

# Multi-Function Programmable Oscilloscope

MPO-2000 series

---

## PROGRAMMING MANUAL



ISO-9001 CERTIFIED MANUFACTURER

**GW INSTEK**

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

<b>INTERFACE OVERVIEW .....</b>	<b>5</b>
Interface Configuration .....	5
Web Server .....	10
<b>COMMAND OVERVIEW .....</b>	<b>14</b>
Command Syntax .....	14
List of Commands in Functional Order ....	16
<b>COMMAND DETAILS .....</b>	<b>35</b>
Common Commands .....	37
Mask Commands .....	42
DMM Commands.....	56
Acquisition Commands.....	61
Autoscale Commands .....	69
Vertical Commands .....	70
Math Commands .....	75
Cursor Commands .....	83
Display Commands .....	95
Hardcopy Commands .....	100
Measure Commands .....	103
Measurement Commands .....	128
Reference Commands .....	137
Run Command .....	140
Stop Command .....	141
Single Command.....	142
Force Command.....	143
Timebase Commands.....	144
Trigger Commands.....	147
System Commands .....	182
Save/Recall Commands .....	183
Ethernet Command .....	188

Time Command .....	191
Bus Decode Commands .....	192
Mark Commands.....	205
Search Commands .....	207
Label Commands .....	239
Segment Commands .....	245
DVM Commands.....	255
Go_NoGo Commands .....	257
AWG Commands .....	263
Data Logging Commands .....	282
Remote Disk Commands.....	285
Spectrum Analyzer Commands .....	288
USB Delay Command .....	304
FRA Commands .....	305
Power Supply Commands.....	312
Python Commands.....	314
<b>APPENDIX.....</b>	<b>320</b>
Error messages .....	320

# INTERFACE OVERVIEW

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

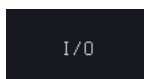
## Interface Configuration

### Configure USB Interface

USB Configuration	PC side connector	Type A, host
	MPO-2000 side connector	Type B, device
	Speed	1.1/2.0
	USB Class	USB-TMC (USB Test and Measurement Class)

### Panel Operation

1. Press the Utility key.
2. Press *I/O* from the bottom menu.
3. Rotate the *VARIABLE* knob to select the *USB Device Port* function.



4. Select *Computer* from the side menu.



5. This oscilloscope is a USB-TMC device. Please install the National Instruments NI-VISA library which can download from the National Instruments web site. Newer versions are likely, and should be compatible with this instrumentation. Download the latest version available for the operating system being used by the controlling computer.

## Configure the Ethernet Interface

Ethernet Configuration	MAC Address	Domain Name
	Instrument Name	DNS IP Address
	User Password	Gateway IP Address
	Instrument IP Address	Subnet Mask
		HTTP Port 80 (fixed)

Background	The Ethernet interface is used for remote control using a socket server connection. For details, please see the Socket Server section on page 8.
------------	--

1. Connect the Ethernet cable to the LAN port on the rear panel.



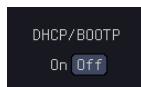
2. Press the *Utility* key.
3. Press *I/O* from the bottom menu.



4. Press *Ethernet* from the side menu.



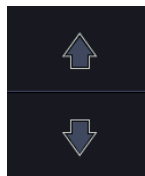
- Set *DHCP/BOOTP* to *On* or *Off* from the side menu.



IP addresses will automatically be assigned with DHCP/BOOTP set to on. For Static IP Addresses, DHCP/BOOTP should be set to off.



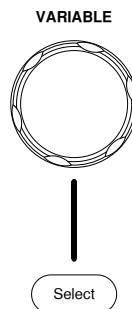
- Use the *Up* and *Down* arrows on the side menu to navigate to each Ethernet configuration item.



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.

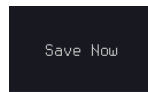
- Use the *Variable* knob to highlight a character and use the *Select* key to choose a character.



Press *Backspace* to delete a character.



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



## Configure Socket Server

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

Configure Socket Server

1. Configure the IP address for the MPO-2000.
2. Press the *Utility* key.

Page 6



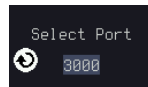
3. Press *I/O* from the bottom menu.



4. Rotate the *VARIABLE* knob to select the *Socket Server* function.



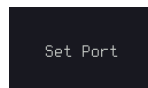
5. Press *Select Port* and choose the port number with the Variable knob.



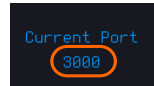
*Please notice that the default socket port for MPO-2000 series is "5025".*

Range 1024~65535

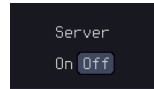
6. Press *Set Port* to confirm the port number.



