THETA:DELta



| Description | Queries the polar angle delta between cursor1 and |
|-------------|---|
| | cursor2. |



| Syntax | :CURSor: | (Y:POLar:THETA:DELta{?} | |
|------------------|--|---|--|
| Return parameter | <nr3></nr3> | Returns the theta delta between cursor1 and cursor2. | |
| Example | :CURSor: | (Y:POLar:THETA:DELta? | |
| | 9.10E+0 | | |
| | Returns t | he delta as 9.1°. | |
| :CURSor:XY:PR | ODuct:P | OSition <x> → Query</x> | |
| Description | Queries the product in XY mode for the specified cursor, where x can be either 1 or 2. | | |
| Syntax | :CURSor:XY:PRODuct:POSition <x>{?}</x> | | |
| Parameter | <x></x> | 1, 2 (Cursor 1, Cursor 2) | |
| Return parameter | <nr3></nr3> | Returns the product value of the Cursor1 or Cursor2. | |
| Example | :CURSor: | (Y:PRODuct:POSition1? | |
| | 9.44E-5 | | |
| | Returns t | he product of cursor1 as 94.4uVV. | |
| :CURSor:XY:PR | ODuct:D | PELta → Query | |
| Description | Queries t | he product delta in XY mode. | |
| Syntax | :CURSor: | <pre><!-- Comparison of the co</td--></pre> | |
| Return parameter | <nr3></nr3> | Returns the product delta. | |
| Example | :CURSor: | Y:PRODuct:DELta? | |
| | 1.22E-5 | | |
| | Returns t | he product delta as 12.2uVV. | |
| :CURSor:XY:RA | Tio:POS | ition <x> → Query</x> | |
| Description | | he ratio in XY mode for the specified here x can be either cursor 1 or 2. | |



| Syntax | :CURSor:XY:RATio:POSition <x>{?}</x> | |
|------------------|--------------------------------------|---------------------------|
| Parameter | <x></x> | 1, 2 (Cursor 1, Cursor 2) |
| Return parameter | <nr3></nr3> | Returns the ratio. |
| Example | :CURSor:XY:RATio:POSition? | |
| | 6.717E+1 | |
| | Returns the ratio value as 6.717V/V. | |

:CURSor:XY:RATio:DELta



| Description | Queries the ratio delta in XY mode. | | |
|------------------|-------------------------------------|--------------------------|--|
| Syntax | :CURSor:XY:RATio:DELta{?} | | |
| Return parameter | <nr3></nr3> | Returns the ratio delta. | |
| Example | :CURSor:XY:RATio:DELta? | | |
| | 5.39E+1 | | |
| | Returns the ratio delta as 53.9V/V. | | |

Display Commands

| = 15 p 14 / Co. | | |
|------------------|--|----------------------------|
| | :DISPlay:INTensity:WAVEform :DISPlay:INTensity:GRATicule :DISPlay:INTensity:BACKLight :DISPlay:INTensity:BACKLight:AUTODim:ENAble :DISPlay:INTENSITy:BACKLight:AUTODim:TIMe .: :DISPlay:PERSistence :DISPlay:GRATicule :DISPlay:WAVEform :DISPlay:OUTPut | 73 74 74 75 75 |
| :DISPlay:INTer | Set → usity:WAVEform —Query | |
| Description | Sets or queries the waveform intensity level. | |
| Syntax | :DISPlay:INTensity:WAVEform { <nrf> ?}</nrf> | |
| Parameter | <nrf> 0.0E+0 to 1.0E+2 (0% to 100%)</nrf> | |
| Return Parameter | <nr3> Returns the intensity.</nr3> | |
| Example: | :DISPlay:INTensity:WAVEform 5.0E+1 Sets the waveform intensity to 50%. Set Set Query | |
| Description | Sets or queries the graticule intensity level. | |
| Syntax | :DISPlay:INTensity:GRATicule { <nrf> ?}</nrf> | |
| Parameter | <nrf> 1.0E+0 to 1.0E+2 (10% to 100%)</nrf> | |
| Return Parameter | <nr3> Returns the graticule intensity.</nr3> | |
| Example | :DISPlay:INTensity:GRATicule 5.0E+1 | |
| | Sets the graticule intensity to 50%. | |



| :DISPlay:INTen | sity:BAC | KLight | Set → Query |
|--------------------------------|---------------------------------|---|-----------------|
| Description | Sets or qu display. | ueries the intensity of th | ne backlight |
| Syntax | :DISPlay:I | NTensity:BACKLight { <n< td=""><td>IRf> ?}</td></n<> | IRf> ?} |
| Parameter | <nrf></nrf> | 1.0E+0 to 1.0E+2 (10% t | to 100%) |
| Return Parameter | <nr3></nr3> | Returns the backlight int | tensity. |
| Example | :DISPlay:I | NTensity:BACKLight 5.0E | E+1 |
| | Sets the b | oacklight intensity to 50 | 1%. |
| :DISPlay:INTen :ENAble | sity:BAC | KLight:AUTODim | Set → Query |
| Description | Sets or qu | ueries the display auto- | dim function. |
| Syntax | :DISPlay:I {OFF OI | NTensity:BACKLight:AU1 N ?} | ГОDim:ENAble |
| Parameter/ | OFF | Turn auto-dim on. | |
| Return parameter | ON | Turn auto-dim off. | |
| Example | :DISPlay:I | NTensity:BACKLight:AUT | ΓODim:ENAble ON |
| | Turns the auto-dim function on. | | |
| :DISPlay:INTEN :TIMe | NSITy:BA | CKLight:AUTODim | Set → Query |
| Description | Sets or qu | ueries the display auto- | dim time. |
| Syntax | :DISPlay:I { <nr1></nr1> | NTensity:BACKLight:AUT ? } | ГОDim:TIMe |
| Parameter/ Return parameter | <nr1></nr1> | 1 ~ 180 minutes. Time | in minutes. |
| Example | :DISPlay:I | NTensity:BACKLight:AU1 | ΓODim:TIMe 10 |

Sets the auto-dim time to 10 minutes.



| :DISPlay:PERSi | stence | Set → Query | |
|------------------|-----------------------------|--|--|
| Description | Sets or qu | eries the waveform persistence level. | |
| Syntax | :DISPlay:P | ERSistence { INFInite OFF <nrf> ? }</nrf> | |
| Parameter | <nrf></nrf> | 1.6E-2 ~ 4.0E+0. (16mS to 10S) Range(1.6E-2, 30E-3, 60E-3, 120E-2, 240E-3, 500E-3, 750E-3, 1, 1.5,2,,9.5,10 | |
| | INFInite | Infinite persistence | |
| | OFF | No persistence | |
| Return Parameter | <nr3></nr3> | Returns the persistence time. | |
| | INFInite | Infinite persistence | |
| | OFF | No persistence | |
| Example | :DISPlay:PERSistence 2.0E+0 | | |
| | Sets the p | ersistence to 2 seconds. | |
| | | Set → | |
| :DISPlay:GRAT | icule | → Query | |
| Description | Sets or qu | eries graticule display type. | |
| Syntax | :DISPlay:G | RATicule {FULL GRID CROSs FRAMe ?} | |
| Parameter | FULL | CROSs | |
| | FRAMe | GRID | |
| Return parameter | Returns th | e graticule type. | |
| Example | :DISPlay:G | RATicule FULL | |
| | Sets the g | raticule to . | |



| :DISPlay:WAVE | form | | Set → Query |
|-----------------------|--|--------------------------------------|-------------------|
| Description | Sets or queries whether the waveforms are drawn as vectors or dots. | | eforms are drawn |
| Syntax | :DISPlay:WAV | Eform {VECTor DC | T ?} |
| Parameter | VECTor DOT | Vectors Dots | |
| Return parameter | Returns VECT | OR or DOT. | |
| Example: DISPlay:OUTP | :DISPlay:WAVEform VECTor Sets the waveform to vectors. Put — Query | | |
| Description | Returns the screen image as a 16 bit RGB run length encoded image. | | |
| Syntax | :DISPlay:OUTPut{?} | | |
| Return parameter | eter Format: header+data+LF | | |
| | • | ssuming the image following would be | |
| | #560072<[co | unt] [color] [count] [c | olor] > <lf></lf> |
| | Where #560072 is the header, each [count] and [color] data are 2 bytes and <lf> is a line feed character.</lf> | | |



Hardcopy Commands

| :Н. | ARDcopy:START | 77 |
|-----|-----------------------|----|
| :H. | ARDcopy:MODe | 77 |
| :H. | ARDcopy:PRINTINKSaver | 78 |
| :H. | ARDcopy:SAVEINKSaver | 78 |
| :H. | ARDcopy:SAVEFORMat | 78 |
| :H. | ARDcopy:ASSIGN | 79 |
| | | |

:HARDcopy:START



| Description | Executing the HARDcopy:START command is the equivalent of pressing the Hardcopy key on the front panel. |
|---------------------|---|
| Syntax | :HARDcopy:START |
| Related Commands | :HARDcopy:MODe |
| | :HARDcopy:PRINTINKSaver |
| | :HARDcopy:SAVEINKSaver |
| | :HARDcopy:SAVEFORMat |
| | :HARDcopy:ASSIGN |
| | (Set → |

:HARDcopy:MODe



| Description | Sets or queries whether hardcopy is set to print or save. | | |
|---------------------|---|------------|--|
| Syntax | :HARDcopy:MODe { PRINT SAVE ? } | | |
| Related Commands | :HARDcopy:START | | |
| Parameter | PRINT | Print mode | |
| | SAVE | Save mode | |
| Return parameter | Returns the mode (PRINT/SAVF) | | |



Example :HARDcopy:MODe PRINT

Sets hardcopy to print.

:HARDcopy:PRINTINKSaver



| Description | Sets Ink | Sets Inksaver On or Off for printing. | | |
|---------------------|----------|--|--|--|
| Syntax | :HARDo | :HARDcopy:PRINTINKSaver { OFF ON ? } | | |
| Related Commands | | :HARDcopy:START :HARDcopy:MODe | | |
| Parameter | ON | Inksaver ON | | |
| | OFF | Inksaver OFF | | |

Return parameter Returns the print Ink Saver mode. (ON/OFF)

Example :HARDcopy:PRINTINKSaver ON

Sets Ink Saver to ON for printing.

:HARDcopy:SAVEINKSaver



| Description | Sets Inksaver On or Off for saving screen images. | | |
|---------------------|---|--------------|--|
| Syntax | :HARDcopy:SAVEINKSaver { OFF ON ? } | | |
| Related Commands | :HARDcopy:START :HARDcopy:MODe | | |
| Parameter | ON | Inksaver ON | |
| | OFF | Inksaver OFF | |

Return parameter Returns the screen image Ink Saver mode (ON/OFF).

Example :HARDcopy:SAVEINKSaver ON

Sets Inksaver to ON for saving screen images.

:HARDcopy:SAVEFORMat



| Description | Sets or queries the image save file type. |
|-------------|---|
| Syntax | :HARDcopy:SAVEFORMat { PNG BMP ? } |



| Related Commands | :HARDcopy :HARDcopy | | |
|---------------------|--|-----------------|-------------|
| Parameter | PNG | PNG file format | |
| | ВМР | BMP file format | |
| Return parameter | Returns the image file format (PNG/BMP). | | |
| Example | :HARDcopy:SAVEFORMat PNG | | |
| | Sets the file format to PNG. | | |
| :HARDcopy:AS | SIGN | | Set → Query |
| Description | Sets or queries what file type the hardcopy key has been assigned to save. | | |
| . | LIADD | ACCICAL | |

| Description | Sets or queries what file type the hardcopy key has been assigned to save. | | |
|---------------------|--|-----------------------------------|--|
| Syntax | :HARDcopy:ASSIGN {IMAGe WAVEform SETUp ALL ?} | | |
| Related Commands | :HARDcopy:START :HARDcopy:MODe | | |
| Parameter | IMAGe Save image files. | | |
| | WAVEform | Save waveforms. | |
| | SETUp | Save the panel setup. | |
| | ALL | Save All (image, waveform, setup) | |
| Return parameter | Returns the file type. (IMAGE/WAVEFORM/SETUP/ALL) | | |
| Example | :HARDcopy:ASSIGN IMAGE. | | |

Set the hardcopy key to save image files.



Measure Commands

| :MEASure:GATing | 81 |
|------------------------------------|----|
| :MEASure:SOURce | 81 |
| :MEASure:METHod | 82 |
| :MEASUrement:REFLevel:PERCent:HIGH | 82 |
| :MEASUrement:REFLevel:PERCent:LOW | 83 |
| :MEASUrement:REFLevel:PERCent:MID | 83 |
| :MEASUrement:REFLevel:PERCent:MID2 | 83 |
| :MEASure:FALL | 84 |
| :MEASure:FOVShoot | 84 |
| :MEASure:FPReshoot | 85 |
| :MEASure:FREQuency | 85 |
| :MEASure:NWIDth | 86 |
| :MEASure:PDUTy | 86 |
| :MEASure:PERiod | 87 |
| :MEASure:PWIDth | 87 |
| :MEASure:RISe | 88 |
| :MEASure:ROVShoot | 88 |
| :MEASure:RPReshoot | 89 |
| :MEASure:PPULSE | 90 |
| :MEASure:NPULSE | 90 |
| :MEASure:PEDGE | 91 |
| :MEASure:NEDGE | 91 |
| :MEASure:AMPlitude | 92 |
| :MEASure:MEAN | 92 |
| :MEASure:CMEan | 93 |
| :MEASure:HIGH | 93 |
| :MEASure:LOW | 94 |
| :MEASure:MAX | 95 |
| :MEASure:MIN | 95 |
| :MEASure:PK2PK | 96 |
| :MEASure:RMS | 96 |
| :MFASure:CRMS | 97 |



| | :MEASure:AR | Ea97 | |
|------------------|---|---|--|
| | :MEASure:CARea98 | | |
| | :MEASure:FRRDelay99 | | |
| | :MEASure:FRFDelay99 | | |
| | :MEASure:FF | RDelay100 | |
| | :MEASure:FF | FDelay100 | |
| | :MEASure:LR | RDelay101 | |
| | :MEASure:LR | FDelay102 | |
| | | RDelay102 | |
| | | FDelay103 | |
| | :MEASure:PH | IAse103 | |
| | | | |
| | | Set | |
| :MEASure:GAT | ing | → Query | |
| Description | Sets or queri | es the measurement gating. | |
| Syntax | :MEASure:GATing { OFF SCREen CURSor ? } | | |
| Parameter | OFF | Full record | |
| | SCREen | Gating set to screen width | |
| | CURSor | Gating between cursors | |
| Return parameter | Returns the | gating. (OFF, SCREEN, CURSOR) | |
| Example | :MEASure:GA | Ting OFF | |
| · | Turns gating off (full record). | | |
| | | (Set)→ | |
| :MEASure:SOU | Rce | Query | |
| Description | Sets or queri or source2. | es the measurement source for source1 | |
| Syntax | :MEASure:SC MATH ? } | OURce <x> { CH1 CH2 CH3 CH4 </x> | |
| Parameter | <x></x> | Source1 or source2 | |
| | CH1~CH4 | Channel 1 to 4 | |



| | MATH | Math | |
|------------------|--|---|--|
| Return parameter | Returns the | source (CH1, CH2, CH3, CH4, MATH) | |
| Example | :MEASure:SC | DURce1 CH1 | |
| | Sets source1 | to channel 1. | |
| | | Set → | |
| :MEASure:MET | Hod | → Query | |
| Description | - | es the method used to determine the neasurement values. | |
| Syntax | :MEASure:METHod { AUTo HIStogram MINMax ? } | | |
| Parameter | AUTo | Set to auto. | |
| | HIStogram | Set to the Histogram method. | |
| | MINMax | Set to the Min-Max method. | |
| Return parameter | Returns the measurement method (AUTO, HISTOGRAM, MINMAX) | | |
| Example | :MEASure:METHod: AUTo | | |
| | Set the measurement method to auto. | | |
| :MEASUremen | t:REFLevel:F | Set → PERCent:HIGH → Query | |
| Description | Sets or queri percentage. | es the high reference level as a | |
| Syntax | :MEASUreme | nt:REFLevel:PERCent:HIGH { <nrf> ?}</nrf> | |
| Parameter | <nrf></nrf> | 0 - 100% | |
| Return parameter | Returns the | high reference level | |
| Example | :MEASUreme | nt:REFLevel:PERCent:HIGH 50.1 | |
| | Set the high | reference level to 50.1%. | |



| :MEASUremen | t:REFLevel:PERCent:LOW → Query | | |
|------------------|---|--|--|
| Description | Sets or queries the low reference level as a percentage. | | |
| Syntax | $: MEASUrement: REFLevel: PERCent: LOW \ \{< NRf> \ \ ?\}$ | | |
| Parameter | <nrf> 0 - 100%</nrf> | | |
| Return parameter | Returns the low reference level. | | |
| Example | :MEASUrement:REFLevel:PERCent:LOW 40.1 | | |
| | Set the low reference level to 40.1%. | | |
| | Set → | | |
| :MEASUremen | t:REFLevel:PERCent:MID → Query | | |
| Description | Sets or queries the first mid reference level as a percentage. | | |
| Syntax | :MEASUrement:REFLevel:PERCent:MID { <nrf> ?}</nrf> | | |
| Parameter | <nrf> 0 - 100%</nrf> | | |
| Return parameter | Returns the mid reference level. | | |
| Example | :MEASUrement:REFLevel:PERCent:MID 50 | | |
| | Set the mid reference level to 50%. | | |
| :MEASUremen | t:REFLevel:PERCent:MID2 → Query | | |
| Description | Sets or queries the second mid reference level as a percentage. | | |
| Syntax | :MEASUrement:REFLevel:PERCent:MID2 { <nrf> ?}</nrf> | | |
| Parameter | <nrf> 0 - 100%</nrf> | | |
| Return parameter | Returns the mid reference level of the second source. | | |
| Example | :MEASUrement:REFLevel:PERCent:MID2 50 | | |
| | Set the mid reference level to 50%. | | |



| :MEASure:FALI | L | → Query | |
|---------------------|---|--|--|
| Description | Returns the fall time measurement result. | | |
| Syntax | :MEASure:FALL{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:FALL? | | |
| | Selects Channel 1 as the source, and then measur the fall time. | | |

:MEASure:FOVShoot



| Description | Returns the fall overshoot amplitude. | | |
|---------------------|---|--|--|
| Syntax | :MEASure:FOVShoot{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the fall overshoot as a percentage | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |



Example :MEASure:SOURce1 CH1

:MEASure:FOVShoot?

1.27E+0

Selects Channel 1, and then measures the fall

overshoot.

:MEASure:FPReshoot



| Description | Returns fall preshoot amplitude. | | |
|---------------------|---|--|--|
| Syntax | :MEASure:FPReshoot{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Returns | Returns the fall preshoot as <nr3>.</nr3> | | |
| Return parameter | <nr3></nr3> | Returns the fall preshoot as a percentage. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:FPReshoot? | | |
| | Selects Charpreshoot. | nnel 1, and then measures the fall | |

:MEASure:FREQuency



| Description | Returns the frequency value. | | |
|---------------------|------------------------------|--|--|
| Syntax | :MEASure:FREQuency{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the frequency in Hz. | |
| | Chan Off | Indicates the source channel is not activated. | |



| Note | Before using this command, select the measurement channel. See the example below. |
|---------|---|
| Example | :MEASure:SOURce1 CH1 :MEASure:FREQuency? >1.0E+3 |
| | Selects Channel 1, and then measures the frequency. |

:MEASure:NWIDth



| Description | Returns the first negative pulse width timing. | | | |
|---------------------|---|--|--|--|
| Syntax | :MEASure:N | :MEASure:NWIDth{?} | | |
| Related Commands | :MEASure:SO | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the negative pulse width in seconds. | | |
| | Chan Off | Indicates the source channel is not activated. | | |
| Note | Before using this command, select the measurement channel. See the example below. | | | |
| Example | :MEASure:SOURce1 CH1 | | | |
| | :MEASure:NWIDth? | | | |
| | 4.995E-04 | | | |
| | Selects Channel 1, and then measures the negative pulse width. | | | |
| | | | | |

: MEASure : PDUTy



| Description | Returns the positive duty cycle ratio as percentage. | | |
|------------------|--|----------------------------------|--|
| Syntax | :MEASure:PDUTy{?} | | |
| Related commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the positive duty ratio. | |



| | Chan Off | Indicates the source channel is not activated. |
|---------|--|---|
| Note | | g this command, select the nt channel. See the example below. |
| Example | :MEASure:SOURce1 CH1 :MEASure:PDUTy? 5.000E+01 Selects Channel 1, and then measures the positive duty cycle. | |

:MEASure:PERiod



| Description | Returns the period. | | |
|---------------------|---|--|--|
| Syntax | :MEASure:Pl | :MEASure:PERiod{?} | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3> Returns the period.</nr3> | | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:PERiod? | | |
| | 1.0E-3 | | |
| | Selects Channel 1, and then measures the period. | | |

:MEASure:PWIDth



| Description | Returns the first positive pulse width. |
|---------------------|---|
| Syntax | :MEASure:PWIDth{?} |
| Related Commands | :MEASure:SOURce <x></x> |



| Return parameter | <nr3></nr3> | Returns the positive pulse width. |
|------------------|-----------------------------|---|
| | Chan Off | Indicates the source channel is not activated. |
| Note | , | g this command, select the nt channel. See the example below. |
| Example | :MEASure:SOURce1 CH1 | |
| | :MEASure:PWIDth? | |
| | 5.0E-6 | |
| | Selects Char pulse width | nnel 1, and then measures the positive |

:MEASure:RISe



| Description | Returns the first pulse rise time. | |
|---------------------|---|--|
| Syntax | :MEASure:RISe{?} | |
| Related Commands | :MEASure:SOURce <x></x> | |
| Return parameter | <nr3></nr3> | Returns the rise time. |
| | Chan Off | Indicates the source channel is not activated. |
| Note | Before using this command, select the measurement channel. See the example below. | |
| Example | :MEASure:SOURce1 CH1 | |
| | :MEASure:RISe? | |
| | 8.5E-6 | |
| | Selects Channel 1, and then measures the rise time. | |

:MEASure:ROVShoot



| Description | Returns the rising overshoot over the entire waveform in percentage. | |
|-------------|--|--|
| Syntax | :MEASure:ROVShoot{?} | |



| Related Commands | :MEASure:SOURce <x></x> | | |
|---------------------|---|--|--|
| Return parameter | <nr3></nr3> | Returns the overshoot. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:ROVShoot? | | |
| | 5.00E+00 | | |
| | Selects Channel 1, and then measures the rise overshoot. | | |

:MEASure:RPReshoot



| Description | Returns rising preshoot over the entire waveform in percentage. | | |
|---------------------|---|--|--|
| Syntax | :MEASure:RF | :MEASure:RPReshoot{?} | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3> Returns the rising preshoot.</nr3> | | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:RPReshoot? | | |
| | 2.13E-2 | | |
| | Selects Channel 1, and then measures the rise preshoot. | | |



| :MEASure:PPU | :PPULSE → Query | |
|---------------------|---|--|
| Description | Returns the number of positive pulses. | |
| Syntax | :MEASure:PF | PULSE{?} |
| Related Commands | :MEASure:SOURce <x></x> | |
| Return parameter | <nr3></nr3> | Returns the number of positive pulses. |
| | Chan Off | Indicates the source channel is not activated. |
| Note | Before using this command, select the measurement channel. See the example below. | |
| Example | :MEASure:SOURce1 CH1 | |
| | :MEASure:PPULSE? | |
| | 6.000E+00 | |
| | Selects Channel 1, and then measures the number of positive pulses. | |

:MEASure:NPULSE



| Description | Returns the number of negative pulses. | |
|---------------------|---|--|
| Syntax | :MEASure:NPULSE{?} | |
| Related Commands | :MEASure:SOURce <x></x> | |
| Return parameter | <nr3></nr3> | Returns the number of negative pulses. |
| | Chan Off | Indicates the source channel is not activated. |
| Note | Before using this command, select the measurement channel. See the example below. | |



Example :MEASure:SOURce1 CH1

:MEASure:NPULSE?

4.000E+00

Selects Channel 1, and then measures the number

of negative pulses.

:MEASure:PEDGE



| Description | Returns the number of positive edges. | | | |
|---------------------|---|--|--|--|
| Syntax | :MEASure:PE | :MEASure:PEDGE{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | | |
| Return parameter | <nr3> Returns the number of positive edges.</nr3> | | | |
| | Chan Off | Indicates the source channel is not activated. | | |
| Note | Before using this command, select the measurement channel. See the example below. | | | |
| Example | :MEASure:SOURce1 CH1 | | | |
| | :MEASure:PEDGE? | | | |
| | 1.100E+01 | | | |
| | Selects Channel 1, and then measures the number of positive edges. | | | |

:MEASure:NEDGE



| Description | Returns the number of negative edges. | | |
|---------------------|---------------------------------------|--|--|
| Syntax | :MEASure:NEDGE{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the number of negative edges. | |
| | Chan Off | Indicates the source channel is not activated. | |



| Note | Before using this command, select the measurement channel. See the example below. |
|---------|--|
| Example | :MEASure:SOURce1 CH1 :MEASure:NEDGE? 1.100E+01 Selects Channel 1, and then measures the number |
| | of negative edges. |

:MEASure:AMPlitude



| Description | Returns the amplitude difference between the Vhigh-Vlow. | | |
|---------------------|---|--|--|
| Syntax | :MEASure:AMPlitude{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the amplitude. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:AMPlitude? | | |
| | 3.76E-3 | | |
| | Selects Channel 1, and then measures the amplitude. | | |

:MEASure:MEAN



| Description | Returns the mean voltage/current of one or more full periods. |
|---------------------|---|
| Syntax | :MEASure:MEAN{?} |
| Related Commands | :MEASure:SOURce <x></x> |



| | | COMMINATED | |
|---------------------|---|--|--|
| Return parameter | <nr3></nr3> | Returns the mean. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:M | EAN? | |
| | 1.82E-3 | | |
| | Selects Channel 1, and then measures the mean value. | | |
| :MEASure:CME | an | → Query | |
| Description | Returns the mean voltage/current of one full period. | | |
| Syntax | :MEASure:CMEan{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the cyclic mean. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:CMEan? | | |
| | 9.480E-01 | | |
| | Selects Channel 1, and then measures the mean value of the first period. | | |
| :MEASure:HIG | Н | — Query | |
| Description | Returns the global high voltage/current. | | |
| | | | |



| Syntax | :MEASure:HIGH{?} | | |
|---------------------|---|--|--|
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the high value. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:HIGH? | | |
| | 3.68E-3 | | |
| | Selects Channel 1, and then measures the high voltage/current. | | |

:MEASure:LOW



| Description | Returns the global low voltage/current. | | |
|---------------------|---|--|--|
| Syntax | :MEASure:LOW{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the global low value. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 :MEASure:LOW? | | |
| | | | |
| | 1.00E-0 | | |
| | Selects Channel 1, and then measures the lov current/voltage. | | |



| :MEASure:MA | EASure:MAX → Query | | |
|---------------------|---|--|--|
| Description | Returns the maximum amplitude. | | |
| Syntax | :MEASure:MAX{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the maximum amplitude. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:MAX? | | |
| | 1.90E-3 | | |
| | Selects Channel 1, and then measures the maximum amplitude. | | |

:MEASure:MIN



| Description | Returns the minimum amplitude. | | |
|---------------------|---|--|--|
| Syntax | :MEASure:MIN{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the minimum amplitude. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |



Example :MEASure:SOURce1 CH1

:MEASure:MIN?

-8.00E-3

Selects Channel 1, and then measures the

minimum amplitude.

:MEASure:PK2PK



Description Returns the peak-to-peak amplitude (difference between maximum and minimum amplitude).

Syntax :MEASure:PK2Pk{?}

Related :MEASure:SOURce<X>

Commands

Return parameter <NR3> Returns the voltage or current peak to

peak measurement.

Chan Off Indicates the source channel is not

activated.

Note Before using this command, select the

measurement channel. See the example below.

Example :MEASure:SOURce1 CH1

:MEASure:PK2Pk?

2.04E-1

Selects Channel 1, and then measures the peak-to-

peak amplitude.

:MEASure:RMS



| Description | Returns the root-mean-square voltage/current of one or more full periods. | |
|---------------------|---|------------------------|
| Syntax | :MEASure:RMS{?} | |
| Related Commands | :MEASure:SOURce <x></x> | |
| Return parameter | <nr3></nr3> | Returns the RMS value. |



| | | COMMAND DETAILS | |
|---------------------|---|--|--|
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:RMS? | | |
| | 1.31E-3 | | |
| | Selects Char voltage/cur | nnel 1, and then measures the RMS crent. | |
| :MEASure:CRM | IS | → Query | |
| Description | Returns the root-mean-square voltage/current of one full periods. | | |
| Syntax | :MEASure:CRMS{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the CRMS value. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:S0 | OURce1 CH1 | |
| • | :MEASure:CRMS? | | |
| | 1.31E-3 | | |
| | Selects Channel 1, and then measures the CRMS voltage/current. | | |

:MEASure:AREa



| Description | Returns the voltage/current area over one or more full periods. |
|-------------|---|
| Syntax | :MEASure:AREa{?} |



| Related Commands | :MEASure:SOURce <x></x> | | |
|---------------------|---|--|--|
| Return parameter | <nr3> Returns the area value.</nr3> | | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:AREa? | | |
| | 1.958E-03 | | |
| | Selects Channel 1, and then measures the area. | | |

| :MEASure:CARea → Query | | → Query | |
|------------------------|---|--|--|
| Description | Returns the voltage/current area over one full period. | | |
| Syntax | :MEASure:CARea{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the area value. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Before using this command, select the measurement channel. See the example below. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:CARea? | | |
| | 1.958E-03 | | |
| | Selects Channel 1, and then measures the area. | | |



| :MEASure:FRRDelay | | → Query | |
|---------------------|--|--|--|
| Description | Returns the delay between the first rising edge of source1 and the first rising edge of source2. | | |
| Syntax | :MEASure:FR | RRDelay{?} | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the delay. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Select the two source channels before entering this command. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:SOURce2 CH2 | | |
| | :MEASure:FRRDelay? | | |
| | -4.68E-6 | | |
| | Select channel 1 and 2 as source1/2, and then measure FRR. | | |

| :MEASure:FRFDelay → Query | | — Query | |
|---------------------------|---|--|--|
| Description | Returns the delay between the first rising edge of source1 and the first falling edge of source2. | | |
| Syntax | :MEASure:FRFDelay{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the delay. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Select the two source channels before entering this command. | | |



Example :MEASure:SOURce1 CH1

:MEASure:SOURce2 CH2

:MEASure:FRFDelay?