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



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



# Web Server

## Web Server Overview

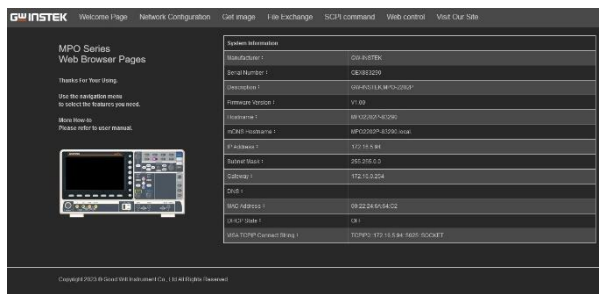
### Background

The MPO-2000 series has an inbuilt web server that can be used to:

- view the system information (Welcome Page)
- set/view the network configuration settings (Network Configuration)
- remotely view the current display image on the unit (Get Display Image)
- execute SCPI command
- send the internal profile of oscilloscope to PC side or receive profile
- Web control function: control oscilloscope remotely from browser and display waveform in real-time

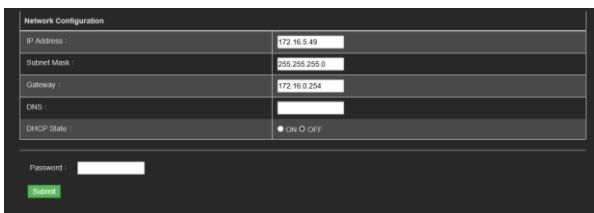
### System Information:

- Manufacturer
- Serial Number
- Firmware version
- Hostname
- Domain name
- IP Address
- Subset Mask
- DNS
- MAC Address
- DHCP State



Network Configuration

- Hostname
- Domain name
- IP Address
- Subnet mask
- Gateway
- DNS
- DHCP State



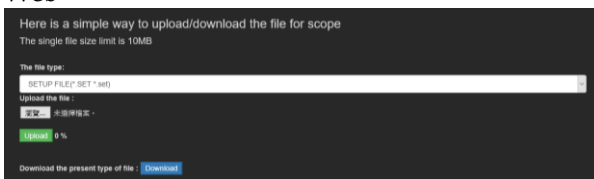
Get Display Image

- Current display image



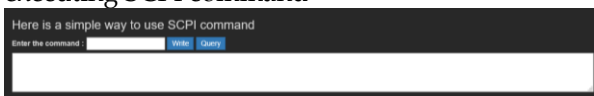
File Exchange

Upload or download profile (\*.set) to oscilloscope Web



SCPI command

Control oscilloscope remotely from browser via executing SCPI command



Web control

Control oscilloscope remotely from browser via graphical user interface (GUI) to display real-time waveform



Panel Operation

1. Configure the Ethernet interface. Page 6
2. Enter the IP address of the MPO-2000 unit into the address bar of a web browser.

For example

<http://172.16.20.255>

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



4. Rotate the *VARIABLE* knob to select the *Web Server* function.



5. Press the *Connect* button in the side menu to connect to internet.



6. The "ONLINE" will be shown for web server when internet connection is established.



7. The MPO-2000 web browser welcome page appears.

GW INSTEK    Welcome Page    Network Configuration    Call usage    File Exchange    SCPI command    Web control    Visit Our Site


---

**MPO Series  
Web Browser Pages**

Thanks for Your Using.

Use the navigation menu to select the features you need.

Basic Setup  
Please refer to user manual.



System Information	
Manufacturer 1	GW INSTEK
Serial Number 1	08381820
Model Name 1	GW INSTEK MPO2000P
Firmware Version 1	1.00
Hardware 1	MS10200-M3000
PCMC Memory 1	MP0000P-A1000 user
IP Address 1	192.16.8.81
Default Host 1	255.255.0.0
Gateway 1	192.16.8.204
DNS 1	
MAC Address 1	08-22-24-64-64-02
LAN Card 1	CM1
LAN TCP/IP Control Setting 1	TCP/IP: 192.16.8.81 8081 SOCKET

Copyright © 2009 by GW INSTEK Inc. All Rights Reserved.

# C **COMMAND OVERVIEW**

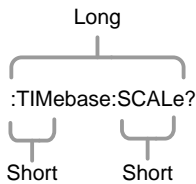
The Command overview chapter lists all MPO-2000 commands in functional order as well as alphabetical order. The command syntax section shows you the basic syntax rules you have to apply when using commands.

## Command Syntax

---

- |                     |   |
|---------------------|---|
| Compatible standard | <ul style="list-style-type: none"> <li>• USB CDC_ACM compatible</li> <li>• SCPI, 1994 (partially compatible)</li> </ul> |
|---------------------|---|

**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.



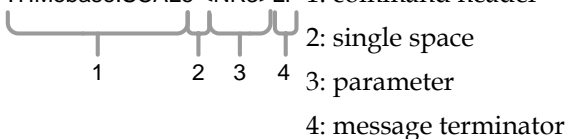
The commands can be written in capitals or lower-case, just so long as the short or long forms are complete. An incomplete command will not be recognized.

Below are examples of correctly written commands.

LONG :TIMEbase:SCALE? :TIMEBASE:SCALE?  
           :timebase:scale?

SHORT :TIM:SCAL? :TIM:SCAL?

Command format :TIMEbase:SCALE <NR3>LF 1: command header



Parameter	Type	Description	Example
	<Boolean>	boolean logic	0, 1
	<NR1>	Integers	0, 1, 2, 3
	<NR2>	floating point	0.1, 3.14, 8.5
	<NR3>	floating point with an exponent	4.5e-1, 8.25e+1
	<NRf>	any of NR1, 2, 3	1, 1.5, 4.5e-1
Message terminator	LF	line feed code	



Commands are non-case sensitive.

## List of Commands in Functional Order

Common commands	*IDN?	37
	*LRN?	37
	*SAV	38
	*RCL	38
	*RST	38
	*CLS	38
	*ESE	38
	*ESR	39
	*OPC	40
	*SRE	40
	*STB	41
<hr/>		
Mask commands	:MASK:STATe	42
	:MASK:SOURce	42
	:MASK:VIOLation	42
	:MASK:VIOLation:SAVE	43
	:MASK:AUTO	43
	:MASK:AUTO:SOURce	43
	:MASK:AUTO:UNITs	44
	:MASK:USER:UNITs	44
	:MASK:USER:AREa<x1>:POINT<x2>	44
	:MASK:USER:AREa<x1>:POINT<x2>:STATe	45
	:MASK:USER:SAVE	45
	:MASK:USER:LOAD	46
	:MASK:USER:CREATE	46
	:MASK:RATIO	46
	:MASK:VIOLation:THREshold	47
	:MASK:STOPAFter	47
	:MASK:STOPAFter:WAVEform	48
	:MASK:STOPAFter:TIME	48
	:MASK:FAILACTion:STOPACQuisition	48
:MASK:FAILACTion:SAVEIMAge	49	

	:MASK:FAILACTion:SAVEWAVEform .....	49
	:MASK:FAILACTion:HARDCopy .....	50
	:MASK:FAILACTion:GONogo .....	50
	:MASK:COMPLETEACTion:GONogo .....	51
	:MASK:COMPLETEACTion:TESTREpeat .....	51
	:MASK:COMPLETEACTion:TESTDelay .....	51
	:MASK:RESults:DISPlay .....	52
	:MASK:RESults:DISPlay:DETail .....	52
	:MASK:RESults:CURRent:STATUS .....	53
	:MASK:RESults:CURRent:WAVEform .....	53
	:MASK:RESults:CURRent:VIOlation .....	53
	:MASK:RESults:CURRent:TIME .....	54
	:MASK:RESults:CURRent:HITSPERSEGment<x> .....	54
	:MASK:RESults:TOTal:WAVEform .....	54
	:MASK:RESults:TOTal:VIOlation .....	54
	:MASK:RESults:TOTal:TESTNUMber .....	55
	:MASK:RESults:TOTal:FTESTNUMber .....	55
	:MASK:RESults:TOTal:TIME .....	55
	:MASK:RESults:TOTal:HITSPERSEGment<x> .....	55
<hr/>		
DMM commands	:DMM .....	56
	:DMM:STATE .....	56
	:DMM:VALue .....	56
	:DMM:HOLD .....	57
	:DMM:MMIN .....	57
	:DMM:MODE .....	57
	:DMM:MODE:RANGe .....	58
	:DMM:TEMPerature:UNITs .....	59
	:DMM:TEMPerature:TYPe .....	59
	:DMM:TEMPerature:SIM .....	60
<hr/>		
Acquisition commands	:ACQuire:AVErAge .....	61
	:ACQuire:MODE .....	61
	:ACQuire<X>:MEMory? .....	62
	:ACQuire:FILTer:SOURce .....	64

	:ACQuire:FILTer.....	64
	:ACQuire:FILTer:FREQuency:UPPER.....	65
	:ACQuire:FILTer:FREQuency:LOWER.....	65
	:ACQuire:FILTer:TYPe.....	66
	:ACQuire:FILTer:TRACking.....	66
	:ACQuire<X>:STATe?.....	66
	:ACQuire:INTERpolation.....	67
	:ACQuire:RECOrdlength.....	67
	:ACQuire:SAMPLerate?.....	67
	:HEADer.....	68
<hr/>		
Autoscale commands	:AUTOSet.....	69
	:AUTORSET:MODE.....	69
<hr/>		
Vertical Scale commands	:CHANnel<X>:BWLimit.....	70
	:CHANnel<X>:COUPling.....	70
	:CHANnel<X>:DESKew.....	71
	:CHANnel<X>:DISPlay.....	71
	:CHANnel<X>:EXPand.....	71
	:CHANnel<X>:IMPedance?.....	72
	:CHANnel<X>:INVert.....	72
	:CHANnel<X>:POSition.....	73
	:CHANnel<X>:PROBe:RATio.....	73
	:CHANnel<X>:PROBe:TYPe.....	74
	:CHANnel<X>:SCALE.....	74
<hr/>		
Math commands	:MATH:DISP.....	75
	:MATH:TYPe.....	75
	:MATH:DUAL:SOURce<X>.....	75
	:MATH:DUAL:OPERator.....	76
	:MATH:DUAL:POSition.....	76
	:MATH:DUAL:SCALE.....	77
	:MATH:FFT:SOURce.....	77
	:MATH:FFT:MAG.....	77
	:MATH:FFT:WINDow.....	78