3.43E-6

Select channel 1 and 2 as source1/2, and then

measures FRF.

:MEASure:FFRDelay



| Description | Returns the delay between the first falling edge of |
|-------------|---|
| | source1 and the first rising edge of source2. |

Syntax :MEASure:FRRDelay{?}

Related :MEASure:SOURce<X>

Commands

Return parameter <NR3> Returns the delay.

Chan Off Indicates the source channel is not activated.

activated

Note Select the two source channels before entering this

command.

Example :MEASure:SOURce1 CH1

:MEASure:SOURce2 CH2

:MEASure:FRRDelay?

-8.56E-6

Select channel 1 and 2 as delay source1/2, and

then measure FFR.

:MEASure:FFFDelay



| Description | Returns the delay between the first falling edge of source1 and the first falling edge of source2. | |
|---------------------|--|--|
| Syntax | :MEASure:FFFDelay{?} | |
| Related Commands | :MEASure:SOURce <x></x> | |



| Return parameter | <nr3></nr3> | Returns the delay. | |
|------------------|----------------------------|--|--|
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Select the two | vo source channels before entering this | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:SOURce2 CH2 | | |
| | :MEASure:FFFDelay? | | |
| | -8.89E-6 | | |
| | Select chanr then measu | nel 1 and 2 as delay source1/2, and re FFF. | |

:MEASure:LRRDelay



| Description | Returns the delay between the first rising edge of source1 and the last rising edge of source2. | | |
|---------------------|--|--|--|
| Syntax | :MEASure:LRRDelay{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3> Returns the delay.</nr3> | | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Select the two source channels before entering this command. | | |
| Example | :MEASure:SOURce1 CH1 :MEASure:SOURce2 CH2 :MEASure:LRRDelay? -8.89E-6 Select channel 1 and 2 as delay source1/2, and then measure LRR. | | |
| | | | |
| | | | |
| | | | |
| | | | |



| :MEASure:LRFDelay | | → Query | |
|---------------------|---|--|--|
| Description | Returns the delay between the first rising edge of source1 and the last rising edge of source2. | | |
| Syntax | :MEASure:LRFDelay{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the delay. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Select the two source channels before entering this command. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:SOURce2 CH2 | | |
| | :MEASure:LRFDelay? | | |
| | -4.99E-6 | | |
| | Select channel 1 and 2 as delay source1/2, and then measure LRF. | | |

| :MEASure:LFRDelay → Query | | | |
|---------------------------|--|--|--|
| Description | Returns the delay between the first falling edge of source1 and the last rising edge of source2. | | |
| Syntax | :MEASure:LFRDelay{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | <nr3></nr3> | Returns the delay. | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Select the two source channels before entering this command. | | |



Example :MEASure:SOURce1 CH1

:MEASure:SOURce2 CH2 :MEASure:LFRDelay?

-9.99E-6

-9.99E-6

then measure LFF.

Select channel 1 and 2 as delay source1/2, and

then measure LFR.

:MEASure:LFFDelay



| Description | Returns the delay between the first falling edge of source1 and the last falling edge of source2. | | |
|---------------------|---|--|--|
| Syntax | :MEASure:LFFDelay{?} | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Return parameter | r <nr3> Returns the delay.</nr3> | | |
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Select the two source channels before entering this command. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:SOURce2 CH2 | | |
| | :MEASure:LFFDelay? | | |

:MEASure:PHAse



| Description | Returns the phase between source 1 and source 2. |
|---------------------|--|
| Syntax | :MEASure:PHAse{?} |
| Related Commands | :MEASure:SOURce <x></x> |
| Commands | |

Select channel 1 and 2 as delay source1/2, and



| Return parameter | <nr3></nr3> | Returns the phase difference. | |
|------------------|---|--|--|
| | Chan Off | Indicates the source channel is not activated. | |
| Note | Select the two source channels before entering this command. | | |
| Example | :MEASure:SOURce1 CH1 | | |
| | :MEASure:SOURce2 CH2 | | |
| | :MEASure:PHAse? | | |
| | 4.50E+01 | | |
| | Select channel 1 and 2 as phase source1/2, and then measure the phase in degrees. | | |

Measurement Commands

| | :MEASUreme | nt:MEAS <x>:SOURCE<x>105</x></x> |
|---------------------|-----------------------------|--|
| | :MEASUreme | nt:MEAS <x>:TYPe106</x> |
| | :MEASUreme | nt:MEAS <x>:STATE106</x> |
| | :MEASUreme | nt:MEAS <x>:VALue107</x> |
| | :MEASUreme | nt:MEAS <x>:MAXimum108</x> |
| | :MEASUreme | nt:MEAS <x>:MEAN109</x> |
| | :MEASUreme | nt:MEAS <x>:MINImum109</x> |
| | :MEASUreme | nt:MEAS <x>:STDdev110</x> |
| | :MEASUreme | nt:STATIstics:MODe111 |
| | :MEASUreme | nt:STATIstics:WEIghting111 |
| | :MEASUreme | nt:STATIstics111 |
| | | |
| | | Set → |
| :MEASUrement | t:MEAS <x>:</x> | SOURCE <x> → Query</x> |
| Description | | es the measurement source for a smatic measurement. This is a statistics nand. |
| Syntax | :MEASUreme CH3 CH4 N | nt:MEAS <x>:SOURCE<x> { CH1 CH2 MATH ? }</x></x> |
| Related commands | :MEASUreme | nt:MEAS <x>:TYPe</x> |
| Parameter | MEAS <x></x> | The automatic measurement number from 1 to 8. |
| | SOURCE <x></x> | SOURCE1: the source for all single channel measurements. |
| | SOURCE <x></x> | SOURCE2: the source for all delay or phase measurements. |
| | CH1 to CH4 | Channel 1, 2, 3, 4 |
| | MATH | Math source |
| Return parameter | CH1 to CH4 | Channel 1 2 3 4 |



| | MATH | Math source | |
|--|---|-----------------------|-----------------|
| Example | :MEASUrement:MEAS1:SOURCE1 | | |
| | >CH1 | | |
| | Returns the | (first) source for me | easurement 1. |
| | | | Set → |
| :MEASUrement:MEAS $<$ X $>$:TYPe \longrightarrow Qu | | | → Query |
| Description | Sets or queries the measurement type for a selected automatic measurement. This is a statistics related command. | | |
| Syntax | :MEASUrement:MEAS <x>:TYPe {PK2pk MAXimum MINImum AMPlitude HIGH LOW MEAN CMEan RMS CRMs AREa CARea ROVShoot FOVShoot RPReshoot FPReshoot FREQuency PERIod RISe FALL PWIdth NWIdth PDUTy PPULSE NPULSE PEDGE NEDGE FRRDelay FRFDelay FFFDelay LRRDelay LRFDelay LFRDelay LFFDelay PHAse ?}</x> | | |
| Related commands | :MEASUrement:MEAS <x>:SOURCE<x></x></x> | | |
| Parameter | MEAS <x></x> | The automatic meas | surement number |
| Return parameter | Returns the measurement type | | |
| Example | :MEASUrement:MEAS1:TYPe RMS | | S |
| | Sets measurement 1 to RMS measurement. | | |
| :MEASUremen | t:MEAS <x></x> | -:STATE | Set → Query |
| Description | Sets or queries the state of a selected measurement. This is a statistics related command. | | |
| Syntax | :MEASUrement:MEAS <x>:STATE { ON OFF 1 0 ? }</x> | | |



Note

:MEASUrement:MEAS<X>:VALue

| Related commands | :MEASUrement:MEAS <x>:SOUrce<x> :MEASUrement:MEAS<x>:TYPe</x></x></x> | | |
|---------------------|---|---|--|
| Parameter | MEAS <x></x> | The automatic measurement number from 1 to 8. | |
| | ON/1 | Turn the measurement on. | |
| | OFF/0 | Turn the measurement off. | |
| Return parameter | 0 | Measurement is off. | |
| | 1 | Measurement is on. | |
| Example | :MEASUrement:MEAS1:STATE 1 | | |
| | Turns measurement 1 on. | | |

| Description | Returns the measurement results for the selected measurement. This is a statistics related command. | | |
|---------------------|---|--|--|
| Syntax | :MEASUrement:MEAS <x>:VALue?</x> | | |
| Related Commands | :MEASure:SOURce <x></x> | | |
| Parameter | MEAS <x></x> | The automatic measurement number from 1 to 8. | |
| Return parameter | <nr3></nr3> | Returns the measurement for the selected measurement number. | |

can be returned.

The measurement source(s), measurement

number, measurement type and measurement state must first be set before a measurement result



Example :MEASUrement:MEAS1:SOUrce1 CH1

:MEASUrement:MEAS1:TYPe PK2PK :MEASUrement:MEAS1:STATE ON :MEASUrement:MEAS1:VALue?

5.000E+0

Selects channel 1 as the source for measurement 1, sets measurement 1 to peak to peak measurement and then turns on the measurement. The result returns the peak to peak measurement.

:MEASUrement:MEAS<X>:MAXimum



| Description | Returns the maximum measurement results for the selected measurement from the last time the statistics were reset. This is a statistics related command. | | |
|---------------------|---|--|--|
| Syntax | :MEASUrement:MEAS <x>:MAXimum?</x> | | |
| Related Commands | :MEASUrement:STATIstics:MODe | | |
| Parameter | MEAS <x></x> | The automatic measurement number from 1 to 8. | |
| Return parameter | <nr3></nr3> | Returns the measurement for the selected measurement number. | |
| Example | :MEASUrement:MEAS3:SOUrce1 CH1 :MEASUrement:MEAS3:TYPe PK2PK :MEASUrement:MEAS3:STATE ON :MEASUrement:STATIstics:MODe ON :MEASUrement:MEAS3:MAXimum? 2.800E-02 Returns the maximum measurement result for | | |
| | measurement number 3. | | |



| Description | Returns the mean measurement results for the selected measurement from the last time the statistics were reset. This is a statistics related command. | | |
|---------------------|---|---|--------|
| Syntax | :MEASUrem | ent:MEAS <x>:MEAN?</x> | |
| Related Commands | :MEASUrem | ent:STATIstics:MODe | |
| Parameter | MEAS <x></x> | The automatic measurement r from 1 to 8. | number |
| Return parameter | <nr3></nr3> | Returns the measurement for selected measurement number | - |
| Example | :MEASUrem | ent:MEAS3:SOUrce1 CH1 | |
| | :MEASUrem | ent:MEAS3:TYPe PK2PK | |
| | :MEASUrement:MEAS3:STATE ON | | |
| | :MEASUrement:STATIstics:MODe ON | | |
| | :MEASUrement:MEAS3:MEAN? | | |
| | 2.090E-02 | | |
| | Returns the mean measurement result for measurement number 3. | | |

| Description | Returns the minimum measurement results for the selected measurement from the last time the statistics were reset. This is a statistics related command. | |
|---------------------|--|---|
| Syntax | :MEASUrement:MEAS <x>:MINImum?</x> | |
| Related Commands | :MEASUrement:STATIstics:MODe | |
| Parameter | MEAS <x></x> | The automatic measurement number from 1 to 8. |



| Return parameter | <nr3></nr3> | Returns the measurement for the selected measurement number. |
|------------------|-------------|--|
| Example | :MEASUrem | ent:MEAS3:SOUrce1 CH1 |
| | :MEASUrem | ent:MEAS3:TYPe PK2PK |
| | :MEASUrem | ent:MEAS3:STATE ON |
| | :MEASUrem | ent:STATIstics:MODe ON |
| | :MEASUrem | ent:MEAS3:MINImum? |
| | 1.600E-02 | |
| | | minimum measurement result for nt number 3. |

:MEASUrement:MEAS<X>:STDdev

| | 1912 dev | | |
|---------------------|---|--|--|
| Description | Returns the standard deviation for the selected measurement from the last time the statistics were reset. This is a statistics related command. | | |
| Syntax | :MEASUrement:MEAS <x>:STDdev?</x> | | |
| Related Commands | :MEASUrement:STATIstics:MODe | | |
| Parameter | MEAS <x></x> | The automatic measurement number from 1 to 8. | |
| Return parameter | <nr3></nr3> | Returns the measurement for the selected measurement number. | |
| Example | :MEASUrem | ent:MEAS3:SOUrce1 CH1 | |
| | :MEASUrement:MEAS3:TYPe PK2PK | | |
| | :MEASUrement:MEAS3:STATE ON | | |
| | :MEASUrement:STATIstics:MODe ON | | |
| | :MEASUrement:MEAS3:STDdev? | | |
| | 1.530E-03 | | |
| | Returns the standard deviation for measurement number 3. | | |



| :MEASUremen | t:STATIstics | :MODe Set → Query |
|--------------------------------|--|---|
| Description | Puts the statics measurement results on the display or queries whether the statistics are displayed. | |
| Syntax | :MEASUreme | nt:STATIstics:MODe {OFF ON ?} |
| Related commands | :MEASUrement:STATIstics | |
| Parameter/ | ON | Display the statistics on the screen. |
| Return parameter | OFF | Remove the statistics from the screen |
| Example | :MEASUreme | nt:STATIstics:MODe ON |
| | Displays stat | tistics on the screen. |
| :MEASUremen Description | Sets and queries the number of samples (weighting) used for the statistics calculations. | |
| Syntax | :MEASUrement:STATIstics:WEIghting { <nr1> ? }</nr1> | |
| Parameter/ Return parameter | <nr1></nr1> | Number of samples (2~1000) |
| Example | :MEASUrement:STATIstics:WEIghting 5 | |
| | Sets the number of samples to 5. | |
| | | |
| :MEASUremen | t:STATIstics | Set → |
| Description | | atics calculations. This command will currently accumulated measurements. |
| Syntax | :MEASUreme | nt:STATIstics {RESET} |



Reference Commands

| NCICICITICE C | Jonnand | u 3 | |
|--------------------|---|--|--|
| | :REF <x>:TIM :REF<x>:TIM :REF<x>:OFF</x></x></x> | Play 112 ebase:POSition 112 ebase:SCALe 113 Set 113 Le 114 | |
| :REF <x>:DISPl</x> | ay | Set → Query | |
| Description | Sets or queries whether a reference waveform will be shown on the display. A reference waveform must first be saved before this command can be used. | | |
| Syntax | :REF <x>:DISF</x> | Play { OFF ON ? } | |
| Parameter | <x></x> | Reference waveform 1, 2, 3, 4. | |
| | OFF | Turns the selected reference waveform off | |
| | ON | Turns the selected reference waveform on | |
| Return parameter | Returns the status of the selected reference waveform. (OFF, ON). | | |
| Example | :REF1:DISPla | y ON | |
| | Turns on reference1 (REF 1) on the display. | | |
| :REF <x>:TIMel</x> | pase:POSitio | Set → On —Query | |
| Description | Sets or returns the selected reference waveform time base position. | | |
| Syntax | :REF <x>:TIM</x> | ebase:POSition { <nrf> ?}</nrf> | |
| Related commands | :REF <x>:DISPlay</x> | | |



| Parameter | <x> <nrf></nrf></x> | Reference waveform 1, 2, 3, 4. Horizontal co-ordinates |
|--------------------|---|--|
| | | |
| Return parameter | <nr3></nr3> | Returns the reference waveform position |
| Example | :REF1:TIN | Nebase:POSition -5.000E-5 |
| | Selects reposition | eference 1, and then sets the horizontal to -50us. |
| :REF <x>:TIMeb</x> | oase:SCA | Set → Query |
| Description | Sets or returns the selected reference waveform time base scale. | |
| Syntax | :REF <x>:TIMebase:SCALe { <nrf> ?}</nrf></x> | |
| Related commands | :REF <x>:DISPlay</x> | |
| Parameter | <x></x> | Reference waveform 1, 2, 3, 4. |
| | <nrf></nrf> | Horizontal scale |
| Return parameter | <nr3></nr3> | Returns the reference waveform scale. |
| Example | :REF1:TIMebase:SCALe 5.00E-4 | |
| | Selects reference 1, and then sets the horizontal scale to 500us/div. | |
| | | Set → |
| :REF <x>:OFFS</x> | et | ——Query |
| Description | Sets or returns the selected reference waveform vertical position (offset). | |
| Syntax | :REF <x>:OFFSet { <nrf> ?}</nrf></x> | |
| Related commands | :REF <x>:DISPlay</x> | |
| Parameter | <x></x> | Reference waveform 1, 2, 3, 4. |
| | <nrf></nrf> | Vertical offset |



| Return parameter | <nr3></nr3> | Returns the reference waveform vertical position. |
|--------------------|---|--|
| Example | :REF1:OF | FSet -5.000E-2 |
| | Selects reference 1, and then sets the vertical position to -50mV/mA. | |
| :REF <x>:SCALe</x> | <u>)</u> | Set → Query |
| Description | Sets or returns the selected reference waveform vertical scale. | |
| Syntax | :REF <x>:SCALe { <nrf> ? }</nrf></x> | |
| Related commands | :REF <x>:DISPlay</x> | |
| Parameter | <x></x> | Reference waveform 1, 2, 3, 4. |
| | <nrf></nrf> | Vertical scale |
| Return parameter | <nr3></nr3> | Returns the reference waveform vertical scale. |
| Example | :REF1:SC | ALe 5.000E-2 |
| | | eference 1, and then sets the vertical scale mA/div. |



Run Command

| :RUN | Set → |
|-------------|--|
| Description | The run command allows the oscilloscope to continuously make acquisitions (equivalent to pressing the Run key on the front panel). |
| Syntax | :RUN |

Stop Command

| :STOP | (Set)→ |
|-------------|---|
| Description | The stop command stops the oscilloscope making further acquisitions (equivalent to pressing the Stop key on the front panel). |
| Syntax | :STOP |

Single Command

| :SINGle | Set → |
|-------------|--|
| Description | The single command allows the oscilloscope to capture a single acquisition when trigger conditions have been fulfilled (equivalent to pressing the Single key on the front panel). |
| Syntax | :SINGle |



Force Command

| :FORCe | Set → |
|-------------|---|
| Description | The Force command forces an acquisition (equivalent to pressing the Force-Trig key on the front panel). |
| Syntax | :FORCe |



Timebase Commands

| | :TIMebase:PO :TIMebase:SO :TIMebase:M :TIMebase:W | (Pand | |
|-------------------------------|---|--|--|
| :TIMebase:EXP | and | Set → Query | |
| Description | Sets or queries the horizontal expansion mode. | | |
| Syntax | | (Pand {CENTer TRIGger ?} | |
| Parameter/Return parameter | | Expand from the center of the display. | |
| | TRIGger | Expand from the trigger point. | |
| Example | :TIMebase:EX | FIMebase:EXPand TRIGger | |
| | Sets the expansion point to the trigger point. | | |
| :TIMebase:POS | Sition | Set → Query | |
| Description | Sets or queries the horizontal position. | | |
| Syntax | :TIMebase:POSition { <nrf> ?}</nrf> | | |
| Parameter | <nrf></nrf> | Horizontal position | |
| Return parameter | <nr3></nr3> | Returns the horizontal position | |
| Example | :TIMebase:POSition 5.00E-4 Sets the horizontal position as 500us. | | |
| :TIMebase:SCA | Le | Query | |
| Description | Sets or queries the horizontal scale. | | |



| Syntax | :TIMebase:SCALe { <nrf> ?}</nrf> | |
|---------------------|---|---|
| Parameter | <nrf></nrf> | Horizontal scale |
| Return parameter | <nr3></nr3> | Returns the horizontal scale. |
| Example | :TIMebase:SCALe 5.00E-2 | |
| | Sets the hori | zontal scale to 50ms/div. |
| | | Set → |
| :TIMebase:MO | De | → Query |
| Description | Sets or queries the time base mode. The time base mode determines the display view window on the scope. | |
| Syntax | :TIMebase:MODe {MAIN WINDow XY ?} | |
| Parameter | MAIN | Sets the time base mode to the main screen. |
| | WINDow | Sets the time base mode to the zoom window. |
| | XY | Sets the time base mode to the XY display. |
| Return parameter | Returns the time base mode (MAIN, WINDOW, XY) | |
| Example | :TIMebase:MODe MAIN | |
| | Sets the time base mode to the main mode. | |
| | | Set → |
| :TIMebase:WIN | IDow:POSit | ion → Query |
| Description | Sets or queries the zoom horizontal position. | |
| Syntax | :TIMebase:WINDow:POSition { <nrf> ?}</nrf> | |
| Related commands | :TIMebase:MODe | |
| Parameter | <nrf></nrf> | Horizontal position for zoom window |
| Return parameter | <nr3></nr3> | Returns the zoom horizontal position. |



Example :TIMebase:WINDow:POSition 2.0E-3
Sets the zoom horizontal position as 20ms.

:TIMebase:WINDow:SCALe → Query

| Description | Sets or queries the zoom horizontal scale. | | |
|---------------------|---|--|--|
| Note | If the oscilloscope is under "ZOOM" mode, the main timebase function will be disabled and cannot be modified. | | |
| Syntax | :TIMebase:WINDow:SCALe { <nrf> ?}</nrf> | | |
| Related commands | :TIMebase:MODe | | |
| Parameter | <nrf></nrf> | Zoom horizontal scale. The range will depend on the time base. | |
| Return parameter | <nr3></nr3> | Returns the zoom horizontal scale. | |
| Example | :TIMebase:WINDow:SCALe 2.0E-3 | | |
| | Sets the zoom horizontal scale to 2ms. | | |



Trigger Commands

| :TRIGger:FREQuency | 122 |
|------------------------------|-----|
| :TRIGger:TYPe | 122 |
| :TRIGger:SOURce | 122 |
| :TRIGger:COUPle | 123 |
| :TRIGger:NREJ | 123 |
| :TRIGger:MODe | 124 |
| :TRIGger:HOLDoff | 124 |
| :TRIGger:LEVel | 124 |
| :TRIGger:HLEVel | 125 |
| :TRIGger:LLEVel | |
| :TRIGger:EDGe:SLOP | 126 |
| :TRIGger:DELay:SLOP | 126 |
| :TRIGger:DELay:TYPe | 127 |
| :TRIGger:DELay:TIMe | 127 |
| :TRIGger:DELay:EVENt | 127 |
| :TRIGger:DELay:LEVel | 128 |
| :TRIGger:PULSEWidth:POLarity | 128 |
| :TRIGger:RUNT:POLarity | 128 |
| :TRIGger:RUNT:WHEn | 129 |
| :TRIGger:RUNT:TIMe | 129 |
| :TRIGger:RISEFall:SLOP | 130 |
| :TRIGger:RISEFall:WHEn | 130 |
| :TRIGger:RISEFall:TIMe | 131 |
| :TRIGger:VIDeo:TYPe | 131 |
| :TRIGger:VIDeo:FIELd | 132 |
| :TRIGger:VIDeo:LINe | 132 |
| :TRIGger:VIDeo:POLarity | 133 |
| :TRIGger:PULSe:WHEn | 133 |
| :TRIGger:PULSe:TIMe | 134 |
| :TRIGger:TIMEOut:WHEn | 134 |
| :TRIGger:TIMEOut:TIMER | 134 |
| ·TRIGger: Al Ternate | 135 |



| :TRIGger:STATe | 135 |
|--|-----|
| :TRIGger:EXTERnal:PROBe:TYPe | 136 |
| :TRIGger:EXTERnal:PROBe:RATio | 136 |
| :TRIGger:BUS:TYPe | 137 |
| :TRIGger:BUS:THReshold:CH <x></x> | 137 |
| :TRIGger:BUS:B1:I2C:CONDition | 137 |
| :TRIGger:BUS:B1:I2C:ADDRess:MODe | 138 |
| :TRIGger:BUS:B1:I2C:ADDRess:TYPe | 139 |
| :TRIGger:BUS:B1:I2C:ADDRess:VALue | 139 |
| :TRIGger:BUS:B1:I2C:ADDRess:DIRection | 140 |
| :TRIGger:BUS:B1:I2C:DATa:SIZe | 140 |
| :TRIGger:BUS:B1:I2C:DATa:VALue | 141 |
| :TRIGger:BUS:B1:UART:CONDition | 142 |
| :TRIGger:BUS:B1:UART:RX:DATa:SIZe | 142 |
| :TRIGger:BUS:B1:UART:RX:DATa:VALue | 143 |
| :TRIGger:BUS:B1:UART:TX:DATa:SIZe | 143 |
| :TRIGger:BUS:B1:UART:TX:DATa:VALue | 144 |
| :TRIGger:BUS:B1:SPI:CONDition | 145 |
| :TRIGger:BUS:B1:SPI:DATa:SIZe | 145 |
| :TRIGger:BUS:B1:SPI:DATa:MISO:VALue | 146 |
| :TRIGger:BUS:B1:SPI:DATa:MOSI:VALue | 147 |
| :TRIGger:BUS:B1:CAN:CONDition | 147 |
| :TRIGger:BUS:B1:CAN:FRAMEtype | 148 |
| : TRIGger: BUS: B1: CAN: IDentifier: MODe | 148 |
| :TRIGger:BUS:B1:CAN:IDentifier:VALue | 149 |
| : TRIGger: BUS: B1: CAN: IDentifier: DIRection | |
| :TRIGger:BUS:B1:CAN:DATa:QUALifier | 150 |
| :TRIGger:BUS:B1:CAN:DATa:SIZe | |
| :TRIGger:BUS:B1:CAN:DATa:VALue | 151 |
| :TRIGger:BUS:B1:LIN:CONDition | |
| :TRIGger:BUS:B1:LIN:DATa:QUALifier | |
| :TRIGger:BUS:B1:LIN:DATa:SIZe | |
| :TRIGger:BUS:B1:LIN:DATa:VALue | |
| :TRIGger:BUS:B1:LIN:ERRTYPE | 155 |
| ·TRIGger:BUS:B1:LIN:IDentifier:VALue | 156 |



| :TRIGger:FREQuency → Query | | | |
|----------------------------|--|--------------------------------|--|
| Description | Queries the trigger frequency. | | |
| Syntax | :TRIGger:FREQuency{?} | | |
| Return parameter | <nr3></nr3> | Returns the trigger frequency. | |
| Example | :TRIGger:FRE | Quency? | |
| | 1.032E+3 | | |
| | Returns the t | trigger frequency. | |
| | | Set → | |
| :TRIGger:TYPe | | → Query | |
| Description | Sets or queries the trigger type. | | |
| Syntax | :TRIGger:TYPe {EDGe DELay PULSEWidth VIDeo RUNT RISEFall BUS TIMEOut ? } | | |
| Parameter | EDGE | Edge trigger | |
| | DELay | Delay trigger | |
| | PULSEWidth | Pulse width trigger | |
| | VIDeo | Video trigger | |
| | RUNT | Runt trigger | |
| | RISEFall | Rise and fall trigger | |
| | BUS | Bus trigger | |
| | TIMEOut | Timeout trigger | |
| Return parameter | Returns the trigger type. | | |
| Example | :TRIGger:TYP | | |
| | Sets the trigger type to edge. | | |
| | (Set)→ | | |
| :TRIGger:SOUF | :TRIGger:SOURce → Query | | |
| Description | Sets or queries the trigger source. | | |



| Syntax | :TRIGger:SOURce { CH1 CH2 CH3 CH4 EXT LINe ? } | | |
|------------------|---|------------------------------|--|
| Parameter | CH1 to CH4 Channel 1 to channel 4 | | |
| | EXT | External source | |
| | LINe | AC Line | |
| Return parameter | Returns the | trigger source. | |
| Example | :TRIGger:SOl | JRce CH1 | |
| | Sets the trigg | ger source to channel 1. | |
| | | <u>Set</u> → | |
| :TRIGger:COUI | ::COUPle ——Query | | |
| Description | Sets or queries the trigger coupling. | | |
| Note | Applicable for edge and delay triggers only. | | |
| Syntax | :TRIGger:COI | JPle {AC DC HF LF ?} | |
| Parameter | AC | AC mode | |
| | DC | DC mode | |
| | HF | High frequency rejection | |
| | LF | Low frequency rejection | |
| Return parameter | Returns the trigger coupling. | | |
| Example | :TRIGger:COUPle AC | | |
| | Sets the trigger coupling to AC. | | |
| | Set → | | |
| :TRIGger:NREJ | → Query | | |
| Description | Sets or queries noise rejection status. | | |
| Syntax | :TRIGger:NREJ {OFF ON ?} | | |
| Parameter | OFF | Turns noise rejection off | |
| | ON | Turns noise rejection on | |
| | | | |

Return parameter Returns the noise rejection status (ON, OFF).



Example :TRIGger:NREJ ON

Turns noise rejection on.

:TRIGger:MODe

| _ | Set) | • |
|---|---------------------|---------------|
| | \(\int_{\text{0}}\) | $\overline{}$ |
| _ | → Query | ע |

| Description | Sets or queries the trigger mode. | |
|------------------|-----------------------------------|---------------------------------|
| Syntax | :TRIGger:MODe {AUTo NORMal ?} | |
| Parameter | AUTo | Auto trigger (Untriggered roll) |
| | NORMal | Normal trigger |
| Return parameter | r Returns the trigger mode. | |

Example :TRIGger:MODe NORMal

Sets the trigger mode to normal.