---

	:MATH:FFT:POSition .....	78
	:MATH:FFT:SCALe.....	78
	:MATH:FFT:HORizontal:SCALe .....	79
	:MATH:FFT:HORizontal:POSition .....	79
	:MATH:DEFine .....	79
	:MATHVAR? .....	81
	:MATHVAR:VAR<X> .....	81
	:MATH:ADVanced:POSition .....	81
	:MATH:ADVanced:SCALe.....	82

---

Cursor	:CURSor:MODE .....	83
commands	:CURSor:SOURce.....	83
	:CURSor:HUNI .....	84
	:CURSor:HUSE .....	84
	:CURSor:VUNI .....	85
	:CURSor:VUSE .....	85
	:CURSor:DDT.....	85
	:CURSor:H1Position .....	86
	:CURSor:H2Position .....	86
	:CURSor:HDELta .....	87
	:CURSor:V1Position.....	87
	:CURSor:V2Position.....	87
	:CURSor:VDELta .....	88
	:CURSor:XY:RECTangular:X:POSition<X> .....	88
	:CURSor:XY:RECTangular:X:DELta .....	88
	:CURSor:XY:RECTangular:Y:POSition<X> .....	89
	:CURSor:XY:RECTangular:Y:DELta .....	89
	:CURSor:XY:POLar:RADIUS:POSition<X> .....	89
	:CURSor:XY:POLar:RADIUS:DELta .....	90
	:CURSor:XY:POLar:THETA:POSition<X> .....	90
	:CURSor:XY:POLar:THETA:DELta .....	90
	:CURSor:XY:PRODuct:POSition<X> .....	91
	:CURSor:XY:PRODuct:DELta.....	91
	:CURSor:XY:RATio:POSition<X>.....	91
	:CURSor:HTRACKing .....	92

	:CURSor:VTRACking .....	92
	:CURSor:MARK .....	92
	:CURSor:SA:SOURce .....	93
	:CURSor:XY:RATio:DELta .....	93
<hr/>		
Display	:DISPlay:INTensity:WAVEform.....	95
commands	:DISPlay:INTensity:GRATICule .....	95
	:DISPlay:INTensity:BACKLight .....	95
	:DISPlay:INTensity:BACKLight:AUTODim:ENable ....	96
	:DISPlay:INTENSITy:BACKLight:AUTODim:TIME .....	96
	:DISPlay:PERSistence .....	96
	:DISPlay:GRATICule .....	97
	:DISPlay:WAVEform .....	97
	:DISPlay:PNGOutput? .....	98
	:DISPlay:RULEr .....	98
	:DISPlay:TRANSREADouts .....	98
	:DISPlay:OUTPut .....	99
<hr/>		
Hardcopy	:HARDcopy:START .....	100
commands	:HARDcopy:MODE .....	100
	:HARDcopy:PRINTINKSaver .....	100
	:HARDcopy:SAVEINKSaver .....	101
	:HARDcopy:SAVEFORMat .....	101
	:HARDcopy:ASSIGN.....	102
<hr/>		
Measure	:MEASure:GATing.....	103
commands	:MEASure:SOURce .....	103
	:MEASure:METHod .....	103
	:MEASurement:REFLevel:PERCent:HIGH .....	104
	:MEASurement:REFLevel:PERCent:LOW .....	104
	:MEASurement:REFLevel:PERCent:MID .....	105
	:MEASurement:REFLevel:PERCent:MID2 .....	105
	:MEASure:FALL.....	105
	:MEASure:FOVShoot .....	106
	:MEASure:FPReshoot .....	106
	:MEASure:FREQUency.....	107

---

:MEASure:NWIDth .....	107
:MEASure:PDUTy.....	108
:MEASure:PERiod .....	109
:MEASure:PWIDth .....	109
:MEASure:RISe.....	110
:MEASure:ROVShoot.....	110
:MEASure:RPReshoot .....	111
:MEASure:PPULSE.....	111
:MEASure:NPULSE .....	112
:MEASure:PEDGE .....	112
:MEASure:NEDGE .....	113
:MEASure:AMPlitude.....	113
:MEASure:MEAN .....	114
:MEASure:CMEan .....	115
:MEASure:HIGH .....	115
:MEASure:LOW .....	116
:MEASure:MAX .....	116
:MEASure:MIN .....	117
:MEASure:PK2PK .....	117
:MEASure:RMS .....	118
:MEASure:CRMS.....	119
:MEASure:AREa .....	119
:MEASure:CARea .....	120
:MEASure:FRRDelay .....	120
:MEASure:FRFDelay.....	121
:MEASure:FFRDelay.....	121
:MEASure:FFFDelay.....	122
:MEASure:LRRDelay .....	123
:MEASure:LRFDelay.....	123
:MEASure:LFRDelay.....	124
:MEASure:LFFDelay.....	124
:MEASure:PHAsE.....	125
:MEASure:PFLI.....	126
:MEASure:FLI .....	126

---

Measurement commands	:MEASUrement:MEAS<X>:SOURCE<X>.....	128
	:MEASUrement:MEAS<X>:TYPE.....	128
	:MEASUrement:MEAS<X>:STATE .....	129
	:MEASUrement:MEAS<X>:VALue .....	130
	:MEASUrement:MEAS<X>:MAXimum .....	130
	:MEASUrement:MEAS<X>:MEAN .....	131
	:MEASUrement:MEAS<X>:MINimum .....	132
	:MEASUrement:MEAS<X>:STDdev .....	132
	:MEASUrement:STATIstics:MODE .....	133
	:MEASUrement:STATIstics:WEIghting .....	133
	:MEASUrement:INDICators:STAT .....	134
	:MEASUrement:INDICators:HORIZ<x>? .....	134
	:MEASUrement:INDICators:VERT<x>? .....	135
	:MEASUrement:INDICators:NUMHORZ? .....	135
	:MEASUrement:INDICators:NUMVERT? .....	135
	:MEASUrement:SHORTCut:SElect .....	136
	:MEASUrement:SHORTCut:SAve.....	136
	:MEASUrement:STATIstics .....	136
<hr/>		
Reference commands	:REF<X>:DISPlay .....	137
	:REF<X>:TIMebase:POSition .....	137
	:REF<X>:TIMebase:SCALE .....	138
	:REF<X>:OFFSet .....	138
	:REF<x>:SCALE .....	139
<hr/>		
Run command	:RUN .....	140
<hr/>		
Stop command	:STOP .....	141
<hr/>		
Single command	:SINGle .....	142
<hr/>		
Force command	:FORCe.....	143
<hr/>		
Timebase commands	:TIMebase:EXPand .....	144
	:TIMebase:POSition .....	144

---

	:TIMebase:SCALE.....	144
	:TIMebase:MODE .....	145
	:TIMebase:WINDow:POSItion .....	145
	:PLAYStop .....	145
	:TIMebase:WINDow:SCALE .....	146

---

Trigger commands	:TRIGger:FREQuency.....	147
	:TRIGger:TYPe.....	147
	:TRIGger:SOURce .....	148
	:TRIGger:COUPle.....	148
	:TRIGger:NREJ .....	148
	:TRIGger:MODE.....	149
	:TRIGger:HOLDoff.....	149
	:TRIGger:LEVel.....	149
	:TRIGger:HLEVel.....	150
	:TRIGger:LLEVel.....	150
	:TRIGger:EDGe:SLOP .....	151
	:TRIGger:DELAy:SLOP .....	151
	:TRIGger:DELAy:TYPe .....	152
	:TRIGger:DELAy:TIME .....	152
	:TRIGger:DELAy:EVENt.....	152
	:TRIGger:DELAy:LEVel .....	153
	:TRIGger:PULSEWidth:POLarity.....	153
	:TRIGger:RUNT:POLarity .....	154
	:TRIGger:RUNT:WHEn .....	154
	:TRIGger:RUNT:TIME .....	155
:TRIGger:RISEFall:SLOP .....	155	
:TRIGger:RISEFall:WHEn.....	155	
:TRIGger:RISEFall:TIME.....	156	
:TRIGger:VIDeo:TYPe .....	156	
:TRIGger:VIDeo:FIELD .....	157	
:TRIGger:VIDeo:LINE.....	157	
:TRIGger:VIDeo:POLarity .....	158	
:TRIGger:PULSe:WHEn .....	158	
:TRIGger:PULSe:TIME .....	159	