:TRIGger:HOLDoff



| Description | Sets or queries the holdoff time. | |
|------------------|---|--------------|
| Syntax | :TRIGger:HOLDoff { <nrf> ?}</nrf> | |
| Parameter | <nrf></nrf> | Holdoff time |
| Return parameter | <nr3> Returns the trigger holdoff time.</nr3> | |
| Example | :TRIGger:HOLDoff 1.00E-8 | |

Sets the trigger holdoff time to 10ns.

:TRIGger:LEVel



| Description | Sets or queries the level. | |
|------------------|--|--|
| Note | Not applicable to Pulse Runt and Rise & Fall triggers. | |
| Syntax | :TRIGger:LEVel {TTL ECL SETTO50 <nrf> ?}</nrf> | |
| Related commands | :TRIGger:TYPe | |
| Parameter | <nrf> Trigger level value.</nrf> | |



| | TTL | Sets the trigger level to TTL. |
|-------------------------|--|--|
| | ECL | Sets the trigger level to ECL. |
| | SETTO50 | Sets the trigger level to the User level (50% by default). |
| Return parameter | <nr3></nr3> | Returns the trigger level. |
| Example1 | :TRIGger:LEVel TTL | |
| | Sets the trigg | ger to TTL. |
| Example2 | :TRIGger:LEVel 3.30E-1 | |
| | Sets the trigger level to 330mV/mA. | |
| | | Set |
| :TRIGger:HLEVel → Query | | |
| Description | Sets or queries the high trigger level. | |
| Nata | A = -1!1.1 - (- = D' 1 E-11 /D-1 - D = (tol | |

| Description | Sets or queries the high trigger level. | |
|---------------------|---|---------------------------------|
| Note | Applicable for Rise and Fall/Pulse Runt triggers. | |
| Syntax | :TRIGger:HLEVel { <nrf> ?}</nrf> | |
| Related commands | :TRIGger:TYPe | |
| Parameter | <nrf></nrf> | High level value. |
| Return parameter | <nr3></nr3> | Returns the trigger high level. |
| Example | :TRIGger:HLEVel 3.30E-1 | |
| | Sets the trigger high level to 330mV/mA. | |

:TRIGger:LLEVel —Query

| Description | Sets or queries the low trigger level. | |
|---------------------|--|---------------------------------------|
| Note | Applicable fo | or Rise and Fall/Pulse Runt triggers. |
| Syntax | :TRIGger:LLEVel { <nrf> ?}</nrf> | |
| Related commands | :TRIGger:TYPe | |
| Parameter | <nrf></nrf> | Low level value. |



| Return parameter | <nr3></nr3> | Returns the trigger low level. |
|----------------------------------|--|-------------------------------------|
| Example | :TRIGger:LLEVel -3.30E-3 | |
| | Sets the trigg | ger low level to -330mV/mA. |
| | | Set → |
| :TRIGger:EDGe | :SLOP | → Query |
| Description | Sets or queri | es the trigger slope. |
| Syntax | :TRIGger:EDG | Ge:SLOP {RISe FALL EITher ? } |
| Related commands | :TRIGger:TYP | e e |
| Parameter | RISe | Rising slope |
| | FALL | Falling slope |
| | EITher | Either rising or falling slope |
| Return parameter | Returns the trigger slope. | |
| Example | :TRIGger:EDGe:SLOP FALL Sets the trigger slope to falling. | |
| | | |
| | | Set → |
| :TRIGger:DELa | y:SLOP | → Query |
| Description | Sets or queries the trigger slope for the delay trigger. | |
| Syntax | :TRIGger:DELay:SLOP {RISe FALL EITher ? } | |
| Related commands | :TRIGger:TYPe | |
| Parameter | RISe | Rising slope |
| | FALL | Falling slope |
| | EITher | Either rising or falling slope |
| Return parameter | Returns the trigger slope. | |
| Example :TRIGger:DELay:SLOP FALL | | ay:SLOP FALL |
| | Sets the trigger slope to falling. | |



| :TRIGger:DELa | y:TYPe | | Set → Query |
|---|---|---|-------------|
| Description | Sets or queri | Sets or queries the trigger delay type. | |
| Syntax | :TRIGger:DEL | _ay:TYPE {TIMe EVE | Nt ?} |
| Related commands | :TRIGger:TYP | 'e | |
| Parameter | TIMe | Sets the delay type | to time. |
| | EVENt | Sets the delay type | to event. |
| Return parameter | Returns the t | trigger delay type. | |
| Example | :TRIGger:DEL | ay:TYPe TIMe | |
| | Sets the dela | y type to time delay | ·. |
| :TRIGger:DELay:TIMe \longrightarrow Query | | | |
| Description | Sets or queri | es the delay time va | lue. |
| Syntax | :TRIGger:DELay:TIMe { <nrf> ?}</nrf> | | |
| Related commands | :TRIGger:DELay:TYPe | | |
| Parameter | <nrf></nrf> | Delay time (1.00E-8 | 3~1.00E+1) |
| Return parameter | <nr3></nr3> | Returns the delay t | ime. |
| Example | :TRIGger:DEL | ay:TIMe 1.00E-6 | |
| | Sets the dela | y time to 1us. | |
| $\begin{array}{ccc} & & & & & \\ & & & & \\ :TRIGger:DELay:EVENt & & & & \\ & & & & \\ \hline & & & & \\ & & & &$ | | | |
| Description | Sets or queries the number of events for the event delay trigger. | | |
| Syntax | :TRIGger:DEL | ay:EVENt { <nr1> ?</nr1> | r} |
| Related commands | :TRIGger:DELay:TYPe | | |



| Parameter | <nr1></nr1> | 1~65535 events |
|--------------------------------|---------------------------------------|--------------------------------------|
| Return parameter | <nr1></nr1> | Returns the number of events. |
| Example | :TRIGger:DELay:EVENt 2 | |
| ' | • | nber of events to 2. |
| | | (Set)→ |
| :TRIGger:DELa | y:LEVel | Query |
| Description | Sets or queri | es the trigger delay level. |
| Syntax | | _ay:LEVel { <nrf> ?}</nrf> |
| Parameter | <nrf></nrf> | Delay trigger level |
| Return parameter | <nr3></nr3> | Returns the delay trigger. |
| Example | :TRIGger:DEI | ay:LEVel 5.00E-3 |
| | Sets the dela | y trigger level to 5mV/mA. |
| | | Set |
| :TRIGger:PULS | EWidth:POI | _arity ——Query |
| Description | Sets or queri | es the pulse width trigger polarity. |
| Syntax | :TRIGger:PUI {POSitive N | _SEWidth:POLarity EGative ?} |
| Related commands | :TRIGger:TYP | Pe |
| Parameter | POSitive | Positive polarity |
| | NEGative | Negative polarity |
| Return parameter | Returns the pulse width polarity. | |
| Example | :TRIGger:PULSEWidth:POLarity POSitive | |
| | Sets the puls | e width polarity to positive. |
| | | <u>Set</u> → |
| :TRIGger:RUNT:POLarity → Query | | |
| Description | Sets or queri | es the Pulse Runt trigger polarity. |



| C . | TDIC DIII | ITROL III (POCIII - LNEC III - L |
|--------------------|---|---------------------------------------|
| Syntax | :TRIGger:RUNT:POLarity { POSitive NEGative EITher ? } | |
| Related commands | :TRIGger:TYP | e |
| Parameter | POSitive | Positive polarity |
| | NEGative | Negative polarity |
| | EITher | Positive or negative polarity |
| Return parameter | Returns the 1 | pulse runt trigger polarity. |
| Example | :TRIGger:RUI | NT:POLarity POSitive |
| | Sets the Puls | e Runt trigger polarity to positive. |
| | | <u>Set</u> → |
| :TRIGger:RUN | Γ:WHEn | → Query |
| Description | Sets or queri | es the Pulse Runt trigger conditions. |
| Syntax | :TRIGger:RUNT:WHEn $\{MOREthan LESSthan EQual UNEQual ?\}$ | |
| Related commands | :TRIGger:TYPe :TRIGger:RUNT:TIMe | |
| Parameter | MOREthan | > |
| | LESSthan | < |
| | Equal | = |
| | UNEQual | ≠ |
| Return parameter | Returns the pulse runt trigger condition. | |
| Example | :TRIGger:RUNT:WHEn UNEQual | |
| | Sets the Puls (≠). | e Runt trigger condition to unequal |
| | | Set → |
| :TRIGger:RUNT:TIMe | | ——Query |
| Description | Sets or queries the Pulse Runt trigger time. | |
| Syntax | :TRIGger:RUNT:TIMe { <nrf> ? }</nrf> | |



| Related commands | :TRIGger:TYPe :TRIGger:RUNT:WHEn | |
|---------------------|-------------------------------------|-----------------------------------|
| Parameter | <nrf></nrf> | Pulse runt time (4nS to 10S) |
| Return Parameter | <nr3></nr3> | Returns the runt time in seconds. |
| Example | :TRIGger:RUNT:TIMe 4.00E-5 | |
| | Sets the runt time to 40.0uS. | |

:TRIGger:RISEFall:SLOP



| Description | Sets or queries the Rise & Fall slope. | |
|------------------|---|--------------------------------|
| Syntax | :TRIGger:RISEFall:SLOP {RISe FALL EITher ?} | |
| Parameter | RISe Rising slope | |
| | FALL | Falling slope |
| | EITher | Either rising or falling slope |
| Return parameter | er Returns the rise & fall slope. | |

:TRIGger:RISEFall:SLOP RISe Example

Sets the Rise & Fall slope to rising.

:TRIGger:RISEFall:WHEn



| Description | Sets or queries the rise/fall trigger conditions. | |
|---------------------|---|---|
| Syntax | :TRIGger:RISEFall:WHEn {MOREthan LESSthan EQual UNEQual ? } | |
| Related commands | :TRIGger:TYPe | |
| | :TRIGger:RISEFall:TIMe | |
| Parameter | MOREthan | > |
| | LESSthan | < |
| | Equal | = |
| | UNEQual | ≠ |

Return parameter Returns the rise/fall trigger condition.



| Example | :TRIGger:RISEFall:WHEn UNEQual |
|---------|--|
| | Sets the Rise and Fall trigger condition to unequal (≠). |

:TRIGger:RISEFall:TIMe → Query

| Description | Sets or queries the Rise and Fall time. | |
|------------------|--|-----------------------------------|
| Syntax | :TRIGger:RISEFall:TIMe { <nrf> ? }</nrf> | |
| Related | :TRIGger:TYPe | |
| commands | :TRIGger:RISEFall:WHEn | |
| Parameter | <nrf></nrf> | Rise and Fall time (4nS to 10S) |
| Return Parameter | <nr3></nr3> | Returns the rise and fall time in |
| | | seconds. |
| Example | :TRIGger:RISEFall:TIMe 4.00E-5 | |
| | Sets the trigger rise & fall to 40.0us. | |

$\begin{array}{ccc} & & & & & & \\ & & & & \\ :TRIGger:VIDeo:TYPe & & & & & \\ & & & & & \\ \hline & & & & \\ Query & & & \\ \end{array}$

| Description | Sets or queries the video trigger type. | |
|---------------------|--|--------------------------|
| Syntax | :TRIGger:VIDeo:TYPE {NTSC PAL SECam EDTV480P EDTV576P HDTV720P HDTV1080I HDTV1080P ? } | |
| Related commands | :TRIGger:TYPe | |
| Parameter | NTSC | NTSC |
| | PAL | PAL |
| | SECam | SECAM |
| | EDTV480P | Extra definition TV 480P |
| | EDTV576P | Extra definition TV 576P |
| | HDTV720P | High definition TV 720P |
| | HDTV1080I | High definition TV 1080i |



| | HDTV1080P | High definition TV 1080P | |
|------------------|--|--|--|
| Return parameter | Returns the video trigger type. | | |
| Example | :TRIGger:VIDeo:TYPe NTSC | | |
| | Sets the vide | o trigger to NTSC. | |
| | | Set → | |
| :TRIGger:VIDeo | o:FIELd | → Query | |
| Description | Sets or queri | es the video trigger field. | |
| Syntax | :TRIGger:VID ALLLines ? } | eo:FIELd { FIELD1 FIELD2 ALLFields | |
| Related commands | :TRIGger:TYP | :TRIGger:TYPe | |
| Parameter | FIELD1 | Trigger on field 1 | |
| | FIELD2 | Trigger on field 2 | |
| | ALLFields | Trigger on all fields | |
| | ALLLines | Trigger on all lines | |
| Return parameter | Returns the video trigger field. | | |
| Example | :TRIGger:VIDeo:FIELd ALLFields | | |
| | Sets the video trigger to trigger on all fields. | | |
| | | <u>Set</u> → | |
| :TRIGger:VIDe | o:LINe | → Query | |
| Description | Sets or queries the video trigger line. | | |
| Syntax | :TRIGger:VIDeo:LINe { <nr1> ?}</nr1> | | |
| Related commands | :TRIGger:TYPe | | |
| Parameter | <nr1></nr1> | Video line | |
| Return parameter | <nr3></nr3> | Returns the video trigger line. | |
| Example | :TRIGger:VID | eo:LINe 1 | |
| | Sets the video trigger to line 1. | | |



| :TRIGger:VIDec | o:POLarity | | Set → Query |
|------------------|--|-----------------------|---------------------|
| Description | Sets or queri | es the video trigger | polarity. |
| Syntax | :TRIGger:VID | eo:POLarity { POSitiv | re NEGative ? } |
| Related commands | :TRIGger:TYP | e | |
| Parameter | POSitive | Positive polarity | |
| | NEGative | Negative polarity | |
| Return parameter | Returns the | video trigger polarit | у. |
| Example | :TRIGger:VID | eo:POLarity POSitive | |
| | Sets the video trigger polarity to positive. | | positive. |
| :TRIGger:PULS | e:WHEn | | Set → Query |
| Description | Sets or queries the pulse width trigger conditions. | | |
| Syntax | :TRIGger:PULSe:WHEn { MOREthan LESSthan EQual UNEQual ? } | | |
| Related | :TRIGger:TYPe | | |
| commands | :TRIGger:PULSe:TIMe | | |
| Parameter | MORE than | > | |
| | LESSthan | < | |
| | EQual | = | |
| | UNEQual | ≠ | |
| Return parameter | Returns the 1 | pulse width trigger | conditions. |
| Example | :TRIGger:PULSe:WHEn UNEQual | | |
| | Sets the trigger pulse width conditions to not equal to (\neq) . | | |



| :TRIGger:PULS | e:TIMe | Set → (Query) |
|--------------------------------|--|--|
| Description | | |
| | | ies the pulse width time. |
| Syntax | | LSe:TIMe { <nrf> ?}</nrf> |
| Related commands | :TRIGger:TYF | |
| Commands | :TRIGger:PUI | LSe:WHEn |
| Parameter | <nrf></nrf> | Pulse width time (4ns~10s) |
| Return parameter | <nr3></nr3> | Returns the pulse width time in seconds. |
| Example | :TRIGger:PUI | LSe:TIMe 4.00E-5 |
| | Sets the trigg | ger pulse width to 40.0us. |
| | | (Set)→ |
| :TRIGger:TIME | Out:WHEn | Query |
| Description | Sets or queries the timeout trigger condition. | |
| Syntax | :TRIGger:TIMEOut:WHEn {HIGH LOW EITher ?} | |
| Related commands | :TRIGger:TIMEOut:TIMER | |
| Parameter | HIGH | Signal is high. |
| | LOW | Signal is low. |
| | EITher | Signal is high or low. |
| Return parameter | Returns the timeout condition (HIGH, LOW, EITHER). | |
| Example1 | :TRIGger:TIMEOut:WHEn LOW | |
| | Sets the time | eout condition to low. |
| | | (Set)→ |
| :TRIGger:TIMEOut:TIMER → Query | | |
| Description | Sets or returns timeout trigger time. | |
| Syntax | :TRIGger:TIMEOut:TIMER { <nrf> ? }</nrf> | |



| B. I I | TDIC TI | 450 | |
|------------------|--|--|--|
| Related commands | :TRIGger:TIMEOut:WHEn | | |
| Parameter | <nrf> Timeout time. (4nS to 10S).</nrf> | | |
| Return parameter | Returns the | timeout time as <nr3>.</nr3> | |
| Example | :TRIGger:TIN | 1EOut:TIMER? | |
| | 8.960e-05 | | |
| | | Set → | |
| :TRIGger:ALTe | rnate | → Query | |
| Description | Sets alternating between source triggers on or off or queries its state. | | |
| Syntax | :TRIGger:ALTernate {OFF ON ?} | | |
| Parameter | OFF | Alternate off | |
| | ON | Alternate on | |
| Return parameter | Returns the Alternate trigger status (ON, OFF). | | |
| Example | :TRIGger:ALTernate ON | | |
| | Turns on alternating between source triggers. | | |
| | | | |
| :TRIGger:STAT | :TRIGger:STATe → Query | | |
| Description | Returns the current state of the triggering system. | | |
| Syntax | :TRIGger:STATe? | | |
| Return parameter | *ARMED | Indicates that the oscilloscope is acquiring pretrigger information. | |

*AUTO

*READY

| acquiring pretrigger information. |
|---|
| Indicates that the oscilloscope is in the automatic mode and acquires data even in the absence of a trigger. |
| Indicates that all pretrigger information has been acquired and that the oscilloscope is ready to accept a trigger. |
| |



| | *SAVE | Indicates that the oscilloscope is in save mode and is not acquiring data. |
|---------|--------------|--|
| | *TRIGGER | Indicates that the oscilloscope triggered and is acquiring the post trigger information. |
| Example | :TRIGger:ST/ | ATe? |
| | AUTO | |

The trigger is in auto mode.

:TRIGger:EXTERnal:PROBe:TYPe

| | Set)- | → |
|---|--------|----------|
| _ | → Que | rv) |

| Description | Sets or queries the external probe type. | |
|---------------------|--|---------|
| Syntax | :TRIGger:EXTERnal:PROBe:TYPe { VOLTage CURRent ? } | |
| Related commands | :TRIGger:EXTERnal:PROBe:RATio | |
| Parameter | VOLTage | Voltage |
| | CURRent | Current |
| Return parameter | Returns the probe type. | |
| Example | :TRIGger:EXTERnal:PROBe:TYPe? CURRENT | |

:TRIGger:EXTERnal:PROBe:RATio



| Description | Sets or queries the external probe ratio (attenuation). | |
|---------------------|---|---------------------------------------|
| Syntax | :TRIGger:EXTERnal:PROBe:RATio { <nrf> ?}</nrf> | |
| Related commands | :TRIGger:EXTERnal:PROBe:TYPe | |
| Parameter | <nrf></nrf> | External probe attenuation factor. |
| Return parameter | <nr3></nr3> | Returns the probe attenuation factor. |



Example :TRIGger:EXTERnal:PROBe:RATio?

5.000000e+01

:TRIGger:BUS:TYPe



| Description | Returns the current bus type. | |
|------------------|-------------------------------|-----------|
| Syntax | :TRIGger:BUS:TYPe? | |
| Return parameter | 12C I ² C mode | |
| | SPI | SPI mode |
| | UART | UART mode |
| | CAN | CAN mode |
| | LIN | LIN mode |
| Example | :TRIGger:BUS:TYPe? | |

:TRIGger:BUS:THReshold:CH<x>

UART



| Description | Sets or queries the threshold level for the selected channel. |
|-------------|---|
| | |

| Syntax | :TRIGger:BUS:THReshold:CH <x> {<nr3> ?}</nr3></x> | |
|------------------|---|-----------------------------|
| | <x> CH1 ~ CH4</x> | |
| | <nr3></nr3> | Threshold level |
| Return Parameter | <nr3></nr3> | Returns the threshold level |
| Example | :TRIGger:BUS:THReshold:CH1 1 | |

Sets the CH1 threshold to 1V.





Description Sets or queries the I²C trigger conditions.



| Syntax | :TRIGger:BUS:B1:I2C:CONDition {STARt STOP REPEATstart ACKMISS ADDRess DATA ADDRANDDATA ? } | | |
|------------------|--|---------|--|
| Parameter | STARt | | Set Start as the I ² C trigger condition. |
| | STOP | | Set Stop as the I ² C trigger condition. |
| | REPEATstart | | Set Repeat of Start as the I ² C trigger condition. |
| | ACKMISS | | Set Missing Acknowledgement as the I ² C trigger condition. |
| | ADDRess | | Set Address as the I ² C trigger condition. |
| | DATA | | Set Data as the I ² C trigger condition. |
| | ADDRANDDA | ATA | Set Address and Data as the I ² C trigger condition. |
| Return parameter | Returns the I ² C bus trigger condition. | | |
| Example | :TRIGger:BUS:B1:I2C:CONDition ADDRess | | |
| | Set Address as the I2C trigger condition. | | |
| | Set → | | |
| :TRIGger:BUS:I | 31:I2C:ADD | Ress: | MODe → Query |
| Description | Sets or queries the I ² C addressing mode (7 or 10 bits). | | |
| Syntax | :TRIGger:BUS:B1:I2C:ADDRess:MODe {ADDR7 ADDR10 ? } | | |
| Related commands | :TRIGger:BUS:B1:I2C:CONDition | | |
| Parameter | ADDR7 | 7 bit a | addressing |
| | ADDR10 | 10 bit | addressing |
| Return Parameter | 0 7 bit addressing | | |



| | 1 | 10 bit addressing | | |
|---|--|---|--|--|
| Example | :TRIGger:BUS:B1:I2C:ADDRess:MODe? | | | |
| | The addressing mode is currenty set to 7 bits. | | | |
| | | (Set)→ | | |
| :TRIGger:BUS: | :TRIGger:BUS:B1:I2C:ADDRess:TYPe → Query | | | |
| Description | Sets the I ² C bus address type, or queries what the setting is. | | | |
| Syntax | :TRIGger:BUS:B1:I2C:ADDRess:TYPe {GENeralcall STARtbyte HSmode EEPROM CBUS ?} | | | |
| Related commands | :TRIGger:BUS:B1:I2C:CONDition | | | |
| Parameter | GENeralcall | Set a general call address (0000 000 0). | | |
| | STARtbyte | Set a start byte address. (0000 000 1) | | |
| | HSmode | Set a high-speed mode address. (0000 1xx x) | | |
| | EEPROM | Set an EEPROM address. (1010 xxx x) | | |
| | CBUS | Set a CBUS address. (0000 001 x) | | |
| Return Parameter | Returns the address type | | | |
| Example | :TRIGger:BUS:B1:I2C:ADDRess:TYPe? CBUS | | | |
| :TRIGger:BUS:B1:I2C:ADDRess:VALue Set → Query | | | | |
| Description | Sets or queries the I ² C bus address value when the I ² C bus is set to trigger on Address or Address/Data. | | | |
| Syntax | :TRIGger:BUS:B1:I2C:ADDRess:VALue { <string> ? }</string> | | | |
| Related commands | :TRIGger:BUS:B1:I2C:ADDRess:MODe | | | |



| Parameter | <sting></sting> | 7/10 characters, must be enclosed in double quotes, "string". | |
|------------------|--|---|--|
| | | x = don't care | |
| | | 1 = binary 1 | |
| | | 0 = binary 0 | |
| Return Parameter | Returns the | address value. | |
| Example1 | :TRIGger:BUS:B1:I2C:ADDRess:MODe ADDR7 | | |
| | :TRIGger:BUS:B1:I2C:ADDRess:VALue "xxx0101" | | |
| | Sets the address to XXX0101 | | |
| Example 2 | :TRIGger:BUS:B1:I2C:ADDRess:VALue? XXX0101 | | |
| | | Set → | |
| :TRIGger:BUS: | B1:I2C:ADD | Ress:DIRection → Query | |
| Description | Sets or queries the address bit as read write or don't care. | | |
| Note | This setting only applies when the I ² C trigger is set to trigger on Address or Address/Data | | |
| Syntax | :TRIGger:BUS:B1:I2C:ADDRess:DIRection { READ WRITE NOCARE ? } | | |
| Related commands | :TRIGger:BUS | S:B1:I2C:CONDition | |
| Parameter | READ | Set read as the data direction. | |
| | WRITE | Set write as the data direction. | |
| | NOCARE | Set either as the data direction. | |
| Return Parameter | Returns the direction (READ, WRITE, NOCARE). | | |
| Example | :TRIGger:BUS:B1:I2C:ADDRess:DIRection READ | | |
| - | Sets the direction to READ. | | |
| | | Set → | |
| :TRIGger:BUS: | :TRIGger:BUS:B1:I2C:DATa:SIZe → Query | | |



| Description | Sets or queries the data size in bytes for the I ² C bus. | | |
|---------------------|---|--|--|
| Note | This setting only applies when the I ² C trigger is set to trigger on Data or Address/Data | | |
| Syntax | :TRIGger:BUS:B1:I2C:DATa:SIZe { <nr1> ? }</nr1> | | |
| Related commands | :TRIGger:BUS:B1:I2C:CONDition | | |
| Parameter | <nr1> Number of data bytes (1 to 5).</nr1> | | |
| Return parameter | <nr1></nr1> | Returns the number of bytes. | |
| Example | :TRIGger:BUS | S:B1:I2C:DATa:SIZe 3 | |
| | Sets the num | aber of bytes to 3. | |
| | | Set | |
| :TRIGger:BUS:I | B1:I2C:DATa | a:VALue → Query | |
| Description | Sets or queries the triggering data value for the I ² C bus when the I ² C bus is set to trigger on Data or Address/Data. | | |
| Syntax | :TRIGger:BUS:B1:I2C:DATa:VALue { <string> ? }</string> | | |
| Related commands | :TRIGger:BUS:B1:I2C:DATa:SIZe | | |
| Parameter | <sting></sting> | The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string". x = don't care 1 = binary 1 | |
| | | 0 = binary 0 | |
| Return Parameter | Returns the data value. | | |
| Example1 | :TRIGger:BUS:B1:I2C:DATa:SIZe 1 | | |
| | :TRIGger:BUS | S:B1:I2C:DATa:VALue "1x1x0101" | |
| | Sets the valu | e to XXX0101 | |



Example 2 :TRIGger:BUS:B1:I2C:DATa:VALue?

1X1X0101

:TRIGger:BUS:B1:UART:CONDition



| Description | Sets or queries the UART triggering condition. | | |
|-------------|--|--|--|
| Syntax | :TRIGger:BUS:B1:UART:CONDition { RXSTArt RXDATA RXENDPacket TXSTArt TXDATA TXENDPacket TXPARItyerr RXPARItyerr ? } | | |
| Parameter | RXSTArt | Set trigger on the RX Start Bit. | |
| | RXDATA | Set trigger on RX Data. | |
| | RXENDPacket | Set trigger on the RX End of Packet condition. | |
| | RXPARItyerr | Set trigger on RX Parity error condition. | |
| | TXSTArt | Set trigger on the TX Start Bit. | |
| | TXDATA | Set trigger on TX Data. | |
| | TXENDPacket | Set trigger on the TX End of Packet condition. | |
| | TXPARItyerr | Set trigger on TX Parity error condition. | |

Return Parameter Returns the triggering condition.

Example :TRIGger:BUS:B1:UART:CONDition TXDATA

Sets the UART bus to trigger on Tx Data.