---

:TRIGger:TIMEOut:WHEn.....	159
:TRIGger:TIMEOut:TIMER .....	160
:TRIGger:STATe .....	160
:TRIGger:EXTERnal:PROBe:TYPe.....	161
:TRIGger:EXTERnal:PROBe:RATio .....	161
:TRIGger:BUS:TYPe .....	161
:TRIGger:BUS:THReshold:CH<x>.....	162
:TRIGger:BUS:B1:I2C:CONDition .....	162
:TRIGger:BUS:B1:I2C:ADDRess:MODE.....	163
:TRIGger:BUS:B1:I2C:ADDRess:TYPe.....	163
:TRIGger:BUS:B1:I2C:ADDRess:VALue .....	164
:TRIGger:BUS:B1:I2C:ADDRess:DIRectioN .....	165
:TRIGger:BUS:B1:I2C:DATA:SIZE .....	165
:TRIGger:BUS:B1:I2C:DATA:VALue.....	166
:TRIGger:BUS:B1:UART:CONDition .....	166
:TRIGger:BUS:B1:UART:RX:DATA:SIZE .....	167
:TRIGger:BUS:B1:UART:RX:DATA:VALue .....	168
:TRIGger:BUS:B1:UART:TX:DATA:SIZE .....	168
:TRIGger:BUS:B1:UART:TX:DATA:VALue .....	169
:TRIGger:BUS:B1:CAN:CONDition.....	169
:TRIGger:BUS:B1:CAN:FRAMeType .....	170
:TRIGger:BUS:B1:CAN:IDentifier:MODE.....	171
:TRIGger:BUS:B1:CAN:IDentifier:VALue .....	171
:TRIGger:BUS:B1:CAN:IDentifier:DIRectioN .....	172
:TRIGger:BUS:B1:CAN:DATA:QUALifier .....	172
:TRIGger:BUS:B1:CAN:DATA:SIZE.....	173
:TRIGger:BUS:B1:CAN:DATA:VALue.....	174
:TRIGger:BUS:B1:LIN:CONDition .....	174
:TRIGger:BUS:B1:LIN:DATA:QUALifier .....	175
:TRIGger:BUS:B1:LIN:DATA:SIZE .....	176
:TRIGger:BUS:B1:LIN:DATA:VALue .....	176
:TRIGger:BUS:B1:LIN:ERRTYPE.....	177
:TRIGger:ALTErnate .....	177
:TRIGger:BUS:B1:SPI:CONDition .....	178
:TRIGger:BUS:B1:SPI:DATA:SIZE .....	178

---

	:TRIGger:BUS:B1:SPI:DATA:MISO:VALue .....	179
	:TRIGger:BUS:B1:SPI:DATA:MOSI:VALue .....	180
	:TRIGger:BUS:B1:LIN:IDentifier:VALue .....	180
<hr/>		
System	:SYSTem:LOCK .....	182
commands	:SYSTem:ERRor.....	182
<hr/>		
Save/Recall	:RECALL:SETUp .....	183
commands	:RECALL:WAVEform .....	183
	:SAVe:IMAGe .....	184
	:SAVe:IMAGe:FILEFormat .....	184
	:SAVe:IMAGe:INKSaver.....	184
	:SAVe:SETUp.....	185
	:SAVe:WAVEform.....	185
	:SAVe:WAVEform:FILEFormat .....	186
<hr/>		
Ethernet	:ETHERnet:DNS:IPADDRESS .....	188
Command	:ETHERnet:DOMAINname .....	188
	:ETHERnet:ENET:ADDResS?.....	188
	:ETHERnet:GATEWay:IPADDRESS .....	189
	:ETHERnet:NAME.....	189
	:ETHERnet:PASSWord.....	189
	:ETHERnet:SUBNETMask .....	190
	:ETHERnet:DHCP .....	190
<hr/>		
Time Command	:DATE .....	191
<hr/>		
Bus Decode	:BUS1 .....	192
Commands	:BUS1:STATE .....	192
	:BUS1:TYPe .....	192
	:BUS1:INPut.....	193
	:BUS1:I2C:ADDResS:RWINClude.....	193
	:BUS1:I2C:SCLK:SOURce .....	194
	:BUS1:I2C:SDA:SOURce.....	194
	:BUS1:UART:BITRate.....	194
	:BUS1:UART:DATABits.....	195

---

:BUS1:UART:PARItY.....	195
:BUS1:UART:PACKEt .....	195
:BUS1:UART:POLARity .....	196
:BUS1:UART:EOFPAcket .....	196
:BUS1:UART:TX:SOURce .....	196
:BUS1:UART:RX:SOURce .....	197
:BUS1:DISPlay:FORMAt .....	197
:LISTer:DATA .....	197
:BUS1:CAN:SOURce.....	198
:BUS1:CAN:PROBe.....	198
:BUS1:CAN:SAMPLEpoint.....	198
:BUS1:CAN:BITRate .....	199
:BUS1:LIN:BITRate .....	200
:BUS1:LIN:IDFORmat .....	200
:BUS1:LIN:POLARity .....	200
:BUS1:LIN:SAMPLEpoint .....	201
:BUS1:LIN:SOURce .....	201
:BUS1:SPI:SCLK:POLARity .....	201
:BUS1:SPI:SS:POLARity.....	202
:BUS1:SPI:WORDSize .....	202
:BUS1:SPI:BITORder .....	202
:BUS1:SPI:SCLK:SOURce .....	203
:BUS1:SPI:SS:SOURce.....	203
:BUS1:SPI:MOSI:SOURce .....	203
:BUS1:SPI:MISO:SOURce .....	204
:BUS1:LIN:STANDard .....	204

---

Mark Commands	:MARK.....	205
	:MARK:CREATE.....	205
	:MARK:DELEte.....	205

---

Search	:SEARCH:COPIY .....	207
Commands	:SEARCH:STATE .....	207
	:SEARCH:TOTAL .....	207
	:SEARCH:TRIGger:TYPe .....	208

---

---

:SEARCH:TRIGger:SOURce.....	208
:SEARCH:TRIGger:EDGE:SLOP .....	209
:SEARCH:TRIGger:LEVel .....	209
:SEARCH:TRIGger:HLEVel .....	210
:SEARCH:TRIGger:LLEVel .....	210
:SEARCH:TRIGger:PULSEWidth:POLarity .....	211
:SEARCH:TRIGger:RUNT:POLarity .....	211
:SEARCH:TRIGger:RISEFall:SLOP .....	212
:SEARCH:TRIGger:PULSe:WHEn .....	212
:SEARCH:TRIGger:PULSe:TIME .....	213
:SEARCH:TRIGger:RUNT:WHEn .....	213
:SEARCH:TRIGger:RUNT:TIME.....	214
:SEARCH:TRIGger:RISEFall:WHEn .....	214
:SEARCH:TRIGger:RISEFall:TIME .....	215
:SEARCH:TRIGger:BUS:TYPe .....	215
:SEARCH:TRIGger:BUS:B1:I2C:CONDition.....	215
:SEARCH:TRIGger:BUS:B1:I2C:ADDRes:MODE .....	216
:SEARCH:TRIGger:BUS:B1:I2C:ADDRes:TYPe .....	217
:SEARCH:TRIGger:BUS:B1:I2C:ADDRes:VALue .....	217
:SEARCH:TRIGger:BUS:B1:I2C:ADDRes	
:DIRection .....	218
:SEARCH:TRIGger:BUS:B1:I2C:DATA:SIZE.....	219
:SEARCH:TRIGger:BUS:B1:I2C:DATA:VALue.....	219
:SEARCH:TRIGger:BUS:B1:UART:CONDition.....	220
:SEARCH:TRIGger:BUS:B1:UART:RX:DATA:SIZE .....	221
:SEARCH:TRIGger:BUS:B1:UART:RX:DATA:VALue .....	221
:SEARCH:TRIGger:BUS:B1:UART:TX:DATA:SIZE .....	222
:SEARCH:TRIGger:BUS:B1:UART:TX:DATA:VALue .....	222
:SEARCH:TRIGger:BUS:B1:CAN:CONDition .....	223
:SEARCH:TRIGger:BUS:B1:CAN:FRAMEtype .....	224
:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODE .....	224
:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:VALue .....	225
:SEARCH:TRIGger:BUS:B1:CAN:IDentifier	
:DIRection .....	226
:SEARCH:TRIGger:BUS:B1:CAN:DATA:QUALifier.....	226

---

:SEARCH:TRIGger:BUS:B1:CAN:DATA:SIZE .....	227
:SEARCH:TRIGger:BUS:B1:CAN:DATA:VALue .....	228
:SEARCH:TRIGger:BUS:B1:LIN:CONDition .....	228
:SEARCH:TRIGger:BUS:B1:LIN:DATA:QUALifier .....	229
:SEARCH:TRIGger:BUS:B1:LIN:DATA:SIZE .....	230
:SEARCH:TRIGger:BUS:B1:LIN:DATA:VALue .....	230
:SEARCH:TRIGger:BUS:B1:LIN:ERRTYPE .....	231
:SEARCH:TRIGger:BUS:B1:LIN:IDentifier:VALue .....	232
:SEARCH:FFTPeak:METHOD .....	232
:SEARCH:FFTPeak:METHOD:MPEak .....	233
:SEARCH:FFTPeak:SINFO .....	234
:SEARCH:TRIGger:BUS:B1:SPI:CONDition .....	234
:SEARCH:TRIGger:BUS:B1:SPI:DATA:SIZE .....	235
:SEARCH:TRIGger:BUS:B1:SPI:DATA:MISO:VALue .....	235
:SEARCH:TRIGger:BUS:B1:SPI:DATA:MOSI:VALue .....	236
:SEARCH:FFTPeak:SOURce .....	237
:SEARCH:TRIGger:BUS:TYPE .....	237
:SEARCH:FFTPeak:LIST .....	238

Label Commands	:CHANnel<X>:LABel .....	239
	:CHANnel<X>:LABel:DISPlay .....	240
	:REF<X>:LABel .....	240
	:REF<X>:LABel:DISPlay .....	241
	:BUS1:LABel .....	242
	:BUS1:LABel:DISPlay .....	242
	:SET<X>:LABel .....	243