: TRIGger: BUS: B1: UART: RX: DATa: SIZe

| Description | Sets or queries the number of bytes for UART data. |
|-------------|--|
| Note | This setting only applies when the UART trigger is set to trigger on Rx Data |
| Syntax | :TRIGger:BUS:B1:UART:RX:DATa:SIZe { <nr1> ?}</nr1> |



| Related commands | :TRIGger:BUS:B1:UART:CONDition | | |
|---------------------------------|---|--|--|
| Parameter | <nr1></nr1> | Number of bytes (1 to 10). | |
| Return parameter | <nr1></nr1> | Returns the number of bytes. | |
| Example | :TRIGger:BUS:B1:UART:RX:DATa:SIZe 5 Sets the number of bytes to 5. | | |
| :TRIGger:BUS:I | B1:UART:RX | :DATa:VALue → Query | |
| Description | Sets or queries the triggering data value for the UART bus when the bus is set to trigger on Rx Data. | | |
| Syntax | :TRIGger:BUS:B1:UART:RX:DATa:VALue { <string> ? }</string> | | |
| Related commands | :TRIGger:BUS | S:B1:UART:RX:DATa:SIZe | |
| Parameter | <sting></sting> | The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string". | |
| | | x = don't care | |
| | | 1 = binary 1 | |
| | | 0 = binary 0 | |
| Return Parameter | Returns the data value. | | |
| Example1 | :TRIGger:BUS:B1:UART:CONDition RXDATA | | |
| :TRIGger:BUS:B1:UART:RX:DATa:SI | | S:B1:UART:RX:DATa:SIZe 1 | |
| | :TRIGger:BUS:B1:UART:RX:DATa:VALue "1x1x01 Sets the value to 1x1x0101 | | |
| | | | |
| Example 2 | :TRIGger:BUS:B1:UART:RX:DATa:VALue? | | |
| | 1X1X0101 | | |
| | | Set → | |
| :TRIGger:BUS:I | B1:UART:TX | :DATa:SIZe → Query | |



| Description | Sets or queries the number of bytes for UART data. | | |
|------------------|---|--|--|
| Note | This setting only applies when the UART trigger is set to trigger on Tx Data | | |
| Syntax | :TRIGger:BUS:B1:UART:TX:DATa:SIZe { <nr1> ?}</nr1> | | |
| Related commands | :TRIGger:BUS:B1:UART:CONDition | | |
| Parameter | <nr1></nr1> | Number of bytes (1 to 10). | |
| Return parameter | <nr1></nr1> | Returns the number of bytes. | |
| Example | :TRIGger:BUS:B1:UART:TX:DATa:SIZe 5 Sets the number of bytes to 5. | | |
| | | Set | |
| :TRIGger:BUS: | B1:UART:TX | ∷DATa:VALue → Query | |
| Description | Sets or queries the triggering data value for the UART bus when the bus is set to trigger on Tx Data. | | |
| Syntax | :TRIGger:BUS:B1:UART:TX:DATa:VALue { <string> ? }</string> | | |
| Related commands | :TRIGger:BUS:B1:UART:TX:DATa:SIZe | | |
| Parameter | <sting></sting> | The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string". | |
| | | x = don't care | |
| | | 1 = binary 1 | |
| | | 0 = binary 0 | |
| Return Parameter | Returns the | data value. | |
| Example1 | :TRIGger:BUS:B1:UART:CONDition TXDATA | | |
| | :TRIGger:BUS:B1:UART:TX:DATa:SIZe 1 | | |
| | :TRIGger:BUS:B1:UART:TX:DATa:VALue "1x1x0101" | | |
| | Sets the value to 1x1x0101 | | |



| Example 2 | :TRIGger:BUS:B1:UART:TX:DATa:VALue? 1X1X0101 | | |
|------------------|---|---|--|
| :TRIGger:BUS: | 31:SPI:CON | Set → Dition — Query | |
| Description | Sets or queries the SPI triggering condition. | | |
| Syntax | :TRIGger:BUS:B1:SPI:CONDition {SS MISO MOSI MISOMOSI ? } | | |
| Parameter | SS | Set to trigger on the Slave Selection condition. | |
| | MISO | Set to trigger on the Master-In Slave-Out condition. | |
| | MOSI | Set to trigger on the Master-Out Slave-In condition. | |
| | MISOMOSI | Set to trigger on the Master-In Slave-Out and Master-Out Slave-In conditions. | |
| Return Parameter | Returns the t | riggering condition. | |
| Example | :TRIGger:BUS | :B1:SPI:CONDition MISO | |
| | Sets the SPI bus to trigger on MISO. | | |
| | | Set | |
| :TRIGger:BUS: | 31:SPI:DATa | :SIZe → Query | |
| Description | Sets or querio | es the number of words for SPI data. | |
| Note | This setting only applies when the SPI trigger is set to trigger on MISO, MOSI or MISO/MOSI | | |
| Syntax | :TRIGger:BUS | :B1:SPI:DATa:SIZe { <nr1> ?}</nr1> | |
| Related commands | :TRIGger:BUS:B1:SPI:CONDition | | |
| Parameter | <nr1></nr1> | Number of words (1 to 32). | |
| Return parameter | <nr1></nr1> | Returns the number of words. | |



| Example | :TRIGger:BUS:B1:SPI:DATa:SIZe 10 Sets the number of words to 10. | | |
|------------------|---|--|--|
| | | (Set)→ | |
| :TRIGger:BUS: | B1:SPI:DATa | a:MISO:VALue → Query | |
| Description | - | es the triggering data value for the SPI e bus is set to trigger on MISO or I. | |
| Syntax | :TRIGger:BUS ? } | S:B1:SPI:DATa:MISO:VALue { <string></string> | |
| Related commands | :TRIGger:BUS | S:B1:SPI:DATa:SIZe | |
| Parameter | <sting></sting> | The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string". | |
| | | x = don't care | |
| | | 1 = binary 1 | |
| | | 0 = binary 0 | |
| Return Parameter | Returns the | data value. | |
| Example1 | :TRIGger:BUS:B1:SPI:CONDition MISO | | |
| | :TRIGger:BUS:B1:SPI:DATa:SIZe 2 | | |
| | :TRIGger:BUS | S:B1:SPI:DATa:MISO:VALue "1x1x0101" | |
| | Sets the valu | e to 1x1x0101 | |
| Example 2 | :TRIGger:BUS | S:B1:SPI:DATa:MISO:VALue? | |
| • | 1X1X0101 | | |



| :TRIGger:BUS: | B1:SPI:DATa | a:MOSI:VALue → Query |
|---------------------|-----------------------|--|
| Description | | es the triggering data value for the SPI e bus is set to trigger on MOSI or I. |
| Syntax | :TRIGger:BUS ? } | S:B1:SPI:DATa:MOSI:VALue { <string></string> |
| Related commands | :TRIGger:BUS | S:B1:SPI:DATa:SIZe |
| Parameter | <sting></sting> | The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string". |
| | | x = don't care |
| | | 1 = binary 1 |
| | | 0 = binary 0 |
| Return Parameter | Returns the | data value. |
| Example1 | :TRIGger:BUS | S:B1:SPI:CONDition MOSI |
| | :TRIGger:BUS | S:B1:SPI:DATa:SIZe 2 |
| | :TRIGger:BUS | S:B1:SPI:DATa:MOSI:VALue "1x1x0101" |
| | Sets the valu | e to 1x1x0101 |
| Example2 | :TRIGger:BUS | S:B1:SPI:DATa:MOSI:VALue? |
| | 1X1X0101 | |
| :TRIGger:BUS: | B1:CAN:CO | $ \begin{array}{ccc} \text{Set} & & \\ \text{NDition} & & & \\ & & & \\ \end{array} $ |
| Description | Sets or retur | ns the CAN trigger condition. |
| Syntax | | S:B1:CAN:CONDition Etype IDentifier DATA IDANDDATA EOF UFFERR ?} |
| Parameter/ | SOF | Triggers on a start of frame |



| Return parameter | EDANAEL | Tributan and the transport forms | |
|------------------|---|--|--|
| рагание. | FRAMEtype | Triggers on the type of frame | |
| | Identifier | Triggers on a matching identifier | |
| | DATA | Triggers on matching data | |
| | IDANDDATA | Triggers on matching identifier and data field | |
| | EOF | Triggers on the end of frame | |
| | ACKMISS | Triggers on a missing acknowledge | |
| | STUFFERR | Triggers on a bit stuffing error | |
| Example1 | :TRIGger:BUS:B1:CAN:CONDition SOF | | |
| | Triggers on a | a start of frame. | |
| Example2 | :TRIGger:BUS:B1:CAN:CONDition? >SOF | | |
| | | Set | |
| :TRIGger:BUS:I | B1:CAN:FRA | MEtype → Query | |
| Description | Sets or returns the frame type for a CAN FRAMEType trigger. | | |
| Syntax | :TRIGger:BUS:B1:CAN:FRAMEtype {DATA REMote ERRor OVERLoad ?} | | |
| Parameter/ | DATA | Sets the frame type to data frame | |
| Return parameter | REMote | Sets the frame type to remote frame | |
| | ERRor | Sets the frame type to error frame | |
| | OVERLoad | Sets the frame type to overload | |
| Example | :TRIGger:BUS:B1:CAN:FRAMEtype DATA | | |
| | Sets the frame type to DATA. | | |
| | | | |
| | | Set | |
| :TRIGger:BUS:I | B1:CAN:IDe | | |



| Syntax | :TRIGger:BUS:B1:CAN:IDentifier:MODe {STANDard EXTended ?} | | |
|--------------------------------|--|---|--|
| Parameter/ | STANDard | Standard addressing mode | |
| Return parameter | EXTended | Extended addressing mode | |
| Example | :TRIGger:BUS:B1:CAN:IDentifier:MODe? >STANDARD | | |
| | Returns the | addressing mode. | |
| :TRIGger:BUS: | B1:CAN:IDe | ntifier:VALue — Query | |
| Description | Sets or returns the identifier string used for the CAN trigger. Note: Only applicable when the trigger condition is set to ID or IDANDDATA. | | |
| | | | |
| Syntax | :TRIGger:BU | 5:B1:CAN:IDentifier:VALue { <string> ?}</string> | |
| Related Commands | :TRIGger:BUS:B1:CAN:IDentifier:MODe | | |
| Parameter/ Return parameter | <string></string> | The size of the string depends on the data size setting. The string must be enclosed in double quotes, "string". | |
| | | String contents: | |
| | | x = don't care | |
| | | A GIOTI COMP | |
| | | 1 = binary 1 | |
| | | | |
| Example | :TRIGger:BUS | 1 = binary 1 | |
| Example | _ | 1 = binary 1 0 = binary 0 | |
| Example | :TRIGger:BU | 1 = binary 1 0 = binary 0 S:B1:CAN:CONDition ID S:B1:CAN:IDentifier:MODe STANDARD S:B1:CAN:IDentifier:VALue | |
| Example | :TRIGger:BUS :TRIGger:BUS "01100X1X01 | 1 = binary 1 0 = binary 0 S:B1:CAN:CONDition ID S:B1:CAN:IDentifier:MODe STANDARD S:B1:CAN:IDentifier:VALue | |



| :TRIGger:BUS:I | 31:CAN:IDe | Set → ntifier:DIRection → Query | | |
|--------------------------------|---|---|--|--|
| Description | Sets or queries the address bit as read, write or don't care. | | | |
| Syntax | :TRIGger:BUS:B1:CAN:IDentifier:DIRection {READ WRITE NOCARE ?} | | | |
| Parameter/ | READ | Sets read as the data direction | | |
| Return parameter | WRITE | Sets write as the data direction | | |
| | NOCARE | Sets either as the data direction | | |
| Example1 | :TRIGger:BUS >WRITE | :TRIGger:BUS:B1:CAN:IDentifier:DIRection? | | |
| Example2 | :TRIGger:BUS | S:B1:CAN:IDentifier:DIRection READ | | |
| | :TRIGger:BUS > READ | S:B1:CAN:IDentifier:DIRection? | | |
| :TRIGger:BUS:I | B1:CAN:DA | Set → Fa:QUALifier → Query | | |
| Description | Note: Only a | ns the CAN data qualifier. pplicable when the triggering set to DATA or IDANDDATA. | | |
| Syntax | :TRIGger:BUS:B1:CAN:DATa:QUALifier {LESSthan MOREthan EQual UNEQual LESSEQual M OREEQual ?} | | | |
| Parameter/ Return parameter | LESSthan | Triggers when the data is less than the qualifier value. | | |
| | MOREthan | Triggers when the data is greater than the qualifier value. | | |
| | EQual | Triggers when the data is equal to the qualifier value. | | |
| | UNEQual | Triggers when the data is not equal to the qualifier value. | | |



| | LESSEQual | Triggers when the data is less than or equal to the qualifier value. | |
|--------------------------------|--|--|--|
| | MOREEQual | Triggers when the data is more than or equal to the qualifier value. | |
| Example | :TRIGger:BUS >EQUAL | 5:B1:CAN:DATa:QUALifier? | |
| | :TRIGger:BUS | S:B1:CAN:DATa:QUALifier MOREthan | |
| | :TRIGger:BUS >MOREthan | 5:B1:CAN:DATa:QUALifier? | |
| | | Set → | |
| :TRIGger:BUS:I | B1:CAN:DA | Ta:SIZe → Query | |
| Description | for a CAN tr | applicable when the condition is set to | |
| Syntax | :TRIGger:BUS | 5:B1:CAN:DATa:SIZe { <nr1> ?}</nr1> | |
| Parameter/ Return parameter | <nr1></nr1> | 1~8 (bytes) | |
| Example | :TRIGger:BUS | S:B1:CAN:DATa:SIZe? | |
| | :TRIGger:BUS | S:B1:CAN:DATa:SIZe 2 | |
| | :TRIGger:BUS >2 | S:B1:CAN:DATa:SIZe? | |
| | | Set → | |
| :TRIGger:BUS:I | B1:CAN:DA | Ta:VALue ———————————————————————————————————— | |
| Description | Description Sets or returns the binary data string to be used a CAN trigger. | | |
| | Note: Only a DATA or ID | applicable when the condition is set to ANDDATA. | |
| Related Commands | :TRIGger:BUS | S:B1:CAN:DATa:SIZe | |
| Syntax | :TRIGger:BUS | 5:B1:CAN:DATa:VALue { <string> ?}</string> | |



| Parameter/ Return parameter | <string></string> | The size of the string depends on the data size setting. The string must be enclosed in double quotes, "string". |
|--------------------------------|---------------------------|--|
| | | String contents: |
| | | x = don't care |
| | | 1 = binary 1 |
| | | 0 = binary 0 |
| Example | :TRIGger:BUS | S:B1:CAN:DATa:SIZe 1 |
| | :TRIGger:BUS | 5:B1:CAN:DATa:VALue "01010X1X" |
| | :TRIGger:BUS >01010X1X | 5:B1:CAN:DATa:VALue? |
| | | (Set)→ |
| :TRIGger:BUS: | B1:LIN:CON | IDition → Query |
| Description | Sets or retur | ns the LIN trigger condition. |
| Syntax | • | 5:B1:LIN:CONDition Dentifier DATA IDANDDATA WAKEup ?} |
| Parameter/ Return parameter | SYNCField | Sets the LIN trigger condition to the sync field. |
| | IDentifier | Sets the LIN trigger condition to identifier field. |
| | DATA | Sets the LIN trigger condition to the data field. |
| | IDANDDATA | Sets the LIN trigger condition to identifier and data field |
| | WAKEup | Sets the LIN trigger condition to wake up. |
| | SLEEP | Sets the LIN trigger condition to sleep. |

error.

Sets the LIN trigger condition to

ERRor



Example :TRIGger:BUS:B1:LIN:CONDition?

>IDANDDATA

 $: TRIGger: BUS: B1: LIN: CONDition\ DATA$

:TRIGger:BUS:B1:LIN:CONDition?

>DATA



| :TRIGger:BUS:I | B1:LIN:DAT | a:QUALifier | Set → Query |
|---------------------------------------|---|--|--|
| Description | Sets or returns the LIN data qualifier. Note: Only applicable when the trigger condition is set to DATA or IDANDDATA. | | |
| Syntax | :TRIGger:BUS:B1:LIN:DATa:QUALifier {LESSthan MOREthan EQual UNEQual LESSEQual M OREEQual ?} | | |
| Parameter/ Return parameter | LESSthan | Triggers when the the qualifier value. | |
| | MOREthan | Triggers when the the qualifier value. | data is greater than |
| | EQual | Triggers when the qualifier value. | data is equal to the |
| | UNEQual | Triggers when the the qualifier value. | data is not equal to |
| | LESSEQual | Triggers when the equal to the qualifi | |
| | MOREEQual | Triggers when the or equal to the qua | |
| Example | :TRIGger:BUS:B1:LIN:DATa:QUALifier? >EQUAL | | |
| | :TRIGger:BUS | S:B1:LIN:DATa:QUAL | ifier MOREthan |
| | :TRIGger:BUS >MORETHAN | 5:B1:LIN:DATa:QUAL N | ifier? |
| | | | Set → |
| :TRIGger:BUS:B1:LIN:DATa:SIZe → Query | | | → Query |
| Description | for the LIN t | rigger. pplicable when the | data string in bytes condition is set to |
| Syntax | :TRIGger:BUS | S:B1:LIN:DATa:SIZe { | [<nr1> ?}</nr1> |



| Parameter/ Return parameter | <nr1></nr1> | 1~8 (bytes) |
|--------------------------------|--|--|
| Example | :TRIGger:BUS | S:B1:LIN:DATa:SIZe? |
| | :TRIGger:BUS | S:B1:LIN:DATa:SIZe 2 |
| | :TRIGger:BUS >2 | S:B1:LIN:DATa:SIZe? |
| | | Set → |
| :TRIGger:BUS: | B1:LIN:DAT | a:VALue → Query |
| Description | Sets or retur the LIN trigg | ns the binary data string to be used for ger. |
| | • | applicable when the condition is set to ANDDATA. |
| Related Commands | :TRIGger:BU | S:B1:LIN:DATa:SIZe |
| Syntax | :TRIGger:BU | S:B1:LIN:DATa:VALue { <string> ?}</string> |
| Parameter/ Return parameter | <string></string> | The size of the string depends on the data size setting. The string must be enclosed in double quotes, "string". |
| | | String contents: |
| | | x = don't care |
| | | 1 = binary 1 |
| | | 0 = binary 0 |
| Example | :TRIGger:BUS | S:B1:LIN:DATa:SIZe 1 |
| | :TRIGger:BUS | S:B1:LIN:DATa:VALue "01010X1X" |
| | :TRIGger:BUS:B1:LIN:DATa:VALue? >01010X1X | |
| | | Set → |
| :TRIGger:BUS:I | B1:LIN:ERR | TYPE → Query |
| Description | Sets or retur trigger. | ns the error type be used for the LIN |



| Syntax | :TRIGger:BUS:B1:LIN:ERRTYPE {SYNC PARIty CHecksum ?} | | |
|--------------------------------|--|--|--|
| Parameter/ | SYNC | Sets the LIN error type to SYNC. | |
| Return parameter | PARIty | Sets the LIN error type to parity. | |
| | CHecksum | Sets the LIN error type to checksum. | |
| Example | :TRIGger:BUS:B1:LIN:ERRTYPE? | | |
| | :TRIGger:BUS | S:B1:LIN:ERRTYPE CHECKSUM | |
| | :TRIGger:BUS >CHECKSUM | S:B1:LIN:ERRTYPE? 1 | |
| | | (Set)→ | |
| :TRIGger:BUS:I | B1:LIN:IDer | itifier:VALue — Query | |
| Description | Sets or returns the identifier string to be used for the LIN trigger. Note: Only applicable when the condition is set to ID or IDANDDATA. | | |
| | | | |
| Syntax | :TRIGger:BU | S:B1:LIN:IDentifier:VALue { <string> ?}</string> | |
| Parameter/ Return parameter | <string></string> | The size of the string depends on the data size setting. The string must be enclosed in double quotes, "string". | |
| | | String contents: | |
| | | x = don't care | |
| | | 1 = binary 1 | |
| | | 0 = binary 0 | |
| Example | :TRIGger:BUS | 5:B1:LIN:CONDition ID | |
| | :TRIGger:BUS:B1:LIN:IDentifier:VALue "00X1X01X" | | |
| | :TRIGger:BUS:B1:LIN:IDentifier:VALue? >01100X1X01X | | |



System Commands

| :SYSTem:LOCK | 157 |
|---------------|-----|
| ·SYSTem·FRRor | 157 |



| Description | Turns the panel lock on off. | |
|-------------|------------------------------|-----------------|
| Syntax | :SYSTem:LOCK {OFF ON ? } | |
| Parameter | OFF | System lock off |
| | ON | System lock on |

Return parameter Returns the status of the panel lock (ON, OFF).

Example :SYSTem:LOCK ON

Turns the panel lock on.

:SYSTem:ERRor → Query

Description Queries the error queue. See the appendix on page 244 for details.

Syntax :SYSTem:ERRor?

Return parameter Returns the last message in the error queue.

Example :SYSTem:ERRor? +0, "No error."



Save/Recall Commands

| :RECAll:SETUp | 158 |
|----------------------------|-----|
| :RECAll:WAVEform | |
| :SAVe:IMAGe | 159 |
| :SAVe:IMAGe:FILEFormat | 159 |
| :SAVe:IMAGe:INKSaver | 160 |
| :SAVe:SETUp | 160 |
| :SAVe:WAVEform | 161 |
| :SAVe:WAVFform:FII FFormat | 162 |

:RECAll:SETUp



| Description | Recalls setu | Recalls setup settings from memory or USB. | |
|-------------|-----------------------|--|--|
| Syntax | | :RECAll:SETUp {S1~S20 <file path>("Disk:/xxx.SET","USB:/xxx.SET")}</file | |
| Parameter | S1~S20 | Recall Set1~Set20 | |
| | <file path=""></file> | Recall a file from the DSO internal files system or from a USB flash drive. | |
| Example | :RECAll:SETU | Jp S1 | |
| | Recalls setu | p setting S1 from memory. | |
| | :RECAll:SETU | Jp "Disk:/DS0001.SET" | |
| | Recall the se memory. | tup setting DS0001.SET from the internal | |

:RECAll:WAVEform



| Description | Recalls a waveform from wave1~wave20 or from file to REF1~4. |
|-------------|--|
| Note | Detail CSV files cannot be recalled. |
| Syntax | :RECAll:WAVEform {W <n> <file path=""> ("Disk:/xxx.LSF","USB:/xxx.LSF")},REF<x></x></file></n> |



| Parameter | n | 1~20 (Wave1~wave20) |
|-----------|---------------|--|
| | | Filename in file path. Example: "Disk:/xxx.LSF","USB:/xxx.LSF", "Disk:/xxx.CSV","USB:/xxx.CSV" |
| | <x></x> | 1,2,3,4 (REF1, REF2, REF3, REF4) |
| Example | :RECAll:WAVE | form W1, REF1 |
| | Recalls the w | raveform stored in Wave1 to reference |

| :SAVe:IMAGe | | Set → | |
|---------------------|---|--|--|
| Description | Saves a screen image to the assigned file path with a specified filename. | | |
| Syntax | :SAVe:IMAGe { <file path=""> ("Disk:/xxx.PNG", "USB:/xxx.BMP)}</file> | | |
| Related commands | :SAVe:IMAGe | | |
| Parameter | xxx.PNG or xxx.BMP | File name (8 characters max) | |
| Example | :SAVe:IMAGe | "Disk:/pic1.PNG" | |
| | | en image named pic1.png to the root (sk:/) of the scope. | |
| | :SAVe:IMAGe "USB:/pic1.BMP" | | |
| | Saves a screen image named pic1.bmp to the root directory of the external USB flash disk. | | |
| | | (Set)→ | |

:SAVe:IMAGe:FILEFormat → Query

| Description | Sets the file format for image. | |
|-------------|--|--|
| Syntax | :SAVe:IMAGe:FILEFormat {PNG BMP ?} | |
| Related | :SAVe:IMAGe | |
| commands | :SAVe:IMAGe:INKSaver | |



| Parameter | PNG | Sets the file format to PNG |
|------------------|------------------------------------|-----------------------------|
| | ВМР | Sets the file format to BMP |
| Return parameter | Returns the f | ile format (PNG, BMP). |
| Example | :SAVe:IMAGe:FILEFormat PNG | |
| | Sets the image file format to PNG. | |

:SAVe:IMAGe:INKSaver



| Description | Turns Ink Saver on or off. | |
|------------------|-------------------------------------|---------------------|
| Syntax | :SAVe:IMAGe:INKSaver {OFF ON ?} | |
| Related | :SAVe:IMAGe | |
| commands | :SAVe:IMAGe | :FILEFormat |
| Parameter | OFF | Turns Inksaver off. |
| | ON | Turns Inksaver on. |
| Return parameter | Returns Ink Saver status (ON, OFF). | |
| Example | :SAVe:IMAGe:INKSaver ON | |

Turns Ink Saver on.

:SAVe:SETUp



| Description | | Saves the current setup to internal memory (Set1~Set20) or the designated file path. | |
|-------------|-----------|--|--|
| Syntax | | Jp { <file path=""> ("Disk:/xxx.SET", SET) S1~S20}</file> | |
| Parameter | S1~S20 | Saves the setup to Set1~Set20 | |
| | File path | Saves the setup to disk to the specified file path. | |



Example :SAVe:SETUp S1
Saves the current setup to Set1 in internal memory.
:SAVe:SETUp "Disk:/DS0001.SET"
Saves the current setup to DS0001.SET in the root directory of the internal memory.

| :SAVe:WAVEf | orm | Set → | |
|---------------------|------------------------------------|---|--|
| Description | | Saves a waveform to internal memory or to a designated file path. | |
| Related commands | :SAVe:WAVE | :SAVe:WAVEform:FILEFormat | |
| Syntax | | :SAVe:WAVEform {CH1~REF4, REF <x> } {CH1~REF4, W1~W20} {CH1~ALL, file path}</x> | |
| Parameter | CH1~REF4, | CH1~CH4, Math, REF1~4 | |
| | <x></x> | 1,2,3,4 (REF1, REF2, REF3, REF4) | |
| | W1~W20 | Wave1~Wave20 | |
| | ALL | All the displayed waveforms on screen. | |
| | File path | Saves the waveform(s) to disk or USB to the specified file path. (LSF or CSV, but note that detail CSV can't be recalled to the scope.) | |
| Example 1 | :SAVe:WAVEform CH1, REF2 | | |
| | Saves the ch | annel1 waveform to REF2. | |
| Example 2 | :SAVe:WAVE | form:FILEFormat LSF | |
| • | :SAVe:WAVEform ALL, "Disk:/ALL001" | | |
| | "ALL001" is | format to LSF. A folder named screated and saves all displayed to the "ALL001" directory in the LSF | |



| Example 3 | :SAVe:WAVEform:FILEFormat FCSV :SAVe:WAVEform ALL, "Disk:/ALL002" |
|-----------|--|
| | Sets the file format to FCSV(fast CSV format). It then saves the all channel's waveforms to the root directory (Disk:/) of the internal flash disk in the CSV format (with the filename ALL002.CSV). |
| Example 4 | :SAVe:WAVEform:FILEFormat LSF :SAVe:WAVEform CH2, "Disk:/DS0003.LSF" |
| | Save the channel 2's waveform to the root directory (Disk:/) of the internal flash disk in the LSF format with DS0003.LSF as the filename. |
| | |



:SAVe:WAVEform:FILEFormat

| Description | Sets the v | Sets the waveform savefile format. | | |
|-------------|------------|--|--|--|
| Syntax | :SAVe:WA | :SAVe:WAVEform:FILEFormat {LSF DCSV FCSV ?} | | |
| Parameter | LSF | Sets the file format to the GDS 2000E's internal file format, LSF. (xxx.LSF) (no support LA) | | |
| | DCSV | Sets the file format to detail CSV. (xxx.CSV) | | |
| | FCSV | Sets the file format to fast CSV. (xxx.CSV) | | |

Return parameter Returns the file format (LSF, DCSV, FCSV).

Example :SAVe:WAVEform:FILEFormat LSF

Sets the file format to LSF.



Ethernet Commands

| | :ETHERr | net:DHCP | 163 | |
|-------------|---|------------------------------------|-------------|--|
| :ETHERnet:D | НСР | | Set → Query | |
| Description | Sets or o | Sets or queries the DHCP settings. | | |
| Syntax | :ETHERr | :ETHERnet:DHCP { OFF ON ? } | | |
| Parameter | ON Turns DHCP on. | | | |
| | OFF | Turns DHCP off | | |
| Example | xample :ETHERnet:DHCP ON Turns DHCP on. | | | |
| | | | | |



Time Commands

| | Time Communas | | | |
|-------------|---|--|--|--|
| | :DATe | 164 | | |
| :DATe | | Set → | | |
| Description | Sets the sy | Sets the system date and time. | | |
| Syntax | :DATe { <st< td=""><td colspan="3">:DATe {<string>}</string></td></st<> | :DATe { <string>}</string> | | |
| Parameter | <string></string> | "YYYYMMDDhhmmss" Where: YYYY: year MM: month DD: day hh: hour mm: minute ss: second | | |
| Example | Sets the tir Year: 2014 | 40802142830" me and date as: , Month: 08, Day: 02, Hour: 14 (2PM), 3, Second: 30. | | |

Bus Decode Commands

| :BUS1 | 166 |
|-----------------------------|-----|
| :BUS1:STATE | 166 |
| :BUS1:TYPe | 166 |
| :BUS1:INPut | |
| :BUS1:I2C:ADDRess:RWINClude | 167 |
| :BUS1:I2C:SCLK:SOURce | 168 |
| :BUS1:I2C:SDA:SOURce | 168 |
| :BUS1:UART:BITRate | 168 |
| :BUS1:UART:DATABits | 169 |
| :BUS1:UART:PARIty | 169 |
| :BUS1:UART:PACKEt | 170 |
| :BUS1:UART:POLARity | 170 |
| :BUS1:UART:EOFPAcket | 170 |
| :BUS1:UART:TX:SOURce | 171 |
| :BUS1:UART:RX:SOURce | 171 |
| :BUS1:SPI:SCLK:POLARity | 172 |
| :BUS1:SPI:SS:POLARity | 172 |
| :BUS1:SPI:WORDSize | 172 |
| :BUS1:SPI:BITORder | 173 |
| :BUS1:SPI:SCLK:SOURce | 173 |
| :BUS1:SPI:SS:SOURce | 173 |
| :BUS1:SPI:MOSI:SOURce | 174 |
| :BUS1:SPI:MISO:SOURce | 174 |
| :BUS1:DISPlay:FORMAt | 174 |
| :LISTer:DATA | 175 |
| :BUS1:CAN:SOURce | 175 |
| :BUS1:CAN:PROBe | 175 |
| :BUS1:CAN:SAMPLEpoint | 176 |
| :BUS1:CAN:BITRate | 176 |
| :BUS1:LIN:BITRate | |
| :BUS1:LIN:IDFORmat | 177 |
| :BUS1:LIN:POLARity | 177 |



| :BUS1:LIN:SAMPLEpoint | 178 |
|-----------------------|-----|
| :BUS1:LIN:SOURce | 178 |
| :BUS1:LIN:STANDard | 178 |

:BUS1

Description Returns the supported BUS types.

Syntax :BUS1?

Return Parameter Returns the supported bus types.

Example BUS1?

I2C,SPI,UART, CAN, LIN

:BUS1:STATE



Query)

Description Sets or queries the state of the bus.

Syntax :BUS1:STATE { OFF | ON | ? }

Related :BUS1:TYPe

commands

Parameter/Return OFF Turns the bus off.

ON Turns the bus on.

Example :BUS1:STATE ON

Turns the bus on.

:BUS1:TYPe



Description Sets or queries the type of bus.

 $Syntax \qquad \qquad :BUS1:TYPe \; \{\; UART \; | \; I2C \; | \; SPI \; | \; CAN \; | \; LIN \; | \; ? \; \}$

Related :BUS1:STATE

commands

Parameter/Return UART Sets the bus to UART mode.

parameter 12C Sets the bus to 1²C mode.



| | SPI | Sets the bus to SPI mode. | |
|----------------------------|---|---------------------------------------|--|
| | CAN | Sets the bus to CAN mode. | |
| | LIN | Sets the bus to LIN mode. | |
| Example | :BUS1:TYPe S | ;PI | |
| · | Sets the bus | to SPI mode. | |
| | | Set → | |
| :BUS1:INPut | | Query | |
| Description | Sets or return | ns the analog source. | |
| Syntax | :BUS1:INPut | {ANAlog ?} | |
| Parameter/Return parameter | ANAlog | Sets the source to the analog inputs. | |
| Example1 | :BUS1:INPut? | | |
| | >ANALOG | | |
| | | <u>Set</u> → | |
| :BUS1:I2C:ADD | Ress:RWIN | Clude → Query | |
| Description | Sets or queries whether the read/write bit is included in the I ² C address. | | |
| Syntax | :BUS1:I2C:ADDRess:RWINClude { OFF ON ? } | | |
| Related commands | :BUS1:STATE | | |
| Parameter | OFF | The R/W bit is not included. | |
| | ON | The R/W bit is included. | |
| Return parameter | 0 | The R/W bit is not included. | |
| | 1 | The R/W bit is included. | |
| Example | :BUS1:I2C:AD | DDRess:RWINClude ON | |

Includes the R/W bit in the I²C address.



| :BUS1:I2C:SCLI | K:SOURce | | | Set → Qu | → ery) |
|----------------------------|---|-------------|--------------------|-------------|---------------|
| Description | Sets or queries which channel is used for the I ² C SCLK source. | | | | |
| Syntax | :BUS1:I2C:SC ? } | LK:SOU | JRce { CH1 | CH2 CH | 13 CH4 |
| Parameter/Return parameter | CH1 to CH4 | Analog | channels 1 ~ | 4. | |
| Example | :BUS1:I2C:SC | LK:SOU | JRce CH1 | | |
| | Sets channel | 1 as the | e SCLK sour | ce. | |
| | | | | Set - | — |
| :BUS1:I2C:SDA | :SOURce | | | → Qu | ery |
| Description | Sets or queries which channel is used for the I ² C SDA source. | | | | |
| Syntax | :BUS1:I2C:SD | A:SOUI | Rce { CH1 C | :H2 CH | 3 CH4 ? } |
| Parameter/Return parameter | CH1 to CH4 Analog channels 1 ~ 4. | | | | |
| Example | :BUS1:I2C:SD | A:SOUI | Rce CH1 | | |
| | Sets channel | 1 as the | e SDA sourc | e. | |
| | | | | Set - | → |
| :BUS1:UART:BI | TRate | | | → Qu | ery |
| Description | Sets or queri | es the L | JART bit rat | e. | |
| Syntax | :BUS1:UART: | BITRate | { <nr1> ? }</nr1> | | |
| Parameter/Return | <nr1></nr1> | UART b | oit rate (0~31 |) | |
| parameter | | <nr1></nr1> | Rate (bps) | <nr1></nr1> | Rate (bps) |
| | | 0 | 50 | 16 | 15200 |
| | | 1 | 75 | 17 | 19200 |
| | | 2 | 110 | 18 | 28800 |
| | | 3 | 134 | 19 | 31250 |

| 4 | 150 | 20 | 38400 |
|----|-------|----|---------|
| 5 | 300 | 21 | 56000 |
| 6 | 600 | 22 | 57600 |
| 7 | 1200 | 23 | 76800 |
| 8 | 1800 | 24 | 115200 |
| 9 | 2000 | 25 | 128000 |
| 10 | 2400 | 26 | 230400 |
| 11 | 3600 | 27 | 460800 |
| 12 | 4800 | 28 | 921600 |
| 13 | 7200 | 29 | 1382400 |
| 14 | 9600 | 30 | 1843200 |
| 15 | 14400 | 31 | 2764800 |

Example

:BUS1:UART:BITRate 10

Sets the bit rate to 2400.

:BUS1:UART:DATABits



| Description | Sets or queries the number UART data for bus 1. | | |
|------------------|---|--------------------------------|--|
| Syntax | :BUS1:UART:DATABits { 5 6 7 8 9 ? } | | |
| Parameter/Return | 5 | 5 data bits in the UART frame. | |
| parameter | 6 | 6 data bits in the UART frame. | |
| | 7 | 7 data bits in the UART frame. | |
| | 8 | 8 data bits in the UART frame. | |
| Example | :BUS1:UART:DATABits 7 | | |

:BUS1:UART:DATABits 7

Sets the UART frame to 7 bits.

:BUS1:UART:PARIty



| Description | Sets or queries the UART bus parity. |
|-------------|---------------------------------------|
| Syntax | :BUS1:UART:PARIty { <nr1> ? }</nr1> |



0: None Parameter/Return < NR1> parameter 1: Odd parity 2: Even parity Example :BUS1:UART:PARIty 1

Sets the parity to odd.

:BUS1:UART:PACKEt



| Descr | iption | Sets or queries the UART packet setting. | | |
|-------|--------------|--|--------|--|
| Synta | x | :BUS1:UART:PACKEt { <nr1> ? }</nr1> | | |
| Paran | neter/Return | <nr1></nr1> | 0: Off | |
| paran | neter | | 1: On | |
| Exam | ple | :BUS1:UART:PACKEt 1 | | |

Turns UART packets on.

:BUS1:UART:POLARity



| Description | Sets or retu | Sets or returns the UART polarity. | | |
|-------------|-----------------------|--|--|--|
| Syntax | :BUS1:UAR | :BUS1:UART: POLARity {NORMal INVerted} | | |
| | :BUS1:UART: POLARity? | | | |
| Parameter | NORMal | Sets normal UART polarity. | | |
| | INVerted | Sets inverted UART polarity. | | |
| Example | :BUS1:UAR | :BUS1:UART:POLARity NORMal | | |
| | :BUS1:UART:POLARity? | | | |
| | NORMAL | | | |

:BUS1:UART:EOFPAcket



| Description | Sets or queries the EOF character for the UART packet setting. |
|-------------|--|
| Syntax | :BUS1:UART:EOFPAcket <nr1></nr1> |



| Parameter/Return parameter | <nr1></nr1> | 0: NULL | |
|----------------------------|---|--|--|
| | | 1: LF (line feed) | |
| | | 2: CR (carriage return) | |
| | | 3: SP (space character) | |
| | | 4: FF | |
| Example | :BUS1:UART:I | EOFPAcket 2 | |
| | Sets the OEF | character to CR. | |
| | | Set → | |
| :BUS1:UART:TX | K:SOURce | → Query | |
| Description | Sets or querion | es which channel is used for the urce. | |
| Syntax | :BUS1:UART:TX:SOURce { OFF CH1 CH2 CH3 CH4 ? } | | |
| Parameter/Return | OFF | Off, no Tx source | |
| parameter | CH1 to CH4 | Analog channels CH1 to CH4 | |
| Example | :BUS1:UART: | TX:SOURce CH1 | |
| | Sets channel 1 as the Tx source. | | |
| | | (Set)→ | |
| :BUS1:UART:R | X:SOURce | Query | |
| Description | Sets or queries which channel is used for the UART Rx source. | | |
| Syntax | :BUS1:UART:RX:SOURce { OFF CH1 CH2 CH3 CH4 ? } | | |
| Parameter/Return | OFF | Off, no Rx source | |
| parameter | CH1 to CH4 | Analog channels CH1 to CH4 | |
| Example | :BUS1:UART: | RX:SOURce CH1 | |
| · | Sets channel 1 as the Rx source. | | |



| :BUS1:SPI:SCLI | <:POLARity | Set → Query |
|---|---|--|
| Description | Sets or querious SPI bus. | es the polarity of the SCLK line for the |
| Syntax | :BUS1:SPI:SC | LK:POLARity { FALL RISE ? } |
| Parameter/Return | FALL | Sets the polarity to falling edge. |
| parameter | RISE | Sets the polarity to rising edge. |
| Example | :BUS1:SPI:SC | LK:POLARity FALL |
| | Sets the polar | rity to falling edge. |
| | | Set |
| :BUS1:SPI:SS:P | OLARity | → Query |
| Description | Sets or querion bus. | es the polarity of the SS line for the SPI |
| Syntax | :BUS1:SPI:SS:POLARity { LOW HIGH ? } | |
| Parameter/Return | | , , , , , , |
| Parameter/Return | LOW | Active low polarity |
| Parameter/Return parameter | | |
| • | HIGH | Active low polarity |
| parameter | HIGH :BUS1:SPI:SS: | Active low polarity Active high polarity |
| parameter | HIGH :BUS1:SPI:SS: | Active low polarity Active high polarity POLARity LOW |
| parameter | HIGH :BUS1:SPI:SS: Sets the SS lin | Active low polarity Active high polarity POLARity LOW ne to active low. |
| Example | HIGH :BUS1:SPI:SS: Sets the SS lin | Active low polarity Active high polarity POLARity LOW ne to active low. Set |
| Example::BUS1:SPI:WOI | HIGH :BUS1:SPI:SS: Sets the SS lin RDSize Sets the num | Active low polarity Active high polarity POLARity LOW ne to active low. Set Query |
| parameter Example :BUS1:SPI:WOI | HIGH :BUS1:SPI:SS: Sets the SS lin RDSize Sets the num :BUS1:SPI:WC | Active low polarity Active high polarity POLARity LOW ne to active low. Set Query ber of bits per word for the SPI bus. |
| Example :BUS1:SPI:WOI Description Syntax Parameter/Return | HIGH :BUS1:SPI:SS: Sets the SS lin RDSize Sets the num :BUS1:SPI:WC | Active low polarity Active high polarity POLARity LOW ne to active low. Set Query ber of bits per word for the SPI bus. DRDSize { <nr1> ? } Bits per word (4~32)</nr1> |



| | | | Set → |
|----------------------------|--|----------------------------|---------------------|
| :BUS1:SPI:BITC | DRder | | Query |
| Description | Sets or queri | es the bit order for th | ne SPI bus. |
| Syntax | :BUS1:SPI:BI | TORder { <nr1> ? }</nr1> | |
| Parameter/Return | <nr1></nr1> | 0: MSB bit first | |
| parameter | | 1: LSB bit first | |
| Example | :BUS1:SPI:BI ⁻ 0 | TORder? | |
| | The bit order | r is currently set as N | ISB bit first. |
| | | | Set → |
| :BUS1:SPI:SCLI | <:SOURce | | → Query |
| Description | Sets or queries which channel is used for the SPI SCLK source. | | |
| Syntax | :BUS1:SPI:SC ? } | LK:SOURce { CH1 C | CH2 CH3 CH4 |
| Parameter/Return parameter | CH1 to CH4 | Analog channels CH1 | to CH4 |
| Example | :BUS1:SPI:SC | LK:SOURce CH1 | |
| | Sets channel | 1 as the SPI SCLK so | ource. |
| | | | Set → |
| :BUS1:SPI:SS:S | OURce | | → Query |
| Description | Sets or querionsource. | es which channel is ι | used for the SPI SS |
| Syntax | :BUS1:SPI:SS | :SOURce { CH1 CH | 2 CH3 CH4 ? } |
| Parameter/Return parameter | CH1 to CH4 | Analog channels CH1 | to CH4 |
| Example | :BUS1:SPI:SS | :SOURce CH1 | |
| | Sets channel | 1 as the SPI SS source | ce. |



| :BUS1:SPI:MO | SI:SOURce | | Set → Query |
|------------------------------|---|---------------------------------------|-----------------|
| Description | Sets or queries which channel is used for the SPI MOSI source. | | |
| Syntax | :BUS1:SPI:M(CH4 ? } | OSI:SOURce { OFF | CH1 CH2 CH3 |
| Parameter/Return parameter | CH1 to CH4 OFF | Analog channels CH No MOSI source. | 1 to CH4 |
| Example | | OSI:SOURce CH1 1 as the SPI MOSI s | |
| :BUS1:SPI:MIS | O:SOURce | | Set → Query |
| Description | Sets or queries which channel is used for the SPI MISO source. | | |
| Syntax | :BUS1:SPI:MISO:SOURce { OFF CH1 CH2 CH3 CH4 ? } | | |
| Parameter/Return parameter | CH1 to CH4 OFF | Analog channels CH No MISO source. | 1 to CH4 |
| Example | :BUS1:SPI:MISO:SOURce CH1 Sets channel CH1 as the SPI MISO source. | | |
| :BUS1:DISPlay:FORMAt → Query | | | |
| Description | Sets or queries the display format for the bus, either binary or hexadecimal. | | |
| Syntax | :BUS1:DISPlay:FORMAt { BINary HEXadecimal ASCII ? } | | |
| Parameter/Return parameter | · | Binary format Hexadecimal forma | t |



Example :BUS1:DISPlay:FORMAt BINary

Sets the display format to binary.

:LISTer:DATA



| Description | Returns the Event Table data as a binary block |
|-------------|--|
| | data. |
| | |

Syntax :LISTer:DATA?

Return Parameter Returns the event table as binary block data. The

binary block data contains comma separated data with

new lines at the end of each row.

:BUS1:CAN:SOURce



| Description | Sets or returns the CAN input source. | |
|------------------|--|-----------------------|
| Syntax | :BUS1:CAN:SOURce { CH1 CH2 CH3 CH4 ? } | |
| Parameter/Return | CH1 ~ CH4 | Analog channel source |

parameter

Example

:BUS1:CAN:SOURCE?

>CH1

Returns the CAN source.

:BUS1:CAN:PROBe



| Description | Sets or returns the signal type of the CAN bus. | |
|------------------|---|----------|
| Syntax | :BUS1:CAN:PROBe {CANH CANL TX RX ? } | |
| Parameter/Return | CANH | CAN-High |
| parameter | CANL | CAN-Low |
| | TX | Transmit |
| | RX | Receive |



Example :BUS1:CAN:PROBe?

>CANH

:BUS1:CAN:PROBe CANL

:BUS1:CAN:PROBe?

>CANL

:BUS1:CAN:SAMPLEpoint



| Description | Returns the sample point of the CAN bus. |
|-------------|--|
| Description | Returns the sample point of the Crity bus. |

Syntax :BUS1:CAN:SAMPLEpoint?

Return Parameter Returns the sample point of the CAN bus as a

percentage of the bit time.

Example :BUS1:CAN:SAMPLEpoint?

50

Returns the sample point as a percentage.

:BUS1:CAN:BITRate



| 5 | | 4 14 4 64 64 74 | |
|----------------------------|---|-----------------|--|
| Description | Sets or returns the bit rate of the CAN bus. | | |
| Syntax | :BUS1:CAN:BITRate {RATE10K RATE20K RATE50K RATE125K RATE250K RATE500K RATE800K RATE1M ?} | | |
| Parameter/Return parameter | RATE10K | 10 kbps | |
| | RATE20K | 20 kbps | |
| | RATE50K | 50 kbps | |

RATE125K 125 kbps RATE250K 250 kbps RATE500K 500 kbps RATE800K 800 kbps RATE1M 1 Mbps



Example :BUS1:CAN:BITRate?

>RATE250K

:BUS1:CAN:BITRate rate10k

:BUS1:CAN:BITRate?

>RATE10K

:BUS1:LIN:BITRate

Set → Query

Description Sets or returns the bit rate of the LIN bus.

Syntax :BUS1:LIN:BITRate {<NR1> | ?}

Parameter/Return <NR1> 1200, 2400, 4800, 9600, 10417, 19200

parameter Example

:BUS1:LIN:BITRate 9600

Sets the LIN bit rate to 9600 kbps.

:BUS1:LIN:IDFORmat



Description Sets or returns the LIN ID format.

Syntax :BUS1:LIN:IDFORmat {NOPARity|PARIty|?}

Parameter/Return NOPARity Don't include parity bits with Id.

parameter

PARIty Include parity bits with Id.

Example :BUS1:LIN:IDFORmat?

NOPARITY

Returns the ID format.

:BUS1:LIN:POLARity



Description Sets or returns the LIN polarity.

Syntax :BUS1:LIN:POLARity {NORMal|INVerted|?}

Parameter/Return NORMal Normal LIN polarity parameter INVerted Inverted LIN polarity



Example :BUS1:LIN:POLARity?

NORMAL

Returns the LIN polarity.

:BUS1:LIN:SAMPLEpoint



| Description | Returns the sample point. | |
|------------------|--|--|
| Syntax | :BUS1:LIN:SAMPLEpoint? | |
| Return Parameter | Returns the sample point of the LIN bus as a percentage. | |
| Example | :BUS1:LIN:SAMPLEpoint? | |

50

Returns the sample point as a percentage.

:BUS1:LIN:SOURce



| Description | Sets or returns the LIN data source. | |
|----------------------------|---|-----------------------|
| Syntax | :BUS1:LIN:SOURce {CH1 CH2 CH3 CH4 ? } | |
| Parameter/Return parameter | CH1 ~ CH4 | Analog channel source |
| Example | :BUS1:LIN:SOURCE? | |

>CH1

>CH1

Returns the LIN source.

:BUS1:LIN:STANDard



| Description | Sets or returns the LIN standard. | | |
|-------------------------------|-------------------------------------|--------------------------|--|
| Syntax | :BUS1:LIN:STANDard {V1X V2X BOTH ?} | | |
| Parameter/Return parameter | V1X | Lin standard version 1.x | |
| | V2X | Lin standard version 2.x | |
| | вотн | Both standards | |



Example :BUS1:LIN:STANDard?

>BOTH

Returns the LIN standard.



Mark Commands

| :MARK | 180 |
|--------------|-----|
| :MARK:CREATE | 180 |
| :MARK:DELEte | 181 |

:MARK



| Description | Move to ne | Move to next or previous event mark. | | |
|-------------|-------------|--------------------------------------|--|--|
| Syntax | :MARK { NE | :MARK { NEXT PREVious } | | |
| Related | :MARK:CREA | :MARK:CREATE | | |
| commands | :MARK:DELE | :MARK:DELEte | | |
| Parameter | NEXT | Move to next mark | | |
| | PREVious | Move to previous mark | | |
| Example | :MARK NEX | :MARK NEXT | | |
| | Moves to th | Moves to the next event mark. | | |

:MARK:CREATE



| Description | position or | Creates a mark on the waveform at the current position or creates a mark for all the events for the current waveform. | | |
|------------------|-------------|---|--|--|
| Syntax | :MARK:CRE | :MARK:CREATE { CURRent ALL } | | |
| Related commands | :MARK | :MARK | | |
| | :MARK:DEL | :MARK:DELEte | | |
| Parameter | CURRent | Creates a mark at the current position | | |
| | ALL | Creates a mark for all the events. | | |
| Example | :MARK:CRE | :MARK:CREATE CURRent | | |
| | Creates a m | Creates a mark at the current position. | | |



| :MARK:DELEt | e | Set → | | | |
|-------------|--|--------------------------------|--|--|--|
| Description | Deletes the current mark or all the marks on a waveform. | | | | |
| Syntax | :MARK:DEL | :MARK:DELEte { CURRent ALL } | | | |
| Related | :MARK | | | | |
| commands | :MARK:CRE | :MARK:CREATE | | | |
| Parameter | CURRent | Deletes the current mark | | | |
| | ALL | Deletes all the marks | | | |
| Example | :MARK:DELEte CURRent | | | | |
| | Deletes the | current mark. | | | |



Search Commands

| :SEARCH:COPY | .183 |
|---|------|
| :SEARCH:STATE | .184 |
| :SEARCH:TOTAL | .184 |
| :SEARCH:TRIGger:TYPe | .184 |
| :SEARCH:TRIGger:SOURce | .185 |
| :SEARCH:TRIGger:EDGE:SLOP | .185 |
| :SEARCH:TRIGger:LEVel | .186 |
| :SEARCH:TRIGger:HLEVel | .186 |
| :SEARCH:TRIGger:LLEVel | .187 |
| :SEARCH:TRIGger:PULSEWidth:POLarity | .187 |
| :SEARCH:TRIGger:RUNT:POLarity | .188 |
| :SEARCH:TRIGger:RISEFall:SLOP | .188 |
| :SEARCH:TRIGger:PULSe:WHEn | .189 |
| :SEARCH:TRIGger:PULSe:TIMe | .189 |
| :SEARCH:TRIGger:RUNT:WHEn | .190 |
| :SEARCH:TRIGger:RUNT:TIMe | .190 |
| :SEARCH:TRIGger:RISEFall:WHEn | .191 |
| :SEARCH:TRIGger:RISEFall:TIMe | .191 |
| :SEARCH:TRIGger:BUS:TYPe | .192 |
| :SEARCH:TRIGger:BUS:B1:I2C:CONDition | .192 |
| $: SEARCH: TRIGger: BUS: B1: I2C: ADDRess: MODe \dots \\$ | .193 |
| : SEARCH: TRIGger: BUS: B1: I2C: ADDRess: TYPe | .193 |
| : SEARCH: TRIGger: BUS: B1: I2C: ADDRess: VALue | .194 |
| :SEARCH:TRIGger:BUS:B1:I2C:ADDRess | |
| :DIRection | |
| :SEARCH:TRIGger:BUS:B1:I2C:DATa:SIZe | |
| :SEARCH:TRIGger:BUS:B1:I2C:DATa:VALue | |
| :SEARCH:TRIGger:BUS:B1:UART:CONDition | |
| $: SEARCH: TRIGger: BUS: B1: UART: RX: DATa: SIZe \dots \\$ | |
| : SEARCH: TRIGger: BUS: B1: UART: RX: DATa: VALue . | |
| $: SEARCH: TRIGger: BUS: B1: UART: TX: DATa: SIZe \dots \\$ | |
| $: SEARCH: TRIGger: BUS: B1: UART: TX: DATa: VALue \ . \\$ | |
| ·SFARCH·TRIGger·BLIS·R1·SPI·CONDition | 200 |



| :SEARCH:TRIGger:BUS:B1:SPI:DATa:SIZe | 200 |
|--|-----|
| :SEARCH:TRIGger:BUS:B1:SPI:DATa:MISO | |
| :VALue | 201 |
| :SEARCH:TRIGger:BUS:B1:SPI:DATa:MOSI | |
| :VALue | 202 |
| :SEARCH:TRIGger:BUS:B1:CAN:CONDition | 203 |
| :SEARCH:TRIGger:BUS:B1:CAN:FRAMEtype | 204 |
| :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODe | 204 |
| : SEARCH: TRIGger: BUS: B1: CAN: IDentifier: VALue . | 205 |
| :SEARCH:TRIGger:BUS:B1:CAN:IDentifier | |
| :DIRection | 205 |
| : SEARCH: TRIGger: BUS: B1: CAN: DATa: QUALifier | 206 |
| :SEARCH:TRIGger:BUS:B1:CAN:DATa:SIZe | 207 |
| :SEARCH:TRIGger:BUS:B1:CAN:DATa:VALue | 207 |
| :SEARCH:TRIGger:BUS:B1:LIN:CONDition | 208 |
| :SEARCH:TRIGger:BUS:B1:LIN:DATa:QUALifier | 209 |
| :SEARCH:TRIGger:BUS:B1:LIN:DATa:SIZe | 210 |
| :SEARCH:TRIGger:BUS:B1:LIN:DATa:VALue | 210 |
| :SEARCH:TRIGger:BUS:B1:LIN:ERRTYPE | 211 |
| :SEARCH:TRIGger:BUS:B1:LIN:IDentifier:VALue | 212 |
| :SEARCH:FFTPeak:METHod | 212 |
| :SEARCH:FFTPeak:METHod:MPEak | |
| :SEARCH:FFTPeak:SINFo | |
| | |

:SEARCH:COPY



| Description | _ | Copies the search settings to the trigger settings or copies the trigger settings to the search settings. | | | |
|-------------|--------------------|---|--|--|--|
| Syntax | :SEARCH:COPY {SEAF | :SEARCH:COPY {SEARCHtotrigger TRIGgertosearch} | | | |
| Parameter | SEARCHtotrigger | Copy the search setting to the trigger settings. | | | |
| | TRIGgertosearch | Copy the trigger settings | | | |



Example :SEARCH:COPY SEARCHtotrigger

Copies the search settings to the trigger settings.

:SEARCH:STATE



Description Sets or queries whether the Search function is on or

off.

Syntax :SEARCH:STATE { OFF | ON | ? }

Parameter/Return OFF Turn the Search function on.

ON Turn the Search function off.

Example :SEARCH:STATE ON

Turn Search on.

:SEARCH:TOTAL



Description Returns the total number of events found from the

search function.

Syntax :SEARCH:TOTAL?

Return parameter <NR1> Number of events.

Example :SEARCH:TOTAL?

5

:SEARCH:TRIGger:TYPe



Description Sets or queries the search trigger type.

Syntax :SEARCH:TRIGger:TYPe { EDGe | PULSEWidth |

RUNT | RISEFall | FFTPeak | BUS | ? }

Parameter/Return EDGe Edge trigger

parameter PULSEWidth Pulse width trigger

RUNT Runt trigger

RISEFall Rise and Fall trigger



| | | COMMINATE DE IMES | | | |
|----------------------------|--|------------------------------------|--|--|--|
| | FFTPeak | FFT Peak trigger | | | |
| | BUS | Bus trigger | | | |
| Example | :SEARCH:TRI | Gger:TYPe EDGE | | | |
| | Sets the search | ch trigger to the edge type. | | | |
| | | Set → | | | |
| :SEARCH:TRIG | ger:SOURce | → Query | | | |
| Description | Sets or queri | es the search trigger source. | | | |
| Syntax | :SEARCH:TRI | Gger:SOURce {CH1 CH2 CH3 CH4 | | | |
| Parameter/Return parameter | CH1 to CH4 | Channel 1 to Channel 4 | | | |
| Example | :SEARCH:TRIGger:SOURce CH1 | | | | |
| | Sets the search trigger source as CH1. | | | | |
| | | <u>Set</u> → | | | |
| :SEARCH:TRIG | ger:EDGE:S | LOP → Query | | | |
| Description | Sets or queri | es the search trigger slope. | | | |
| Syntax | :SEARCH:TRIGger:EDGe:SLOP { RISe FALL EITher ? } | | | | |
| Related commands | :SEARCH:TRIGger:TYPe | | | | |
| Parameter | RISe | Rising slope | | | |
| | FALL | Falling slope | | | |
| | EITher | Either rising or falling slope | | | |
| Return parameter | Returns the t | rigger slope. | | | |
| Example | :SEARCH:TRIGger:EDGe:SLOP FALL | | | | |
| | Sets the search trigger slope to falling. | | | | |
| | | | | | |



| :SEARCH:TRIG | ger:LEVel | | Set → Query | |
|------------------|--|---|---------------------|--|
| Description | Sets or queries the search trigger level. | | | |
| Syntax | :SEARCH:TRIGger:LEVel {TTL ECL SETTO50 <nrf> ?}</nrf> | | | |
| Related commands | :SEARCH:TRIGger:TYPe | | | |
| Parameter | <nrf></nrf> | Trigger level value | | |
| | TTL | Sets the search trigg | er level to TTL. | |
| | ECL | Sets the search trigg | er level to ECL. | |
| | SETTO50 | Sets the search trigg User level (50% by o | | |
| Return parameter | <nr3></nr3> | Returns the trigger. | | |
| Example1 | :SEARCH:TRI | Gger:LEVel TTL | | |
| | Sets the sear | ch trigger level to TT | TL. | |
| Example2 | :SEARCH:TRIGger:LEVel 3.30E-1 | | | |
| | Sets the search trigger level to 330mV/mA. | | | |
| :SEARCH:TRIG | ger:HLEVel | | Set → Query | |
| Description | Sets or queri | es the high level sea | rch trigger. | |
| Note | Applicable for Rise and Fall/Pulse Runt search triggers. | | | |
| Syntax | :SEARCH:TRIGger:HLEVel { <nrf> ?}</nrf> | | | |
| Related commands | :SEARCH:TRIGger:TYPe | | | |
| Parameter | <nrf></nrf> | High level value. | | |
| Return parameter | <nr3></nr3> | Returns the high lev | vel search trigger. | |
| Example | :SEARCH:TRI | Gger:HLEVel 3.30E-1 | | |
| | Sets the high level search trigger to 330mV/mA. | | | |



| :SEARCH:TRIG | ger:LLEVel | Set → Query | | |
|---------------------|---|--|--|--|
| Description | Sets or queries the low level search trigger. | | | |
| Note | Applicable fo | or Rise and Fall/Pulse Runt triggers. | | |
| Syntax | :SEARCH:TRIGger:LLEVel { <nrf> ?}</nrf> | | | |
| Related commands | :SEARCH:TRI | Gger:TYPe | | |
| Parameter | <nrf></nrf> | Low level value. | | |
| Return parameter | <nr3></nr3> | Returns the low level. | | |
| :SEARCH:TRIG | | level search trigger to 330mV/mA. Set Query Query | | |
| Description | Sets or queri polarity. | es the pulse width search trigger | | |
| Syntax | :SEARCH:TRIGger:PULSEWidth:POLarity {POSitive NEGative ?} | | | |
| Related commands | :SEARCH:TRIGger:TYPe | | | |
| Parameter | POSitive | Positive polarity | | |
| | NEGative | Negative polarity | | |
| Return parameter | Returns the pulse width polarity. | | | |
| Example | :SEARCH:TRIGger:PULSEWidth:POLarity POSitive | | | |
| | Sets the pulse width polarity to positive. | | | |



| :SEARCH:TRIG | ger:RUNT:P | OLarity Set → Query | | |
|------------------|--|--|--|--|
| Description | Sets or queries the Pulse Runt search trigger polarity. | | | |
| Syntax | :SEARCH:TRI NEGative E | Gger:RUNT:POLarity {POSitive ITher ?} | | |
| Related commands | :SEARCH:TRI | Gger:TYPe | | |
| Parameter | POSitive | Positive polarity | | |
| | NEGative | Negative polarity | | |
| | EITher | Positive or negative polarity | | |
| Return parameter | Returns the j | pulse runt search trigger polarity. | | |
| Example | :SEARCH:TRI | Gger:RUNT:POLarity POSitive | | |
| | Sets the Pulse Runt search trigger polarity to positive. | | | |
| | | <u>Set</u> → | | |
| :SEARCH:TRIG | ger:RISEFal | l:SLOP → Query | | |
| Description | Sets or queries the slope of the Rise and Fall search trigger. | | | |
| Syntax | :SEARCH:TRIGger:RISEFall:SLOP { RISe FALL EITher ? } | | | |
| Related commands | :SEARCH:TRIGger:TYPe | | | |
| Parameter | RISe | Rising slope | | |
| | FALL | Falling slope | | |
| | EITher | Either rising or falling slope | | |
| Return parameter | Returns the | rise & fall slope. | | |
| Example | :SEARCH:TRIGger:RISEFall :SLOP RISe | | | |
| | :2EARCH:1KI | Gger:RISEFall :SLOP RISe | | |



| :SEARCH:TRIG | ger:PULSe:\ | WHEn | Set → Query | |
|------------------|---|--|---------------------|--|
| Description | Sets or queries the pulse width search trigger conditions. | | | |
| Syntax | | Gger:PULSe:WHEn { Qual UNEQual ?} | • | |
| Related | :SEARCH:TRI | Gger:TYPe | | |
| commands | :SEARCH:TRI | Gger:PULSe:TIMe | | |
| Parameter | MOREthan | > | | |
| | LESSthan | < | | |
| | EQual | = | | |
| | UNEQual | ≠ | | |
| Return parameter | Returns the | pulse width search t | rigger conditions. | |
| Example | :SEARCH:TRIGger:PULSe:WHEn UNEQual Sets the pulse width search trigger conditions to not equal to (\neq). | | | |
| | | | | |
| | | | Set → | |
| :SEARCH:TRIG | ger:PULSe: | ГІМе | Query | |
| Description | Sets or queri | es the pulse width s | earch trigger time. | |
| Syntax | :SEARCH:TRIGger:PULSe:TIMe { <nrf> ?}</nrf> | | | |
| Related | :SEARCH:TRIGger:TYPe | | | |
| commands | :SEARCH:TRIGger:PULSe:WHEn | | | |
| Parameter | <nrf> Pulse width time (4ns~10s)</nrf> | | | |
| Return parameter | <nr3></nr3> | Returns the pulse w seconds. | vidth time in | |
| Example | :SEARCH:TRIGger:PULSe:TIMe 4.00E-5 | | | |
| | Sets the pulse width search trigger to 40.0us. | | | |



Set)→ :SEARCH:TRIGger:RUNT:WHEn (Query Sets or queries the pulse runt search trigger Description conditions. :SEARCH:TRIGger:RUNT:WHEn {MOREthan | Syntax LESSthan | EQual | UNEQual | ? } Related :SEARCH:TRIGger:TYPe :SEARCH:TRIGger:RUNT:TIMe commands MOREthan Parameter > LESSthan < Equal **UNEQual** ¥ Return parameter Returns the pulse runt search trigger conditions. :SEARCH:TRIGger:RUNT:WHEn UNEQual Example Sets the pulse runt search trigger condition to unequal (≠). Set) :SEARCH:TRIGger:RUNT:TIMe (Query Sets or queries the pulse runt search trigger time. Description Syntax :SEARCH:TRIGger:RUNT:TIMe {<NRf> | ? } Related :SEARCH:TRIGger:TYPe commands :SEARCH:TRIGger:RUNT:WHEn <NRf>Parameter Pulse runt time (4nS to 10S) Returns the runt time in seconds. Return Parameter < NR3> Example :SEARCH:TRIGger:RUNT:TIMe 4.00E-5 Sets the pulse runt time to 40.0uS.



| :SEARCH:TRIG | ger:RISEFall | l:WHEn | Set → Query | |
|------------------|--|---|-------------------|--|
| Description | Sets or queries the rise and fall search trigger conditions. | | | |
| Syntax | | Gger:RISEFall:WHEn Qual UNEQual ? } | {MOREthan | |
| Related commands | :SEARCH:TRI :SEARCH:TRI | Gger:TYPe Gger:RISEFall:TIMe | | |
| Parameter | MOREthan LESSthan Equal UNEQual | > < = = ≠ | | |
| Return parameter | Returns the 1 | rise and fall search t | rigger condition. | |
| Example | :SEARCH:TRIGger:RISEFall:WHEn UNEQual Sets the rise and fall search trigger condition to unequal (#). Set | | | |
| :SEARCH:TRIG | ger:RISEFall | : i livie | Query | |
| Description | Sets or queri | es the rise and fall t | ime. | |
| Syntax | :SEARCH:TRIGger:RISEFall:TIMe { <nrf> ? }</nrf> | | | |
| Related | :SEARCH:TRIGger:TYPe | | | |
| commands | :SEARCH:TRIGger:RISEFall:WHEn | | | |
| Parameter | <nrf> Rise and Fall time (4nS to 10S)</nrf> | | | |
| Return Parameter | <nr3></nr3> | Returns the rise and seconds. | d fall time in | |
| Example | :SEARCH:TRIGger:RISEFall:TIMe 4.00E-5 Sets the trigger rise and fall time to 40.0us. | | | |



| :SEARCH:TRIGger:BUS:TYPe | | | | → Query |
|--------------------------|---|---|---|--|
| Description | Returns the current bus type. | | | |
| Syntax | :SEARCH:TRIGger:BUS:TYPe? | | | |
| Return parameter | 12C mode | | | |
| | SPI | SPI mod | le | |
| | UART | UART n | node | |
| | CAN | CAN mo | ode | |
| | LIN | LIN mod | de | |
| Example | :SEARCH:TRI | IGger:BUS | S:TYPe? | |
| | UART | | | |
| | | | | Set → |
| :SEARCH:TRIG | ger:BUS:B1 | :I2C:CO | NDition | → Query |
| Description | Sets or queri | ies the I²C | C search tri | gger conditions. |
| Syntax | :SEARCH:TRIGger:BUS:B1:I2C:CONDition {STARt STOP REPEATstart ACKMISS ADDRess DATA ADDRANDDATA ? } | | | |
| Parameter | STARt | Set Start as the I ² C search trigger condition. | | |
| | STOP | | Stop as the dition. | e I ² C search trigger |
| | REPEATstart | | - | Start as the I ² C condition. |
| | ACKMISS | as t | Set Missing Acknowledgement as the I ² C search trigger condition. | |
| | ADDRess Set Address as the I trigger condition. | | | |
| | DATA | | Set Data as the I ² C search trigger condition. | |



commands

| | ADDRANDD <i>i</i> | | Set Address a search trigger | nd Data as the I ² C condition. |
|--|--|--------|-----------------------------------|--|
| Return parameter | Returns the I ² C bus search trigger condition. | | | |
| Example | :SEARCH:TRIGger:BUS:B1:I2C:CONDition ADDRess | | | |
| | Set Address | as the | e I ² C search tri | igger condition. |
| | | | | |
| :SEARCH:TRIG | ger:BUS:B1: | :12C:/ | ADDRess | Set |
| :MODe | | | | → Query |
| Description | Sets or querion bits) for the s | | | ng mode (7 or 10 |
| Syntax | :SEARCH:TRI {ADDR7 AD | | | DDRess:MODe |
| Related commands | :SEARCH:TRIGger:BUS:B1:I2C:CONDition | | | |
| Parameter | ADDR7 | 7 bit | addressing | |
| | ADDR10 | 10 bi | t addressing | |
| Return Parameter | 0 | 7 bit | addressing | |
| | 1 | 10 bi | t addressing | |
| Example | :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:MODe? | | | |
| | The addressing mode is current set to 7 bits. | | | |
| :SEARCH:TRIGger:BUS:B1:I2C:ADDRess :TYPe | | | | |
| Description | | | ddress type, or earch trigger. | r queries what the |
| Syntax | :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:TYPe {GENeralcall STARtbyte HSmode EEPROM CBUS ?} | | | |
| Related | :SEARCH:TRIGger:BUS:B1:I2C:CONDition | | | |



| Parameter | GENeralcall | Set a general call address (0000 000 0). |
|------------------------|---|---|
| | STARtbyte | Set a start byte address. (0000 000 1) |
| | HSmode | Set a high-speed mode address. (0000 1xx x) |
| | EEPROM | Set an EEPROM address. (1010 xxx x) |
| | CBUS | Set a CBUS address. (0000 001 x) |
| Return Parameter | Returns the | address type |
| Example | :SEARCH:TR CBUS | IGger:BUS:B1:I2C:ADDRess:TYPe? |
| :SEARCH:TRIG :VALue | ger:BUS:B1 | :I2C:ADDRess Set → Query |
| Description | Sets or queries the I ² C bus address value when the I ² C search trigger is set to trigger on Address or Address/Data. | |
| Syntax | :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:VALue { <string> ? }</string> | |
| Related commands | :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:MODe | |
| Parameter | <sting> 7/10 characters, must be enclosed in double quotes "string".</sting> | |
| | | x = don't care |
| | | 1 = binary 1 |
| | | 0 = binary 0 |
| Return Parameter | Returns the address value in binary. | |
| Example 1 | :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:MODe ADDR7 :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:VALue "xxx0101" | |
| | Sets the address to XXX0101 | |



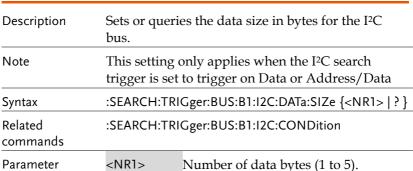
| Example 2 | :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:VALue? XXX0101 | | |
|------------------|--|---|--|
| :SEARCH:TRI | Gger:BUS:B | 1:I2C:ADDRess Set → Query | |
| Description | - | Sets or queries the address bit as read write or don't care for the search function. | |
| Note | trigger is se | This setting only applies when the I ² C search trigger is set to trigger on Address or Address/Data | |
| Syntax | | :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:DIRection { READ WRITE NOCARE ? } | |
| Related commands | :SEARCH:TRIGger:BUS:B1:I2C:CONDition | | |
| Parameter | READ | Set read as the data direction. | |
| | WRITE | Set write as the data direction. | |
| | NOCARE | Set either as the data direction. | |
| Return Paramete | r Returns the | Returns the direction (READ, WRITE, NOCARE). | |

Example :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:DIRection

READ

Sets the direction to READ.

$: SEARCH: TRIGger: BUS: B1: I2C: DATa: SIZe \longrightarrow \bigcirc Query$



Set



| Return parameter | <nr1></nr1> | Returns the number of bytes. |
|--|--|---|
| Example | :SEARCH:TRIGger:BUS:B1:I2C:DATa:SIZe 3 | |
| | Sets the num | aber of bytes to 3. |
| | | Set → |
| :SEARCH:TRIG | ger:BUS:B1 | :I2C:DATa:VALue → Query |
| Description | | ies the triggering data value for the I ² C e I ² C search trigger is set to trigger on lress/Data. |
| Syntax | :SEARCH:TR { <string> ?</string> | IGger:BUS:B1:I2C:DATa:VALue } |
| Related commands | :SEARCH:TR | IGger:BUS:B1:I2C:DATa:SIZe |
| Parameter | <string></string> | The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string". x = don't care 1 = binary 1 |
| | | 0 = binary 0 |
| Return Parameter | Returns the data value. | |
| Example 1 | :SEARCH:TRIGger:BUS:B1:I2C:DATa:SIZe 1 | |
| | :SEARCH:TRIGger:BUS:B1:I2C:DATa:VALue "1x1x0101" | |
| | Sets the value to XXX0101 | |
| Example 2 | :SEARCH:TRIGger:BUS:B1:I2C:DATa:VALue? | |
| | 1X1X0101 | |
| :SEARCH:TRIGger:BUS:B1:UART :CONDition Set → Query | | |
| Description | Sets or queri condition. | ies the UART search triggering |



Related

commands

| Syntax | :SEARCH:TRIGger:BUS:B1:UART:CONDition { RXSTArt RXDATA RXENDPacket TXSTArt TXDATA TXENDPacket TXPARItyerr RXPARItyerr ? } | |
|--|---|---|
| Parameter | RXSTArt | Set search trigger on the RX Start Bit. |
| | RXDATA | Set search trigger on RX Data. |
| | RXENDPacket | Set search trigger on the RX End of Packet condition. |
| | RXPARItyerr | Set search trigger on RX Parity error condition. |
| | TXSTArt | Set search trigger on the TX Start Bit. |
| | TXDATA | Set search trigger on TX Data. |
| | TXENDPacket | Set search trigger on the TX End of Packet condition. |
| | TXPARItyerr | Set search trigger on TX Parity error condition. |
| Return Parameter | Returns the search triggering condition. | |
| Example | :SEARCH:TRIGger:BUS:B1:UART:CONDition TXDATA | |
| | Sets the UART bus to trigger on Tx Data for the search function. | |
| :SEARCH:TRIGger:BUS:B1:UART:RX:DATa: Set → Query | | |
| Description | Sets or queries | the number of bytes for UART data. |
| Note | This setting only applies when the UART search trigger is set to trigger on Rx Data | |
| Syntax | :SEARCH:TRIGger:BUS:B1:UART:RX:DATa:SIZe { <nr1> ?}</nr1> | |

:SEARCH:TRIGger:BUS:B1:UART:CONDition



| | | GDS-2000E Programming Manual | |
|-----------------------|---|--|--|
| Parameter | <nr1></nr1> | Number of bytes (1 to 10). | |
| Return parameter | <nr1></nr1> | Returns the number of bytes. | |
| Example | | IGger:BUS:B1:UART:RX:DATa:SIZe 5 nber of bytes to 5. | |
| :SEARCH:TRIG VALue | ger:BUS:B1 | :UART:RX:DATa: Set → Query | |
| Description | - | ies the search triggering data value for us when the bus is set to trigger on Rx | |
| Syntax | :SEARCH:TRIGger:BUS:B1:UART:RX:DATa:VALue { <string> ? }</string> | | |
| Related commands | :SEARCH:TR | IGger:BUS:B1:UART:RX:DATa:SIZe | |
| Parameter | <sting></sting> | The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string". | |
| | | x = don't care | |
| | | 1 = binary 1 | |
| | | 0 = binary 0 | |
| Return Parameter | Returns the data value. | | |
| Example1 | $: SEARCH: TRIGger: BUS: B1: UART: CONDition\ RXDATA$ | | |
| | :SEARCH:TRIGger:BUS:B1:UART:RX:DATa:SIZe 1 | | |
| | :SEARCH:TRIGger:BUS:B1:UART:RX:DATa:VALue "1x1x0101" | | |
| | Sets the valu | ue to 1x1x0101 | |
| | | | |

:SEARCH:TRIGger:BUS:B1:UART:RX:DATa:VALue?

Example 2

1X1X0101



:SEARCH:TRIGger:BUS:B1:UART:TX:DATa :SIZe



| Description | Sets or queries the number of bytes for UART data. | | |
|------------------|---|------------------------------|--|
| Note | This setting only applies when the UART search trigger is set to trigger on Tx Data | | |
| Syntax | :SEARCH:TRIGger:BUS:B1:UART:TX:DATa:SIZe { <nr1> ?}</nr1> | | |
| Related commands | :SEARCH:TRIGger:BUS:B1:UART:CONDition | | |
| Parameter | <nr1></nr1> | Number of bytes (1 to 10). | |
| Return parameter | <nr1></nr1> | Returns the number of bytes. | |
| Example | :SEARCH:TRIGger:BUS:B1:UART:TX:DATa:SIZe 5 Sets the number of bytes to 5. | | |

:SEARCH:TRIGger:BUS:B1:UART:TX:DATa: \longrightarrow Query

| Description | - | Sets or queries the search triggering data value for the UART bus when the bus is set to trigger on Tx Data. | |
|------------------|-----------------|--|--|
| Syntax | | :SEARCH:TRIGger:BUS:B1:UART:TX:DATa:VALue { <string> ? }</string> | |
| Related commands | :SEARCH:1 | TRIGger:BUS:B1:UART:TX:DATa:SIZe | |
| Parameter | <sting></sting> | The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string". | |
| | | x = don't care | |
| | | 1 = binary 1 | |
| | | 0 = binary 0 | |



| Return Parameter | Returns the da | ata value. | |
|------------------|---|---|--|
| Example 1 | :SEARCH:TRIG | :SEARCH:TRIGger:BUS:B1:UART:CONDition TXDATA | |
| | :SEARCH:TRIG | ger:BUS:B1:UART:TX:DATa:SIZe 1 | |
| | :SEARCH:TRIG | ger:BUS:B1:UART:TX:DATa:VALue | |
| | Sets the value | to 1x1x0101 | |
| Example 2 | :SEARCH:TRIG | ger:BUS:B1:UART:TX:DATa:VALue? | |
| | 1X1X0101 | | |
| | | Set → | |
| :SEARCH:TRIC | iger:BUS:B1:S | SPI:CONDition → Query | |
| Description | Sets or queries | s the SPI search triggering condition. | |
| Syntax | :SEARCH:TRIGger:BUS:B1:SPI:CONDition {SS MISO MOSI MISOMOSI ? } | | |
| Parameter | SS | Set to trigger on the Slave Selection condition. | |
| | MISO | Set to trigger on the Master-In Slave-Out condition. | |
| | MOSI | Set to trigger on the Master-Out Slave-In condition. | |
| | MISOMOSI | Set to trigger on the Master-In Slave-Out and Master-Out Slave-In conditions. | |
| Return Parameter | Returns the triggering condition. | | |
| Example | :SEARCH:TRIGger:BUS:B1:SPI:CONDition MISO | | |
| • | | us to trigger on MISO. | |
| | | (Set)→ | |
| :SEARCH:TRIC | ger:BUS:B1:S | | |
| Description | Sets or queries the number of words for SPI data for the search function. | | |



| Note | This setting only applies when the SPI search trigger is set to trigger on MISO, MOSI or MISO/MOSI | |
|---------------------|--|------------------------------|
| Syntax | :SEARCH:TRIGger:BUS:B1:SPI:DATa:SIZe { <nr1> ?}</nr1> | |
| Related commands | :SEARCH:TRIGger:BUS:B1:SPI:CONDition | |
| Parameter | <nr1></nr1> | Number of words (1 to 32). |
| Return parameter | <nr1></nr1> | Returns the number of words. |
| Example | :SEARCH:TRIGger:BUS:B1:SPI:DATa:SIZe 10 Sets the number of words to 10. | |

| :SEARCH:TRIGger:BUS:B1:SPI:DATa:MISO: | Set → |
|---------------------------------------|---------|
| VALue | → Query |

| Description | Sets or queries the search triggering data value for the SPI bus when the bus is set to trigger on MISO or MISO/MOSI. | |
|------------------|---|--|
| Syntax | :SEARCH:TRIGger:BUS:B1:SPI:DATa:MISO:VALue { <string> ? }</string> | |
| Related commands | :SEARCH:TRIGger:BUS:B1:SPI:DATa:SIZe | |
| Parameter | <sting></sting> | The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string". |
| | | x = don't care |
| | | 1 = binary 1 |
| | | 0 = binary 0 |

Return Parameter Returns the data value.



| Example 1 | :SEARCH:TI | RIGger:BUS:B1:SPI:CONDition MISO | |
|------------------|--|--|--|
| | :SEARCH:TRIGger:BUS:B1:SPI:DATa:SIZe 2 | | |
| | :SEARCH:TI "1x1x0101" | RIGger:BUS:B1:SPI:DATa:MISO:VALue | |
| | Sets the val | ue to 1x1x0101 | |
| Example 2 | :SEARCH:TI | RIGger:BUS:B1:SPI:DATa:MISO:VALue? | |
| | 1X1X0101 | | |
| :SEARCH:TRIC | ger:BUS:B | 1:SPI:DATa:MOSI: Set → Query | |
| Description | | ries the search triggering data value for when the bus is set to trigger on MOSI MOSI. | |
| Syntax | :SEARCH:TRIGger:BUS:B1:SPI:DATa:MOSI:VALue { <string> ? }</string> | | |
| Related commands | :SEARCH:TI | RIGger:BUS:B1:SPI:DATa:SIZe | |
| Parameter | <sting></sting> | The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string". | |
| | | x = don't care | |
| | | 1 = binary 1 | |
| | | 0 = binary 0 | |
| Return Parameter | Returns the data value. | | |
| Example1 | :SEARCH:TRIGger:BUS:B1:SPI:CONDition MOSI | | |
| | :SEARCH:TRIGger:BUS:B1:SPI:DATa:SIZe 2 | | |
| | :SEARCH:TI "1x1x0101" | :SEARCH:TRIGger:BUS:B1:SPI:DATa:MOSI:VALue "1x1x0101" | |
| | Sets the val | ue to 1x1x0101 | |



| Example2 | :SEARCH:TRIGger:BUS:B1:SPI:DATa:MOSI:VALue? |
|----------|---|
| | 1X1X0101 |

| :SEARCH:TRIG | ger:BUS:B1: | Set → CAN:CONDition → Query |
|-----------------------------|--|--|
| Description | Sets or returns the CAN search trigger condition. | |
| Syntax | :SEARCH:TRIGger:BUS:B1:CAN:CONDition {SOF FRAMEtype IDentifier DATA IDANDDATA EOF ACKMISS STUFFERR ?} | |
| Parameter/ Return parameter | SOF | Sets search to trigger on a start of frame |
| | FRAMEtype | Sets search to trigger on the type of frame |
| | Identifier | Sets search to trigger on a matching identifier |
| | DATA | Sets search to trigger on matching data |
| | IDANDDATA | Sets search to trigger on matching identifier and data field |
| | EOF | Sets search to trigger on the end of frame |
| | ACKMISS | Sets search to trigger on a missing acknowledge |
| | STUFFERR | Sets search to trigger on a bit stuffing error |
| Example1 | :SEARCH:TRI | Gger:BUS:B1:CAN:CONDition SOF |
| | Triggers sear | rch on a start of frame. |
| Example2 | :SEARCH:TRI >SOF | Gger:BUS:B1:CAN:CONDition? |



| :SEARCH:TRIG | ger:BUS:B1 | Set → :CAN:FRAMEtype → Query |
|------------------|---------------|---|
| Description | | ns the frame type for the CAN e search trigger. |
| Syntax | | Gger:BUS:B1:CAN:FRAMEtype te ERRor OVERLoad ?} |
| Parameter/ | DATA | Sets the frame type to data frame |
| Return parameter | REMote | Sets the frame type to remote frame |
| | ERRor | Sets the frame type to error frame |
| | OVERLoad | Sets the frame type to overload |
| Example | :SEARCH:TRI | Gger:BUS:B1:CAN:FRAMEtype DATA |
| | Sets the fram | ne type to DATA. |

| :SEARCH:TRIGger:BUS:B1:CAN:IDentifier: | Set |
|--|---------|
| MODe | → Query |

| Description | Sets or returns the CAN identifier mode for the bus. | |
|------------------|--|--------------------------|
| Syntax | :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODe {STANDard EXTended ?} | |
| Parameter/ | STANDard | Standard addressing mode |
| Return parameter | EXTended | Extended addressing mode |
| Example | :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODe? >STANDARD | |
| | :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODe EXTENDED :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODe? >EXTENDED | |
| | | |



| :SEARCH:TRIG VALue | ger:BUS:B1 | :CAN:IDentifier: Set → Query |
|--------------------------------|---------------------------------------|--|
| Description | Sets or retur CAN search | ns the identifier string used for the trigger. |
| | | applicable when the search trigger set to ID or IDANDDATA. |
| Syntax | :SEARCH:TRI { <string> ?}</string> | IGger:BUS:B1:CAN:IDentifier:VALue |
| Related Commands | :SEARCH:TR | IGger:BUS:B1:CAN:IDentifier:MODe |
| Parameter/ Return parameter | <string></string> | The size of the string depends on the data size setting. The string must be enclosed in double quotes, "string". |
| | | String contents: |
| | | x = don't care |
| | | 1 = binary 1 |
| | | 0 = binary 0 |
| Example | :SEARCH:TR | IGger:BUS:B1:CAN:CONDition ID |
| | :SEARCH:TRI STANDARD | IGger:BUS:B1:CAN:IDentifier:MODe |
| | :SEARCH:TRI | IGger:BUS:B1:CAN:IDentifier:VALue X" |
| | :SEARCH:TRI >01100X1X0 | IGger:BUS:B1:CAN:IDentifier:VALue? IX |
| :SEARCH:TRIG :DIRection | ger:BUS:B1 | :CAN:IDentifier |
| Description | Sets or queri don't care. | ies the address bit as read, write or |
| Syntax | | Gger:BUS:B1:CAN:IDentifier:DIRection E NOCARE ?} |



| GEN IN IN IEK | L | GDS-2000E Programming Manual |
|--------------------------------|--|--|
| Parameter/ Return parameter | READ WRITE NOCARE | Sets read as the data direction Sets write as the data direction Sets either as the data direction |
| Example2 | :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:DIRection? >WRITE :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:DIRection READ | |
| | :SEARCH:TR >READ | RIGger:BUS:B1:CAN:IDentifier:DIRection? |
| :SEARCH:TRIG QUALifier | ger:BUS:B1 | I:CAN:DATa: Set → Query |
| Description | Note: Only | rns the CAN data qualifier. applicable when the search triggering set to DATA or IDANDDATA. |
| Syntax | | RIGger:BUS:B1:CAN:DATa:QUALifier MOREthan EQual UNEQual LESSEQual M } |
| Parameter/ Return parameter | LESSthan | Sets search to trigger when the data is less than the qualifier value. |
| | MOREthan | Sets search to trigger when the data is greater than the qualifier value. |

EQual Sets search to trigger when the data is equal to the qualifier value. **UNEQual** Sets search to trigger when the data is not equal to the qualifier value. Sets search to trigger when the data is LESSEQual less than or equal to the qualifier value. MOREEQual Sets search to trigger when the data is more than or equal to the qualifier

value.



| Example | :SEARCH:TRI >EQUAL | Gger:BUS:B1:CAN:DATa:QUALifier? | |
|--------------------------------|--|--|--|
| | :SEARCH:TRI MOREthan | Gger:BUS:B1:CAN:DATa:QUALifier | |
| | :SEARCH:TRI >MORETHAN | Gger:BUS:B1:CAN:DATa:QUALifier? N | |
| | | (Set)→ | |
| :SEARCH:TRIG | ger:BUS:B1 | :CAN:DATa:SIZe → Query | |
| Description | Sets or returns the length of the data string in bytes for the CAN search trigger. | | |
| | Note: Only applicable when the condition is set to DATA or IDANDDATA. | | |
| Syntax | :SEARCH:TRI | Gger:BUS:B1:CAN:DATa:SIZe { <nr1> ?}</nr1> | |
| Parameter/ Return parameter | <nr1></nr1> | 1~8 (bytes) | |
| Example | :SEARCH:TRI >1 | Gger:BUS:B1:CAN:DATa:SIZe? | |
| | :SEARCH:TRI | Gger:BUS:B1:CAN:DATa:SIZe 2 | |
| | :SEARCH:TRI >2 | Gger:BUS:B1:CAN:DATa:SIZe? | |
| :SEARCH:TRIG | aor: DI IC: D1 | ·CAN·DATa· (Set) | |
| VALue VALue | ger.bO3.bT | Query | |
| Description | Sets or return the CAN sea | ns the binary data string to be used for arch trigger. | |
| Related Commands | :SEARCH:TRI | Gger:BUS:B1:CAN:DATa:SIZe | |
| Syntax | :SEARCH:TRI { <string> ?}</string> | Gger:BUS:B1:CAN:DATa:VALue | |
| Parameter/ Return parameter | <string></string> | The size of the string depends on the data size setting. The string must be enclosed in double quotes, "string". | |
| | | | |



| | | String contents: x = don't care | | |
|--------------------------------|--|--|--|--|
| | | 1 = binary 1 | | |
| | 0 = binary 0 | | | |
| Example | :SEARCH:TRI | Gger:BUS:B1:CAN:DATa:SIZe 1 | | |
| | :SEARCH:TRIGger:BUS:B1:CAN:DATa:VALue "01010X1X" | | | |
| | :SEARCH:TRI >01010X1X | :SEARCH:TRIGger:BUS:B1:CAN:DATa:VALue? >01010X1X | | |
| :SEARCH:TRIG | ger:BUS:B1: | Set → :LIN:CONDition → Query | | |
| Description | Sets or return | ns the LIN search trigger condition. | | |
| Syntax | | Gger:BUS:B1:LIN:CONDition Dentifier DATA IDANDDATA WAKEup SL | | |
| Parameter/ Return parameter | SYNCField | Sets the LIN search trigger condition to the sync field. | | |
| | IDentifier | Sets the LIN search trigger condition to identifier field. | | |
| | DATA | Sets the LIN search trigger condition to the data field. | | |
| | IDANDDATA | Sets the LIN search trigger condition to identifier and data field | | |
| | WAKEup | Sets the LIN search trigger condition to wake up. | | |
| | SLEEP | Sets the LIN search trigger condition to sleep. | | |
| | ERRor | Sets the LIN search trigger condition | | |

to error.



Example :SEARCH:TRIGger:BUS:B1:LIN:CONDition?
>IDANDDATA
:SEARCH:TRIGger:BUS:B1:LIN:CONDition DATA
:SEARCH:TRIGger:BUS:B1:LIN:CONDition?
>DATA

| :SEARCH:TRIG QUALifier | ger:BUS:B1 | :LIN:DATa: Set → Query |
|--------------------------------|--------------|---|
| Description | Note: Only a | ns the LIN data qualifier. applicable when the search trigger set to DATA or IDANDDATA. |
| Syntax | | Gger:BUS:B1:LIN:DATa:QUALifier OREthan EQual UNEQual LESSEQual M |
| Parameter/ Return parameter | LESSthan | Sets search to trigger when the data is less than the qualifier value. |
| | MOREthan | Sets search to trigger when the data is greater than the qualifier value. |
| | EQual | Sets search to trigger when the data is equal to the qualifier value. |
| | UNEQual | Sets search to trigger when the data is not equal to the qualifier value. |
| | LESSEQual | Sets search to trigger when the data is less than or equal to the qualifier value. |
| | MOREEQual | Sets search to trigger when the data is more than or equal to the qualifier value. |



| Example | :SEARCH:TRI >EQUAL | Gger:BUS:B1:LIN:DA | Ta:QUALifier? |
|--------------------------------|--|--------------------------------------|--------------------------|
| | :SEARCH:TRIGger:BUS:B1:LIN:DATa:QUALifier MOREthan | | |
| | :SEARCH:TRIGger:BUS:B1:LIN:DATa:QUALifier? >MORETHAN | | |
| | | | Set → |
| :SEARCH:TRIG | ger:BUS:B1: | LIN:DATa:SIZe | Query |
| Description | Sets or returns the length of the data string in bytes for the LIN search trigger. | | |
| | Note: Only applicable when the condition is set to DATA or IDANDDATA. | | |
| Syntax | :SEARCH:TRI | Gger:BUS:B1:LIN:DA | Ta:SIZe { <nr1> ?}</nr1> |
| Parameter/ Return parameter | <nr1></nr1> | 1~8 (bytes) | |
| Example | :SEARCH:TRI >1 | Gger:BUS:B1:LIN:DA | Ta:SIZe? |
| | :SEARCH:TRI | Gger:BUS:B1:LIN:DA | Ta:SIZe 2 |
| | :SEARCH:TRI >2 | Gger:BUS:B1:LIN:DA | Ta:SIZe? |
| CEADOU TRIC | DUC DI | LINIDAT | |
| :SEARCH:TRIG VALue | ger:BUS:BT: | LIN:DATa: | Set → Query |
| Description | Sets or return the LIN sear | ns the binary data st ch trigger. | ring to be used for |
| | Note: Only a DATA or ID | pplicable when the ANDDATA. | condition is set to |
| Related Commands | :SEARCH:TRI | Gger:BUS:B1:LIN:DA | Ta:SIZe |
| Syntax | :SEARCH:TRI { <string> ?}</string> | Gger:BUS:B1:LIN:DA | Ta:VALue |
| | | | |



| Parameter/ Return parameter | <string></string> | The size of the string depends on the data size setting. The string must be enclosed in double quotes, "string". | |
|--------------------------------|--|--|--|
| | | String contents: | |
| | | x = don't care | |
| | | 1 = binary 1 | |
| | | 0 = binary 0 | |
| Example | :SEARCH:TRIGger:BUS:B1:LIN:DATa:SIZe 1 | | |
| | :SEARCH:TRIGger:BUS:B1:LIN:DATa:VALue "01010X1X" | | |
| | :SEARCH:TRI >01010X1X | Gger:BUS:B1:LIN:DATa:VALue? | |
| | | (Set)→ | |
| :SEARCH:TRIG | ger:BUS:B1 | :LIN:ERRTYPE —Query | |
| Description | Sets or return search trigge | ns the error type to be used for the LIN er. | |
| Syntax | :SEARCH:TRIGger:BUS:B1:LIN:ERRTYPE {SYNC PARIty CHecksum ?} | | |
| Parameter/ | SYNC | Sets the LIN error type to SYNC. | |
| Return parameter | PARIty | | |
| morani panamere. | FAIRITY | Sets the LIN error type to parity. | |
| riciani parameter | CHecksum | Sets the LIN error type to parity. Sets the LIN error type to checksum. | |
| Example | CHecksum | 71 1 7 | |
| | CHecksum :SEARCH:TRI | Sets the LIN error type to checksum. | |

>CHECKSUM



| :SEARCH:TRIGger:BUS:B1:LIN:IDentifier: | |
|--|--|
| VALue | |



| **** | | | |
|--------------------------------|--|--|--|
| Description | Sets or returns the identifier string to be used for the LIN search trigger. | | |
| | Note: Only applicable when the condition is set to ID or IDANDDATA. | | |
| Syntax | :SEARCH:TRIGger:BUS:B1:LIN:IDentifier:VALue { <string> ?}</string> | | |
| Parameter/ Return parameter | <string></string> | The size of the string depends on the data size setting. The string must be enclosed in double quotes, "string". | |
| | | String contents: | |
| | | x = don't care | |
| | | 1 = binary 1 | |
| | | 0 = binary 0 | |
| Example | :SEARCH:TRIGger:BUS:B1:LIN:CONDition ID | | |
| | :SEARCH:TRIGger:BUS:B1:LIN:IDentifier:VALue "00X1X01X" | | |

>01100X1X01X

:SEARCH:FFTPeak:METHod



Description Sets or returns the FFT peak method type. Related :SEARCH:TRIGger:TYPe Commands :SEARCH:FFTPeak:METHod:MPEak :SEARCH:TRIGger:LEVel Syntax :SEARCH:FFTPeak:METHod {MPEak | LEVel | ?}

:SEARCH:TRIGger:BUS:B1:LIN:IDentifier:VALue?



| Parameter/ Return parameter | MPEak | Sets the peak meth type. | od to the Max Peak |
|--------------------------------|--------------------------------------|---|--------------------|
| | LEVel | Sets the peak meth type. | ods to the Level |
| Example | :SEARCH:FFTPeak:METHod LEVel | | I |
| | :SEARCH:FFTPeak:METHod? | | |
| | >LEVEL | | |
| | :SEARCH:TRIGger:LEVel? | | |
| | >1.000E+00 | | |
| | :SEARCH:TRIGger:LEVel 2 | | |
| | :SEARCH:TRIGger:LEVel? >2.000E+00 | | |
| | | | |
| | | | Set → |
| :SEARCH:FFTP | eak:METHc | od:MPEak | → Query |
| Description | | ve peak number (1 ~ f the active peak nu | , |

| Description | Sets the active peak number (1 \sim 10) or return the frequency of the active peak number. | | |
|---------------------|--|-------------------------------|--|
| Related Commands | :SEARCH:TRIGger:TYPe :SEARCH:FFTPeak:METHod | | |
| Syntax | :SEARCH:FFTPeak:METHod:MPEak { <nr1> ?}</nr1> | | |
| Parameter | <nr1></nr1> | Active peak number. | |
| Return parameter | <nr3></nr3> | Frequency of the active peak. | |
| Example | :SEARCH:FFTPeak:METHod MPEak | | |
| | :SEARCH:FFTPeak:METHod? | | |
| | >MPEAK | | |
| | :SEARCH:FFTPeak:METHod:MPEak? | | |
| | >1.000E+00 | | |
| | Peak:METHod:MPEak 2 | | |
| | :SEARCH:FFTPeak:METHod:MPEak? | | |
| | >2.000E+00 | | |



Set)→ :SEARCH:FFTPeak:SINFo (Query Sets or returns the State Info to "Mark" or "Peak". Description Related :SEARCH:TRIGger:TYPe Commands Syntax :SEARCH:FFTPeak:SINFo {MARK | PEAK | ?} Parameter/ MARK Sets the State Info to Mark. Return parameter PEAK Sets the State Info to Peak. Example :SEARCH:FFTPeak:SINFo? >PEAK :SEARCH:FFTPeak:SINFo mark :SEARCH:FFTPeak:SINFo? >MARK



Label Commands

| Laber Commanus | | | |
|---------------------|---|--|--|
| | :CHANnel <x: :REF<x>:LAB :REF<x>:LAB :BUS1:LABel :BUS1:LABel</x></x></x: | >:LABel | |
| CHANnel <x>:</x> | LABel | Set → Query | |
| Description | Sets or returned channel. | ns the file label for the selected | |
| Syntax | :CHANnel <x>:LABel {<string> ?}</string></x> | | |
| Related commands | :CHANnel <x>:LABel:DISPlay</x> | | |
| Parameter | <x></x> | Channel 1, 2, 3, 4 | |
| | <string></string> | The string must be no more than 8 characters and only contain alphanumeric characters in addition to period, dash and underscore characters. The string must be enclosed in double quotes, "string". | |
| Return parameter | <string></string> | Returns the label for the selected channel. No return indicates that there has not been a file label assigned for the selected channel. | |
| Example1 | :CHANnel1:LABel "CH1_lab" | | |
| | Sets the char | nnel 1 label as "CH1_lab". | |
| Example2 | :CHANnel1:L | ABel? | |
| | | | |



| :CHANnel <x>:LABel:DISPlay \longrightarrow Query</x> | | | |
|---|--|--|--|
| Description | Turns the label on/off for the selected channel or returns its status. | | |
| Syntax | :CHANnel <x>:LABel:DISPlay { OFF ON ? }</x> | | |
| Related commands | :CHANnel <x>:LABel</x> | | |
| Parameter | <x></x> | Channel 1, 2, 3, 4 | |
| | OFF | Turns the file label off for the selected channel. | |
| | ON | Turns the file label on for the selected channel. | |
| Return parameter | Returns the status of the file label for the selected channel (ON, OFF). | | |
| Example | :CHANnel1:LABel "CH1" | | |
| | :CHANnel1:LABel:DISPlay ON | | |
| | :CHANnel1:LABel:DISPlay? ON | | |
| | | nnel 1 label to "CH1" and then turns play on. The query return shows that n. | |
| | | Set | |
| :REF <x>:LABel</x> | | → Query | |
| Description | Sets or returns the file label for the selected reference waveform. | | |
| Syntax | :REF <x>:LABel {<string> ?}</string></x> | | |
| Related commands | :REF <x>:LABel:DISPlay</x> | | |
| Parameter | <x></x> | REF 1, 2, 3, 4 | |



| | <string></string> | The string must be no more than 8 characters and only contain alphanumeric characters in addition to period, dash and underscore characters. The string must be enclosed in double quotes, "string". | |
|---------------------|---|--|--|
| Return parameter | <string></string> | Returns the label for the selected reference waveform. No return indicates that there has not been a file label assigned for the selected reference waveform. | |
| Example1 | :REF1:LABel ' | 'REF1_lab" | |
| | Sets the REF | 1 label as "REF1_lab". | |
| Example2 | :REF1:LABel? | | |
| | REF1_lab | | |
| :REF <x>:LABel</x> | :DISPlay | Set → Query | |
| Description | | pel on/off for the selected reference returns its status. | |
| Syntax | :REF <x>:LABel:DISPlay { OFF ON ? }</x> | | |
| Related commands | :REF <x>:LABel</x> | | |
| Parameter | <x></x> | Reference waveform 1, 2, 3, 4 | |
| | OFF | Turns the file label off for the selected reference waveform. | |
| | ON | Turns the file label on for the selected reference waveform. | |

Return parameter Returns the status of the file label for the selected reference waveform (ON, OFF).



Example :REF1:LABel "REF1" :REF1:LABel:DISPlay ON :REF1:LABel:DISPlay? ON Sets the label for reference waveform 1 to "REF1" and then turns the label display on. The query return shows that the label is on. Set) → Query :BUS1:LABel Description Sets or returns the file label for the bus. :BUS1:LABel {<string> | ?} Syntax Related :BUS1:LABel:DISPlay commands Parameter <string> The string must be no more than 8 characters and only contain alphanumeric characters in addition to period, dash and underscore characters. The string must be enclosed in double quotes, "string". Returns the label for the bus. No Return parameter <string> return indicates that there has not been a file label assigned for bus. Example1 :BUS1:LABel "Bus" Sets the bus label as "Bus". Example 2 :BUS1:LABel? Bus :BUS1:LABel:DISPlay Query Turns the label on/off for the bus or returns its Description status.

:BUS1:LABel:DISPlay { OFF | ON | ? }

Syntax



Example1

| Related commands | :BUS1:LABel | | |
|---------------------|--|--|--|
| Parameter | OFF | Turns the file label off for the bus. | |
| | ON | Turns the file label on for the bus. | |
| Return parameter | Returns the s OFF). | status of the file label for the bus (ON, | |
| Example | :BUS1:LABel | "Bus" | |
| | :BUS1:LABel: | DISPlay ON | |
| | :BUS1:LABel: ON | DISPlay? | |
| | | l for the bus to "Bus" and then turns play on. The query return shows that n. | |
| | | Set → | |
| :SET <x>:LABel</x> | | → (Query) | |
| Description | Sets or returns the file label for the selected setup. | | |
| Syntax | :SET <x>:LABel {<string> ?}</string></x> | | |
| Related commands | :SET <x>:LABel:DISPlay</x> | | |
| Parameter | <x></x> | Setup number 1 to 20 | |
| | <string></string> | The string must be no more than 8 characters and only contain alphanumeric characters in addition to period, dash and underscore characters. The string must be enclosed in double quotes, "string". | |
| Return parameter | <string></string> | Returns the label for the selected setup. No return indicates that there has not been a file label assigned for the selected setup. | |

:SET1:LABel "SET1_lab"

Sets the label for setup 1 as "SET1_lab".



Example2 :SET1:LABel?

SET1_lab

Segment Commands

| :SEGMents:CURRent | :SEGMents:STATE | 221 |
|--------------------------------|--------------------------------|-----|
| :SEGMents:TIMe | :SEGMents:CURRent | 222 |
| :SEGMents:DISPALL | :SEGMents:TOTalnum | 222 |
| :SEGMents:MEASure:MODe | :SEGMents:TIMe | 223 |
| :SEGMents:MEASure:PLOT:SOURce | :SEGMents:DISPALL | 223 |
| :SEGMents:MEASure:PLOT:DIVide | :SEGMents:MEASure:MODe | 223 |
| :SEGMents:MEASure:PLOT:SELect | :SEGMents:MEASure:PLOT:SOURce | 224 |
| :SEGMents:MEASure:PLOT:RESults | :SEGMents:MEASure:PLOT:DIVide | 224 |
| :SEGMents:MEASure:TABle:SOURce | :SEGMents:MEASure:PLOT:SELect | 225 |
| :SEGMents:MEASure:TABle:SELect | :SEGMents:MEASure:PLOT:RESults | 225 |
| :SEGMents:MEASure:TABle:LIST | :SEGMents:MEASure:TABle:SOURce | 226 |
| :SEGMents:MEASure:TABle:SAVe | :SEGMents:MEASure:TABle:SELect | 226 |
| :SEGMents:SAVe | :SEGMents:MEASure:TABle:LIST | 227 |
| :SEGMents:SAVe:SOURce | :SEGMents:MEASure:TABle:SAVe | 227 |
| :SEGMents:SAVe:SELect:STARt228 | :SEGMents:SAVe | 228 |
| | :SEGMents:SAVe:SOURce | 228 |
| :SEGMents:SAVe:SELect:END | :SEGMents:SAVe:SELect:STARt | 228 |
| | :SEGMents:SAVe:SELect:END | 229 |

:SEGMents:STATE



| Description | Turns the segmented memory function on/off or queries its state. | | |
|------------------|--|---------------------------------|--|
| Syntax | :SEGMents:STATE { OFF ON ? } | | |
| Related | :RUN | | |
| commands | :STOP | | |
| Parameter/ | OFF | Turns the segmented memory off. | |
| Return parameter | ON | Turns the segmented memory on. | |
| Example1 | :SEGMents:STATE ON | | |

Turns segmented memory on.



| :SEGMents:CU | RRent | | Set → Query |
|------------------|--|----------------------------------|-----------------|
| Description | Sets or queries the current segment. The total number of segments depends on the record length. | | |
| Syntax | :SEGMents:C {SETTOMIN | URRent SETTOMAX <nr1> </nr1> | ?} |
| Related commands | :SEGMents:S :SEGMents:T | | |
| Parameter/ | SETTOMIN | Current segment = | min segment |
| Return parameter | SETTOMAX | Current segment = | max segment |
| | <nr1></nr1> | 1~29000 | |
| Example1 | :SEGMents:C | URRent 10 | |
| | Sets the curr | ent segment to segr | nent number 10. |
| :SEGMents:TO | Talnum | | Set → Query |
| Description | Sets or queries the total number of segments for the segmented memory function. The total number of segments depends on the record length. | | |
| Syntax | :SEGMents:TOTalnum {SETTOMIN SETTOMAX <nr1> ?}</nr1> | | |
| Related | :SEGMents:STATE | | |
| commands | :SEGMents:CURRent | | |
| Parameter/ | SETTOMIN | Sets to the minimu | m number |
| Return parameter | SETTOMAX | Sets to the maximu | m number |
| | <nr1></nr1> | 1~29000 | |
| Example1 | :SEGMents:TOTalnum SETTOMAX | | |
| | Sets the number of segments to max number (29000). | | |



| :SEGMents:TIN | Лe | | → Query | |
|--------------------------------|---|----------------------|-------------------|--|
| Description | Returns the time of the current segment in relation to the first segment. | | | |
| Syntax | :SEGMents:Tl | IMe? | | |
| Related | :SEGMents:S | TATE | | |
| commands | :SEGMents:C | URRent | | |
| Return parameter | The segment | time as <nr3>.</nr3> | | |
| Example | :SEGMents:T | IMe? | | |
| | >8.040E-03 | | | |
| | Returns the s | segment time. | | |
| | | | Set → | |
| :SEGMents:DIS | SPALL | | → Query | |
| Description | Sets or queries whether all the segments are displayed on the screen. | | | |
| Syntax | :SEGMents:DISPALL {OFF ON ?} | | | |
| Related | :SEGMents:STATE | | | |
| commands | :SEGMents:CURRent | | | |
| Parameter/ | OFF | Turns the display a | all function off. | |
| Return parameter | ON | Turns the display a | all function on. | |
| Example1 | :SEGMents:D | ISPALL ON | | |
| | Turns the dis | splay all function o | n. | |
| | ∑Set → | | | |
| :SEGMents:MEASure:MODe → Query | | | | |
| Description | Sets or queri | es the measuremen | t mode. | |
| Syntax | :SEGMents:MEASure:MODe {OFF PLOT TABle ?} | | | |
| Related commands | :MEASUrement:MEAS <x></x> | | | |



| Parameter/ Return parameter | OFF | Disables the automatic measurement function for the segments measurement. | | |
|--------------------------------|---|---|--|--|
| | PLOT | Sets the measurement mode to Statistics. | | |
| | TABLE | Sets the measurement mode to a measurement list. | | |
| Example | :SEGMents:N >PLOT | //EASure:MODe? | | |
| | Returns the r | neasurement mode as Statistics. | | |
| | | Set → | | |
| :SEGMents:ME | ASure:PLO | T:SOURce → Query | | |
| Description | C-1 | the statistics are used | | |
| Description | | ies the statistics source. | | |
| Syntax | :SEGMents:MEASure:PLOT:SOURce { <nr1> ?}</nr1> | | | |
| Related | :SEGMents:MEASure:MODe | | | |
| commands | :SEGMents:N | MEASure:PLOT:DIVide | | |
| | :SEGMents:MEASure:PLOT:SELect | | | |
| | :SEGMents:N | MEASure:PLOT:RESults | | |
| Parameter/ Return parameter | <nr1></nr1> | 1~8 (Automatic measurement item 1~8) | | |
| Example1 | :SEGMents:N | MEASure:PLOT:SOURce 1 | | |
| | Sets the sour | ce as auto measurement item 1. | | |
| | | (Set)→ | | |
| :SEGMents:ME | ASure:PLO | T:DIVide → Query | | |
| Description | Sets or queri function. | ies the number of bins for the statistics | | |
| Syntax | :SEGMents:MEASure:PLOT:DIVide { <nr1> ? }</nr1> | | | |
| Related | ed :SEGMents:MEASure:PLOT:SOURce | | | |
| commands | :SEGMents:MEASure:PLOT:SELect | | | |
| | | | | |



| Parameter/ Return parameter | <nr1></nr1> | 1~20 |
|--------------------------------|-------------|------|
| | | |

Example 1:SEGMents:MEASure:PLOT:DIVide 5

Sets the number of bins to 5 for the statistics function.

$\begin{array}{ccc} & & & & & & \\ & & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ &$

| Description | Sets or queries which bin to view the statics of. | | |
|------------------|---|------------------------------------|--|
| Syntax | :SEGMents:MEASure:PLOT:SELect { <nr1> ? }</nr1> | | |
| Related | :SEGMents:MEASure:PLOT:SOURce :ds :SEGMents:MEASure:PLOT:DIVide | | |
| commands | | | |
| Parameter | <nr1> 1~20</nr1> | | |
| | | (cannot exceed the number of bins) | |
| Return parameter | Return the bin number as <nr3>.</nr3> | | |
| Example1 | :SEGMents:MEASure:PLOT:SELect 5 | | |
| | Set to bin nur | nber 5. | |

:SEGMents:MEASure:PLOT:RESults → Query)

| Description | Returns the results of the currently selected bin for the statistics measurement. |
|---------------------|---|
| Note | At least one automatic measurement must be turned on. |
| Syntax | :SEGMents:MEASure:PLOT:RESults? |
| Related commands | :SEGMents:STATE :SEGMents:MEASure:MODe PLOT :SEGMents:MEASure:PLOT:SOURce :SEGMents:MEASure:PLOT:DIVide :SEGMents:MEASure:PLOT:SELect |
| Return parameter | Returns the statistics measurements as a string. |



Example :SEGMents:STATE ON
STOP
:SEGMents:MEASure:MODe PLOT
:SEGMents:MEASure:PLOT:SOURce 1
:SEGMents:MEASure:PLOT:DIVide 10
:SEGMents:MEASure:PLOT:SELect 1
:SEGMents:MEASure:PLOT:RESults?
> MAX,1.000kHz;MIN,1.000kHz;MEAN,1.000kHz;
Bin Statistics,1 of 10;Percent,10.00%;Count,1;
Measured,10;Unmeasured,0;Bin Range,
1.000kHz~1.000kHz;

Plots the results for automatic measurement #1, bin 1 of 10.

:SEGMents:MEASure:TABle:SOURce



| .SEGIVICITES.IVIE | ., (5416.17 (51) | c.5001kcc | , (4,5,7) |
|------------------------------------|---|----------------------------------|-------------------|
| Description | Sets or queri | es the source of the | measurement list. |
| Syntax | :SEGMents:MEASure:TABle:SOURce {CH1 CH2 CH3 CH4 ? } | | |
| Related | :SEGMents:N | IEASure:MODe | |
| commands | :SEGMents:MEASure:TABle:SELect | | |
| | :SEGMents:N | IEASure:TABle:LIST | |
| Parameter/ Return parameter | CH1~CH4 | Channel 1 to 4 | |
| Example1 | :SEGMents:MEASure:TABle:SOURce CH1 | | |
| | Sets the sour | ce to CH1. | |
| : SEGMents: MEASure: TABle: SELect | | | |
| Description | Sets or querion | es a segment to vie tt table. | w in the |

:SEGMents:MEASure:TABle:SELect {<NR1> | ? }

Syntax



| Related commands | :SEGMents:TOTalnum | | | |
|------------------|--|-----------------------------|-------------------|--|
| Parameter | <nr1> 1~29000</nr1> | | | |
| Return parameter | Returns the n | umber of segments a | as <nr3>.</nr3> | |
| Example1 | :SEGMents:N | IEASure:TABle:SELec | t 10 | |
| | Select segme | nt number 10. | | |
| | | | | |
| :SEGMents:ME | ASure:TABl | e:LIST | → Query | |
| Description | Returns the i | neasurement result data. | s of each segment | |
| Syntax | :SEGMents:N | IEASure:TABle:LIST? | | |
| Return parameter | Returns the measurements results as a block data for each segment. | | | |
| Example | :SEGMents:MEASure:TABle:LIST? | | | |
| | >"GW GDS-2074E, serial number PXXXXXX, | | | |
| | version V1.37", | | | |
| | Segment Summary : CH1, Seg., Pk-Pk (V), Pk-Pk (V), | | | |
| | 1, 8.00m, 8.00 | | | |
| | 2, 8.00m, 8.00 | | | |
| | 3, 8.00m, 8.00 | | | |
| | 4, 8.00m, 8.00 | | | |
| | 5, 8.00m, 8.00 | 0m, | | |
| | 6, 8.00m, 8.00 | Om, | | |
| | 7, 8.00m, 8.00 | | | |
| | 8, 8.00m, 8.00 | | | |
| | 9, 12.0m, 12.0 | | | |
| | 10, 8.00m, 8.0 | JUm, | | |
| :SEGMents:ME | ASure:TABl | e:SAVe | Set → | |
| | | | | |

| Description | Saves the list of segment automatic measurement results. |
|-------------|--|
| Syntax | :SEGMents:MEASure:TABle:SAVe |



| :SEGMents:SAVe | | | Set → | |
|--------------------------------|--|----------------------|-----------------|--|
| Description | Saves the se | gments. | | |
| Syntax | :SEGMents:SAVe | | | |
| Related | :SEGMents:SAVe:SOURce | | | |
| Commands | :SEGMents:SAVe:SELect:STARt | | | |
| | :SEGMents:S | AVe:SELect:END | | |
| Example | :SEGMents:S | AVe:SOURce CH1 | | |
| | :SEGMents:S | AVe:SELect:STARt 1 | | |
| | :SEGMents:S | AVe:SELect:END 10 | | |
| | :SEGMents:S | AVe | | |
| :SEGMents:SA | Ve:SOURce | | Set → Query | |
| Description | Sets or queri save. | ies the source segme | ent waveform to | |
| Syntax | :SEGMents:SAVe:SOURce {CH1 CH2 CH3 CH4 ? } | | | |
| Parameter/ Return parameter | CH1~CH4 | Channel 1 to 4. | | |
| Example | :SEGMents:S | AVe:SOURce CH1 | | |
| | >Sets the source to CH1. | | | |
| :SEGMents:SA | $ \underbrace{\text{Set}} \rightarrow $ $:SEGMents:SAVe:SELect:STARt \longrightarrow \underbrace{\text{Query}} $ | | | |
| Description | Sets or queries the starting segment to save from. The number of possible segments depends on the record length. | | | |
| Syntax | :SEGMents:SAVe:SELect:STARt {SETTOMIN SETTOMAX <nr1> ? }</nr1> | | | |
| Related commands | :SEGMents:T | OTalnum | | |



| Parameter/ Return parameter | SETTOMIN | Sets the starting segment to min segment. | |
|--------------------------------|--|---|--|
| | SETTOMAX | Sets the starting segment to the max segment. | |
| | <nr1></nr1> | Sets the segment to 1~29000 | |
| Example | :SEGMents:S | AVe:SELect:STARt 2 | |
| | Sets the star | ting segment to segment number 2. | |
| | | Set → | |
| :SEGMents:SA | Ve:SELect:E | ND ——Query | |
| Description | Sets or queries the ending segment to save from. The number of possible segments depends on the record length. | | |
| Syntax | :SEGMents:SAVe:SELect:END {SETTOMIN SETTOMAX <nr1> ? }</nr1> | | |
| Related commands | :SEGMents:TOTalnum | | |
| Parameter/ Return parameter | SETTOMIN | Sets the starting segment to min segment. | |
| | SETTOMAX | Sets the starting segment to the max segment. | |
| | <nr1></nr1> | Sets the segment to 1~29000. | |
| Return parameter | <nr3></nr3> | Returns the ending segment as NR3. | |
| Example | :SEGMents:S | AVe:SELect:END 10 | |
| | Sets the ending segment to segment number 10. | | |



DVM Commands

The DVM commands are only available when the optional DVM software is installed.

| :DVM:STATE | 230 |
|-------------|-----|
| :DVM:SOURce | 230 |
| :DVM:MODe | 231 |
| :DVM:VALue | 231 |

Set → Query)

:DVM:STATE

| Description | Sets or queries the DVM state to on or off. | | |
|------------------|---|--------------------|--|
| Syntax | :DVM:STATE {OFF ON ? } | | |
| Related | :DVM:SOURce | | |
| commands | :DVM:MODe | | |
| Parameter/ | OFF | Turns the DVM off. | |
| Return parameter | ON | Turns the DVM on. | |
| F | .D\/\.c\\T | ON | |

Example :DVM:STATE ON

Turns the DVM state on.

:DVM:SOURce



| Description | Sets or queries the source of the DVM. | | |
|--------------------------------|--|-----------------|--|
| Syntax | :DVM:SOURce {CH1 CH2 CH3 CH4 ?} | | |
| Related | :DVM:STATE | | |
| commands | :DVM:MODe | | |
| Parameter/ Return parameter | CH1~CH4 | Channel 1 to 4. | |
| | | · | |

Example :DVM:SOURce CH1

Sets the DVM source to channel 1.



| :DVM:MODe | | Set → Query | |
|------------------|---|-----------------------------------|--|
| Description | Sets or queries the DVM mode. | | |
| Syntax | :DVM:MODe {ACRMS DC DCRMS DUTY FREQuency ?} | | |
| Related | :DVM:SOUR | ce | |
| commands | :DVM:STATE | | |
| Parameter/ | ACRMS | Sets the mode to AC RMS | |
| Return parameter | DC | Sets the mode to DC | |
| | DCRMS | Sets the mode to DC RMS | |
| | DUTY | Sets the mode to AC Duty | |
| | FREQuency | Sets the mode to AC frequency | |
| Example | :DVM:MODe | DUTY | |
| | Sets the DVI | M mode to DUTY. | |
| :DVM:VALue | | → Query | |
| Description | Returns the measurement value of the selected mode. | | |
| Syntax | :DVM:VALue | ? | |
| Related | :DVM:SOURce | | |
| commands | :DVM:STATE | | |
| | :DVD:MODe | | |
| Return parameter | Returns the | measurement value as <nr3>.</nr3> | |
| Example | :DVM:VALue | ? | |
| | >8.410E-04 | | |
| | measurement. | | |



Go_NoGo Commands

The GoNoGo APP must first be launched (or use the command, ":GONogo:SCRipt") before any of the Go_NoGo or Template commands can be used.

| :GONogo:CLEar | 232 |
|----------------------------|-----|
| :GONogo:EXECute | 232 |
| :GONogo:FUNCtion | |
| :GONogo:NGCount | |
| :GONogo:NGDefine | 233 |
| :GONogo:SOURce | 234 |
| :GONogo:VIOLation | |
| :GONogo:SCRipt | |
| :TEMPlate:MODe | |
| :TEMPlate:MAXimum | 235 |
| :TEMPlate:MINimum | 235 |
| :TEMPlate:POSition:MAXimum | 236 |
| :TEMPlate:POSition:MINimum | 236 |
| :TEMPlate:SAVe:MAXimum | 236 |
| :TEMPlate:SAVe:MINimum | 237 |
| :TEMPlate:TOLerance | 237 |
| :TEMPlate:SAVe:AUTo | 237 |
| | |

:GONogo:CLEar Description Clears the Go/NoGo counter. Syntax :GONogo:CLEar :GONogo:EXECute Description Enables or disables the Go/NoGo function or queries its state. Syntax :GONogo:EXECute {OFF|ON|?}



| Parameter/ | OFF | Disabled | |
|--------------------------------|---|---|--|
| Return Parameter | ON | Enabled | |
| Example | :GONogo:EXECute OFF | | |
| | Turns Go/N | IoGo off. | |
| | | | |
| :GONogo:FUN | Ction | Set → | |
| Description | Initializes the Go/NoGo APP. This must be run after the Go/NoGo APP has been started. | | |
| Syntax | :GONogo:FU | NCtion | |
| | | | |
| :GONogo:NGC | Count | — Query | |
| Description | Returns the | Go/NoGo counter. | |
| Syntax | :GONogo:NGCount{?} | | |
| Return parameter | Returns a string in the following format "number of violations, total tests" | | |
| Example | :GONogo:NGCount? | | |
| | > 3,25 | | |
| | Indicates that 3 violations occurred over 25 tests. | | |
| | Set → | | |
| :GONogo:NGE | Define | → Query | |
| Description | Sets the Go/ | NoGo "When" conditions. | |
| Syntax | :GONogo:NC | GDefine {EXITs ENTers ?} | |
| Parameter/ Return Parameter | EXITs | Sets the NoGo condition to when the input signal exceeds the limit boundary. | |
| | ENTers | Sets the NoGo condition to when the input signal stays within the limit boundary. | |
| Example | :GONogo:NGDefine EXITs | | |
| | Sets the Go/ | o/NoGo condition to EXITs. | |



Set)→ :GONogo:SOURce (Query Sets the source for the Go/NoGo signal. Description :GONogo:SOURce {CH1|CH2|CH3|CH4|?} Syntax CH1~CH4 Parameter/ Return Parameter Example :GONogo:SOURce CH1 Sets the source to CH1. Set :GONogo:VIOLation Query Description Sets or returns actions for the Go/NoGo violations. :GONogo:VIOLation {STOP | CONTinue | ?} Syntax The waveform will be frozen. Parameter/ **STOP** Return Parameter **CONTinue** Ignore the violation. Example :GONogo:VIOLation STOP Sets violation action to STOP. :GONogo:SCRipt Set)→ Activates/Deactivates the Go/NoGo APP or Description queries its state. :GONogo:SCRipt {OFF | ON | ?} Syntax Parameter/ ON Turns Go/NoGo APP on. Return Parameter Turns the Go/NoGo APP off. **OFF** :GONogo:SCRipt? Example >ON

The Go/NoGo script is on.



| :TEMPlate:MO | De | Set → Query | |
|------------------|--|---|--|
| Description | Sets or returns the Go/NoGo template mode. | | |
| Syntax | :TEMPlate:M0 | ODe {MAXimum MINimum AUTO ?} | |
| Parameter/ | MAXimum | Maximum template | |
| Return Parameter | MINimum | Minimum template | |
| | AUTO | Auto template | |
| Example | :TEMPlate:M0 | ODe AUTO | |
| | Sets the temp | plate mode to AUTO. | |
| :TEMPlate:MAX | Kimum | Set → Query | |
| Description | | ueries which waveform memory ~W20) is set to the maximum | |
| Syntax | :TEMPlate:MA | AXimum {REF1 W1~W20 ?} | |
| Parameter/ | REF1 Re | eference one | |
| Return Parameter | W1~W20 W | aveform memory 1 to 20 | |
| Example | :TEMPlate:MAXimum REF1 | | |
| | Saves the ma | res the maximum template to REF1. | |
| | | | |
| Description | | ueries which waveform memory ~W20) is set to the minimum | |
| Syntax | :TEMPlate:MI | Nimum {REF2 W1~W20 ?} | |
| Parameter/ | REF2 Re | eference one | |
| Return Parameter | W1~W20 Waveform memory 1 to 20 | | |
| Example | :TEMPlate:MI | Nimum REF2 | |
| | Saves the mi | nimum template to REF2. | |



| :TEMPlate:POS | ition:MAX | (imum | Set ———————————————————————————————————— |
|---|---|---|--|
| Description | Sets or que template. | eries the position of t | he maximum |
| Syntax | :TEMPlate: | POSition:MAXimum { | <nr2> ?}</nr2> |
| Parameter | <nr2></nr2> | Desired template pos divisions) | ition (-12.0 ~ +12.0 |
| Return parameter | Returns the " <nr2>Div</nr2> | e position in the follow," | ving format: |
| Example | :TEMPlate: | POSition:MAXimum 3 | 3.00 |
| | Sets the madivisions. | aximum template po | sition to 3.00 |
| | | | Set → |
| :TEMPlate:POS | ition:MIN | imum | → Query |
| Description | Sets or alle | eries the position of t | ho minimum |
| | template. | eries the position of t | ne minimum |
| Syntax | template. | POSition:MINimum { | |
| | template. | | <nr2> ?}</nr2> |
| Syntax | template. :TEMPlate: <nr2></nr2> | POSition:MINimum { Desired template pos divisions) e position in the follow | <nr2> ?} ition (-12.0 ~ +12.0</nr2> |
| Syntax Parameter | template. :TEMPlate: <nr2> Returns the "<nr2>Div</nr2></nr2> | POSition:MINimum { Desired template pos divisions) e position in the follow | <nr2> ?} iition (-12.0 ~ +12.0 ving format:</nr2> |
| Syntax Parameter Return parameter | template. :TEMPlate: <nr2> Returns the "<nr2>Div :TEMPlate:</nr2></nr2> | POSition:MINimum { Desired template pos divisions) e position in the follow | <nr2> ?} sition (-12.0 ~ +12.0 ving format:</nr2> |
| Syntax Parameter Return parameter | template. :TEMPlate: <nr2> Returns the "<nr2>Div :TEMPlate: Sets the midivisions.</nr2></nr2> | POSition:MINimum { Desired template positions) position in the follow POSition:MINimum 3 inimum template po | <nr2> ?} sition (-12.0 ~ +12.0 ving format:</nr2> |
| Syntax Parameter Return parameter Example | template. :TEMPlate: <nr2> Returns the "<nr2>Div :TEMPlate: Sets the midivisions. e:MAXimu</nr2></nr2> | POSition:MINimum { Desired template positions) position in the follow POSition:MINimum 3 inimum template po | <nr2> ?} sition (-12.0 ~ +12.0 sing format: .00 sition to 3.00</nr2> |



| :TEMPlate:SAVe:MINimum | | | <u>Set</u> → |
|--------------------------------|--------------------------------------|-----------------------------------|------------------|
| Description | Saves the maximum template. | | |
| Syntax | :TEMPlate:SAVe:MINimum | | |
| :TEMPlate:TOL | erance | | Set → Query |
| Description | Sets or que | eries the tolerance as | a percentage. |
| Syntax | :TEMPlate:TOLerance { <nr2> ?}</nr2> | | |
| Parameter/ Return Parameter | <nr2></nr2> | The auto tolerance ra | nge (0.4% ~ 40%) |
| Example | :TEMPlate:TOLerance 10 | | |
| | Sets the tolerance to 10%. | | |
| :TEMPlate:SAVe:AUTo Set → | | | |
| Description | | AUTO template (max templates). | simum and |
| Syntax | :TEMPlate: | SAVe:AUTo | |



Data Logging Commands

| :DATALOG:STATE | 238 |
|-------------------|-----|
| :DATALOG:SOURce | 238 |
| :DATALOG:SAVe | 239 |
| :DATALOG:INTerval | 239 |
| :DATALOG:DURation | 240 |

Set → Query

:DATALOG:STATE

| Description | Sets or querio | es the state of the data logging app. | |
|------------------|---------------------------|---------------------------------------|--|
| Syntax | :DATALOG:STATE {OFF ON ?} | | |
| Related | :DATALOG:SO | DURce | |
| commands | :DATALOG:SAVe | | |
| | :DATALOG:INTerval | | |
| | :DATALOG:DURation | | |
| Parameter/ | OFF | Turns the data logging off. | |
| Return parameter | ON | Turns the data logging on. | |
| Example | :DATALOG:ST | TATE ON | |

Turns the data logging app on.

Set → Query

:DATALOG:SOURce

| Description | Sets or querie | es the data logging source channel. | |
|------------------|---------------------------------|-------------------------------------|--|
| Syntax | :DATALOG:SOURce {CH1~CH4 all ?} | | |
| Related | :DATALOG:ST | ATE | |
| commands | :DATALOG:SAVe | | |
| | :DATALOG:INTerval | | |
| | :DATALOG:DI | JRation | |
| Parameter/Return | CH1 ~CH4 | Channel 1, 2, 3 or 4 | |



| parameter | all | All displayed chanr | nels. |
|------------------|---|---|----------------|
| Example | :DATALOG:SOURce CH1 Sets the source to CH1. | | |
| | | | |
| | | | Set → |
| :DATALOG:SA | √e | | → Query |
| Description | Sets or queri waveform. | es the save format as | s image or |
| Syntax | :DATALOG:S/ | AVe {IMAGe WAVEfor | m ?} |
| Related | :DATALOG:S | ГАТЕ | |
| commands | :DATALOG:S0 | OURce | |
| | :DATALOG:IN | lTerval | |
| | :DATALOG:DURation | | |
| Parameter/Return | IMAGe | Save as images. | |
| parameter | WAVEform | Save as waveforms. | |
| Example | :DATALOG:S/ | AVe WAVEform | |
| | Sets the save format to waveform. | | |
| | | | Set → |
| :DATALOG:INT | Γerval | | → Query |
| Description | seconds. The dependent o | es the recording inte e interval times that on the settings of the SOURCE and DATA | can be set are |
| Syntax | :DATALOG:IN | Terval { <nr1> ?}</nr1> | |
| Related | :DATALOG:STATE | | |
| commands | :DATALOG:SOURce | | |
| | :DATALOG:SAVe | | |
| | :DATALOG:DURation | | |
| | | | |



| Parameter/Return parameter | <nr1></nr1> | Sets returns the interval time in discrete seconds: | |
|----------------------------|------------------------------------|--|--|
| | | Interval time for DATALOG:SOURCE = All or DATALOG:SAVE=IMAGE: 5, 10, 15, 20, 25, 30, 35, 60, 120 | |
| | | Interval time for DATALOG:SOURCE = CH1~CH4: 2, 3, 4, 5, 10, 20, 30, 60, 120 | |
| Example | :DATALOG:IN | NTerval 5 | |
| | Sets the reco | rding interval to 5 seconds. | |
| | | <u>Set</u> → | |
| :DATALOG:DU | Ration | → Query | |
| Description | Sets or queri minutes. | es the recording duration time in | |
| Syntax | :DATALOG:DURation { <nr1> ?}</nr1> | | |
| Related | :DATALOG:STATE | | |
| | :DATALOG:SOURce | | |
| commands | :DATALOG:S0 | OURce | |
| commands | :DATALOG:SA | | |
| commands | | AVe | |
| Parameter/Return parameter | :DATALOG:SA | AVe | |
| Parameter/Return | :DATALOG:SA | AVe NTerval Sets returns the duration time in | |
| Parameter/Return | :DATALOG:SA | AVe NTerval Sets returns the duration time in discrete minutes: 5, 10, 15, 20, 25, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330, 360, 390, 420, 450, 480, 510, 540, 570, 600, 1200, 1800, 2400, 3000, 3600, 4200, 4800, 5400, 6000 | |



Syntax

Remote Disk Commands

| Kerriote Dis | sk Collin | iaiius | |
|--------------------------------|--|---|---------------------------------|
| | :REMOTEDIS :REMOTEDIS :REMOTEDIS :REMOTEDIS | sk:IPADDresssk:PATHNamesk:USERNamesk:PASSWordsk:MOUNTsk:AUTOMountsk:AUTOMount | 241 241 242 242 243 |
| :REMOTEDisk: | IPADDress | | Set → Query |
| Description | Sets or retur | ns the IP address of | remote disk. |
| Syntax | :REMOTEDis | sk:IPADDress { <string< td=""><td>:> ;}</td></string<> | :> ;} |
| Parameter/ Return parameter | <string></string> | IP address enclosed Eg., 172.16.20.255 | l in double quotes. |
| Example | :REMOTEDisk:IPADDress "172.16.20.255" | | |
| | Sets the remote disk IP address as 172.16.20.255. | | |
| :REMOTEDisk: | PATHName | e | Set → Query |
| Description | Sets or retur | ns the file path of the | e remote disk. |
| Syntax | | sk:PATHName { <string< td=""><td></td></string<> | |
| Parameter/ Return parameter | <string></string> | File path in enclosed eg., "remote_disk" | d in double quotes |
| Example | :REMOTEDis | sk:PATHName "remot | e_disk" |
| | Sets the file | path to c:/remote_di | isk. |
| | | | Set → |
| :REMOTEDisk: | USERName | 2 | → Query |
| Description | Sets or queri remote disk | ies the account usern | name for the |

 $: REMOTED is k: USERName \ \{<\! string\! > |\ ?\ \}$



| · | | | |
|----------------------------|---|--|--|
| Parameter/Return parameter | <string></string> | User name enclosed in double quotes eg., "User_Name". | |
| Example | :REMOTEDisk:USERName "User_Name" | | |
| | Sets the account name as User_Name. | | |
| | | Set → | |
| : REMOTED is k: | PASSWord | Query | |
| Description | Sets or queri remote disk. | es the account password for the | |
| Syntax | :REMOTEDis | k:PASSWord { <string> ? }</string> | |
| Parameter/Return parameter | <string></string> | Username password enclosed in double quotes eg., "Password". | |
| Example | :REMOTEDis | k:PASSWord "Password" | |
| | Sets the account password as Password. | | |
| | | Set → | |
| : REMOTED is k: | MOUNT | → Query | |
| Description | Turns remot | e disk on/off or queries its state. | |
| Syntax | :REMOTEDisk:MOUNT { OFF ON ? } | | |
| Parameter/Return | OFF | Unmount remote disk | |
| parameter | ON Mount remote disk | | |
| Example | :REMOTEDis | k:IPADDress "172.16.5.154" | |
| | :REMOTEDisk:PATHName "remote_disk" :REMOTEDisk:USERName "guest" | | |
| | | | |
| | :REMOTEDisk:PASSWord "password" :REMOTEDisk:MOUNT ON | | |
| | | | |
| | Sets the remote disk parameters and mounts the remote disk. | | |



| :REMOTEDisk: | AUTOMoun | t Set → Query | |
|----------------------------|---|--|--|
| Description | Turns automount on/off or queries its state. The remote disk must be configured beforehand. | | |
| Syntax | :REMOTEDisk:AUTOMount { OFF ON ? } | | |
| Parameter/Return parameter | | Don't mount the remote disk at start up. | |
| | | Automatically mount the remote disk on start up. | |
| Example | :REMOTEDisk:AUTOMount ON | | |
| | Turns the automount function on. | | |



Error messages

| Description | The following error messages may be returned from the :SYSTem:ERRor? query. For details see page 157. | | |
|---------------|---|--|--|
| List of error | Error number, "Error Description" | | |
| messages | +0, "No error." | | |
| | -100, "Command error" | | |
| | -101, "Invalid character" | | |
| | -102, "Syntax error" | | |
| | -103, "Invalid separator" | | |
| | -104, "Data type error" | | |
| | -105, "GET not allowed" | | |
| | -108, "Parameter not allowed" | | |
| | -109, "Missing parameter" | | |
| | -110, "Command header error" | | |
| | -111, "Header separator error" | | |
| | -112, "Program mnemonic too long" | | |
| | -113, "Undefined header" | | |
| | -114, "Header suffix out of range" | | |
| | -115, "Unexpected number of parameters" | | |
| | -120, "Numeric data error" | | |
| | -121, "Invalid character in number" | | |
| | -123, "Exponent too large" | | |
| | -124, "Too many digits" | | |
| | -128, "Numeric data not allowed" | | |
| | -130, "Suffix error" | | |
| | -131, "Invalid suffix" | | |
| | -134, "Suffix too long" | | |
| | -138, "Suffix not allowed" | | |

- -140, "Character data error"
- -141, "Invalid character data"
- -144, "Character data too long"
- -148, "Character data not allowed"
- -150, "String data error"
- -151, "Invalid string data"
- -158, "String data not allowed"
- -160, "Block data error"
- -161, "Invalid block data"
- -168, "Block data not allowed"
- -170, "Expression error"
- -171, "Invalid expression"
- -178, "Expression data not allowed"
- -180, "Macro error"
- -181, "Invalid outside macro definition"
- -183, "Invalid inside macro definition"
- -184, "Macro parameter error"
- -200, "Execution error"
- -201, "Invalid while in local"
- -202, "Settings lost due to rtl"
- -203, "Command protected"
- -210, "Trigger error"
- -211, "Trigger ignored"
- -212, "Arm ignored"
- -213, "Init ignored"
- -214, "Trigger deadlock"
- -215, "Arm deadlock"
- -220, "Parameter error"
- -221, "Settings conflict"
- -222, "Data out of range"
- -223, "Too much data"
- -224, "Illegal parameter value"
- -225, "Out of memory"
- -226, "Lists not same length"
- -230, "Data corrupt or stale"
- -231, "Data questionable"
- -232, "Invalid format"
- -233, "Invalid version"
- -240, "Hardware error"

- -241, "Hardware missing"
- -250, "Mass storage error"
- -251, "Missing mass storage"
- -252, "Missing media"
- -253, "Corrupt media"
- -254, "Media full"
- -255, "Directory full"
- -256, "File name not found"
- -257, "File name error"
- -258, "Media protected"
- -260, "Expression error"
- -261, "Math error in expression"
- -270, "Macro error"
- -271, "Macro syntax error"
- -272, "Macro execution error"
- -273, "Illegal macro label"
- -274, "Macro parameter error"
- -275, "Macro definition too long"
- -276, "Macro recursion error"
- -277, "Macro redefinition not allowed"
- -278, "Macro header not found"
- -280, "Program error"
- -281, "Cannot create program"
- -282, "Illegal program name"
- -283, "Illegal variable name"
- -284, "Program currently running"
- -285, "Program syntax error"
- -286, "Program runtime error"
- -290, "Memory use error"
- -291, "Out of memory"
- -292, "Referenced name does not exist"
- -293, "Referenced name already exists"
- -294, "Incompatible type"
- -300, "Device-specific error"
- -310, "System error"
- -311, "Memory error"
- -312, "PUD memory lost"
- -313, "Calibration memory lost"
- -314, "Save/recall memory lost"

- -315, "Configuration memory lost"
- -320, "Storage fault"
- -321, "Out of memory"
- -330, "Self-test failed"
- -340, "Calibration failed"
- -350, "Queue overflow"
- -360, "Communication error"
- -361, "Parity error in program message"
- -362, "Framing error in program message"
- -363, "Input buffer overrun"
- -365, "Time out error"
- -400, "Query error"
- -410, "Query INTERRUPTED"
- -420, "Query UNTERMINATED"
- -430, "Query DEADLOCKED"
- -440, "Query UNTERMINATED after indefinite response"



NDEX

| ACQuire | INPut | 16 |
|----------------------------|------------------|----|
| FILTer | LABel | |
| TRACking44 | DISPlay | |
| ACQuire | LIN | |
| | BITRate | 17 |
| AVERage38 | IDFORmat | 17 |
| FILTer | POLARity | 17 |
| FREQuency42 | SAMPLEpoint | 17 |
| FILTer42 | SOURce | |
| SOURce41 | STANDard | 17 |
| MEMory39 | SPI | |
| MODe39 | BITORder | 17 |
| ACQuire | MISO | |
| = | SOURce | 17 |
| STATe44 | MOSI | |
| ACQuire | SOURce | 17 |
| | SCLK POLARity | 17 |
| INTERpolation44 | SOURce | |
| ACQuire | SS SS | 17 |
| RECOrdlength45 | POLARity | 17 |
| O | SOURce | |
| ACQuire | WORDSize | |
| HEADer45 | STATE | 16 |
| | TYPe | 16 |
| AUTORSET | UART | |
| MODe46 | BITRate | 16 |
| AUTOSet 46 | DATABits | 16 |
| AU103et40 | EOFPAcket | 17 |
| BUS1166 | PACKEt | 17 |
| | PARIty | 16 |
| CAN | RX | |
| BITRate | SOURce | 17 |
| PROBe175 SAMPLEpoint176 | TX | |
| SOURce175 | SOURce | 17 |
| DISplay | CHANnel | |
| FORMAt174 | BWLimit | 4 |
| I2C | COUPling | |
| ADDRess | DESKew | |
| RWINClude167 | | |
| SCLK | DISPlay | |
| SOURce168 | EXPand | |
| SDA | IMPedance | |
| | INVert | 5 |

GWINSTEK

| LABel215 DISPlay216 | DATe | 164 |
|------------------------|---------------|-----|
| POSition50 | DISplay | |
| PROBe | | |
| TYPE51 | INTENSITY | |
| PROBe | BACKLight | |
| RATio51 | AUTODim | |
| SCALe51 | TIMe | |
| | DISPlay | |
| CLS34 | GRATicule | 75 |
| CURSor | INTensity | |
| | BACKLight | 74 |
| DDT65 | BACKLight | |
| H1Position66 | AUTODim | |
| H2Position66 | ENAble | 74 |
| HDELta67 | GRATicule | 73 |
| HUNI64 | WAVEform | |
| HUSE64 | OUTPut | 76 |
| MODe63 | PERSistence | 75 |
| SOURce63 | WAVEform | 76 |
| V1Position67 | DVM | |
| VDELta68 | | |
| VUNI65 | MODe | |
| VUSE65 | SOURce | |
| XY | STATE | 230 |
| POLar | VALue | 231 |
| RADIUS | Error list | 244 |
| DELta70 | 21101 1101 | |
| POSition69 | Ethernet | |
| THETA | interface | 7 |
| DELta70 | | |
| POSition70 | ETHERnet | |
| PRODuct DELta71 | DHCP | 163 |
| POSition71 | EORC- | 117 |
| RATio | FORCe | 116 |
| DELta72 | GONogo | |
| RATio | · · | 200 |
| POSition71 | CLEar | |
| RECTangular | EXECute | |
| X | FUNCtion | |
| DELta68 | NGCount | |
| POSition68 | NGDefine | |
| Y DELta69 | SCRipt | |
| POSition69 | SOURce | |
| | VIOLation | 234 |
| DATALOG | HARDcopy | |
| DURation240 | | |
| INTerval239 | ASSIGN | |
| SAVe239 | MODe | |
| SOURce238 | PRINTINKSaver | |
| STATE238 | SAVEFORMat | |
| J11111230 | SAVEINKSaver | 78 |

GWINSTEK

| START77 | GATing | 81 |
|--------------------|-------------|-----|
| HEADer45 | HIGH | 93 |
| 11EADel 45 | LFFDelay | |
| IDN?32 | LFRDelay | |
| | LOW | |
| Interface5 | LRFDelay | |
| T TOTAL | LRRDelay | |
| LISTer | MAX | |
| DATA175 | MEAN | |
| I DNI2 | | |
| LRN? 32 | METHod | |
| MARK180 | MIN | |
| | NEDGE | |
| CREATE | NPULSE | |
| DELEte181 | NWIDth | |
| MATH | PDUTy | |
| A DV 1 | PEDGE | 91 |
| ADVanced | PERiod | 87 |
| OPERator | PHAse | 103 |
| SCALe60 | PK2PK | 96 |
| DISP53 | PPULSE | 90 |
| | PWIDth | |
| DUAL OPERator54 | RISe | |
| POSition55 | RMS | |
| SCALe55 | ROVShoot | |
| SOURce54 | RPReshoot | |
| FFT | SOURce | |
| HORizontal | | 01 |
| POSition58 | MEASUrement | |
| SCALe58 | MEAS | |
| MAG56 | MAXimum | 108 |
| POSition57 | MEAN | |
| SCALe57 | MINImum | 109 |
| SOURce56 | SOURCE | 105 |
| WINDow56 | STATE | 106 |
| TYPE54 | STDdev | |
| MATHVAR 59 | TYPe | |
| | VALue | 107 |
| MEASure | REFLevel | |
| AMPlitude92 | PERCent | |
| AREa97 | HIGH | |
| CARea98 | LOW MID | |
| CMEan93 | MID2 | |
| | STATIstics | |
| CRMS97 | MODe | |
| FALL84 | WEIghting | |
| FFFDelay100 | 0 0 | |
| FFRDelay100 | RCL | 33 |
| FOVShoot84 | RECAIl | |
| FPReshoot85 | | |
| FREQuency85 | SETUp | |
| FRFDelay99 | WAVEform | |
| EDDD alary 00 | | |



| REF | DIRection 205 |
|--------------------------|------------------|
| KET | MODe 204 |
| DISPlay112 | VALue 205 |
| LABel216 | I2C |
| DISPlay217 | ADDRess |
| OFFSet113 | DIRection 195 |
| SCALe114 | MODe 193 |
| TIMebase | TYPe193 |
| POSition112 | VALue 194 |
| SCALe113 | CONDition 192 |
| SCALE113 | DATa |
| Remote control | SIZe |
| interface configuration5 | VALue 196 LIN |
| REMOTEDisk | CONDition 208 |
| | DATa |
| AUTOMount243 | QUALifier 209 |
| IPADDress241 | SIZe210 |
| MOUNT242 | VALue 210 |
| PASSWord242 | ERRTYPE 21: |
| PATHName241 | IDentifier |
| USERName21 | VALue 212 |
| USEKName241 | SPI |
| RST34 | CONDition 200 |
| | DATa |
| RUN115 | MISO 201 |
| | MOSI 202 |
| SAV33 | SIZe200 |
| CATA | UART |
| SAVe | CONDition 196 |
| IMAGe159 | RX |
| FILEFormat159 | DATa197, 198 |
| INKSaver | TX |
| SETUp160 | DATa 199 |
| * | TYPe 192 |
| WAVEform161 | EDGE |
| FILEFormat162 | SLOP |
| SEARCH | HLEVel186 |
| CODY 199 | LEVel 180 |
| COPY183 | LLEVel187 |
| FFTPeak | PULSe |
| METHod212 | TIMe 189 |
| MPEak213 | WHEn189 |
| SINFo214 | PULSEWidth |
| STATE184 | POLarity |
| TOTAL184 | RISEFall |
| TRIGger | SLOP |
| BUS | TIMe 19 |
| B1 | WHEn |
| CAN | RUNT |
| CONDition203 | POLarity |
| DATa | TIMe |
| QUALifier206 | WHEn |
| SIZe207 | SOURce |
| VALue207 | TYPe |
| | |
| FRAMEtype204 IDentifier | SEGMents |



| CURRent222 | MODe | 118 |
|------------------|-------------------|---------|
| DISPALL223 | POSition | 117 |
| MEASure | SCALe | 117 |
| MODe223 | WINDow | |
| PLOT | POSition | . 118 |
| DIVide224 | | . 119 |
| RESults225 | 1121/ '004 | |
| SELect225 | · · | |
| SOURce224 | 112101140 | 135 |
| TABle LIST227 | BUS | |
| SAVe | , D1 | |
| SELect | CAIN | 4.45 |
| SOURce226 | CONDITION | 147 |
| SAVe228 | DATa | 150 |
| SELect | QUALifier SIZe | |
| END229 | | |
| STARt228 | V 1154C | |
| SOURce228 | THE HALL PER | 140 |
| STATE221 | | 150 |
| TIMe223 | | |
| TOTalnum222 | | |
| | I2C | |
| SET | ADDRess | |
| LABel219 | DIRection | 140 |
| SINGle115 | MODe | |
| 511NG1e 115 | 1110 | |
| Socket server | VALue | |
| | CONDition | 137 |
| function check10 | DATa SIZe | 140 |
| Socket server | VALue | |
| interface9 | | 111 |
| mieriace9 | CONDition | 152 |
| STOP 115 | DATa | |
| ON COTT | QUALifier | 154 |
| SYSTem | SIZe | 154 |
| ERRor157 | VALue | 155 |
| LOCK157 | ERRTYPE | 155 |
| | IDentifier | |
| TEMPlate | VALue | 156 |
| MAXimum235 | SPI | 1.45 |
| MINimum235 | CONDition | 145 |
| MODe235 | DATa MISO | |
| POSition | VALue | 146 |
| MAXimum236 | | 140 |
| MINimum236 | | 147 |
| SAVe | SIZe | |
| AUTo237 | | - |
| MAXimum236 | | 142 |
| MINimum237 | RX | |
| TOLerance237 | | |
| TIMebase | SIZe | |
| Thrievase | VALue | 143 |
| EXPand117 | TX | |



| DATa | TIME | 134 |
|--------------|--------------------------|-----|
| SIZe143 | WHEn | 133 |
| VALue144 | PULSEWidth | |
| THReshold | POLarity | 128 |
| CH137 | RISEFall | |
| TYPe137 | SLOP | 130 |
| COUPle123 | TIMe | 131 |
| DELay | WHEn | 130 |
| EVENt127 | RUNT | |
| LEVel128 | POLarity | 128 |
| SLOP126 | TIMe | |
| TIME127 | WHEn | 129 |
| TYPE127 | SOURce | 122 |
| EDGe | STATe | 135 |
| SLOP126 | TIMEOut | |
| EXTERnal | TIMER | 134 |
| PRObe | WHEn | |
| RATio136 | TYPE | |
| TYPe136 | VIDeo | 122 |
| FREQuency122 | FIELd | 132 |
| HLEVel125 | LINe | |
| HOLDoff124 | POLarity | |
| LEVel124 | TYPe | |
| LLEVel125 | | |
| MODe124 | USB | |
| NREJ | function check | 6 |
| | remote control interface | 5 |
| PULSe | | |