Segment	:SEGMENTS:STATE .....	245
Commands	:SEGMENTS:CURREnt .....	245
	:SEGMENTS:TOTALnum .....	246
	:SEGMENTS:TIME .....	246
	:SEGMENTS:DISPALL .....	247
	:SEGMENTS:MEASURE:MODE .....	247
	:SEGMENTS:MEASURE:PLOT:SOURce .....	248
	:SEGMENTS:MEASURE:PLOT:DIVide .....	248

	:SEGMENTS:MEASURE:PLOT:SELECT .....	248
	:SEGMENTS:MEASURE:PLOT:RESULTS .....	249
	:SEGMENTS:MEASURE:TABLE:SOURCE .....	250
	:SEGMENTS:MEASURE:TABLE:SELECT .....	250
	:SEGMENTS:MEASURE:TABLE:LIST .....	250
	:SEGMENTS:MEASURE:TABLE:SAVE .....	251
	:SEGMENTS:SAVE .....	251
	:SEGMENTS:SAVE:SOURCE .....	252
	:SEGMENTS:SAVE:SELECT:START .....	252
	:SEGMENTS:MASK:STATE .....	252
	:SEGMENTS:MASK:SOURCE .....	253
	:SEGMENTS:MASK:CREATE .....	253
	:SEGMENTS:MASK:UNITS .....	254
	:SEGMENTS:SAVE:SELECT:END .....	254
<hr/>		
DVM Commands	:DVM:STATE .....	255
	:DVM:SOURCE .....	255
	:DVM:MODE .....	256
	:DVM:VALUE .....	256
<hr/>		
Go-NoGo Commands	:GONOGO:CLEAR .....	257
	:GONOGO:EXECUTE .....	257
	:GONOGO:FUNCTION .....	257
	:GONOGO:NGCOUNT .....	257
	:GONOGO:NGDEFINE .....	258
	:GONOGO:SOURCE .....	258
	:GONOGO:VIOLATION .....	258
	:GONOGO:SCRIPT .....	259
	:TEMPLATE:MODE .....	259
	:TEMPLATE:MAXIMUM .....	259
	:TEMPLATE:MINIMUM .....	260
	:TEMPLATE:POSITION:MAXIMUM .....	260
	:TEMPLATE:POSITION:MINIMUM .....	261
	:TEMPLATE:SAVE:MAXIMUM .....	261
	:TEMPLATE:SAVE:MINIMUM .....	261

---

:TEMPLate:TOLerance.....	261
:TEMPLate:SAVe:AUTo .....	262

---

AWG Commands	:AWG:UTIL.....	263
	:AWG:UTIL:AMPcpl .....	263
	:AWG:UTIL:FREQcpl.....	263
	:AWG:UTIL:FREQcpl:OFFSet .....	264
	:AWG:UTIL:FREQcpl:RATio.....	264
	:AWG:UTIL:TRACking.....	264
	:AWG<x>:AMPlitude.....	265
	:AWG<x>:FREQUency.....	265
	:AWG<x>:FUNCTion .....	265
	:AWG<x>:OFFSet .....	266
	:AWG<x>:OUTPut:LOAd:IMPEDance .....	266
	:AWG<x>:OUTPut:STATE .....	267
	:AWG<x>:PHAsE.....	267
	:AWG<x>:PULSe:DUTYcycle .....	267
	:AWG<x>:RAMP:SYMmetry .....	268
	:AWG<x>:MODulation:STATE .....	268
	:AWG<x>:MODulation:TYPe.....	268
	:AWG<x>:MODulation:AM:DEPth .....	269
	:AWG<x>:MODulation:AM:FREQ.....	269
	:AWG<x>:MODulation:AM:SHApe.....	269
	:AWG<x>:MODulation:AM:PHAsE .....	270
	:AWG<x>:MODulation:AM:DUTYcycle .....	270
	:AWG<x>:MODulation:AM:SYMmetry .....	270
	:AWG<x>:MODulation:AM:RATE .....	271
	:AWG<x>:MODulation:FM:DEV .....	271
	:AWG<x>:MODulation:FM:FREQ .....	272
	:AWG<x>:MODulation:FM:SHApe .....	272
	:AWG<x>:MODulation:FM:PHAsE .....	273
	:AWG<x>:MODulation:FM:DUTYcycle.....	273
	:AWG<x>:MODulation:FM:SYMmetry .....	273
	:AWG<x>:MODulation:FM:RATE.....	274
	:AWG<x>:MODulation:FSK:FREQ .....	274

---

	:AWG<x>:MODulation:FSK:RATE .....	275
	:AWG<x>:SWEp:STATE .....	275
	:AWG<x>:SWEp:TYPe .....	275
	:AWG<x>:SWEp:START .....	276
	:AWG<x>:SWEp:STOP .....	276
	:AWG<x>:SWEp:TIME.....	276
	:AWG<x>:SWEp:SPAN .....	277
	:AWG<x>:SWEp:CENTer .....	277
	:AWG<x>:ARBitrary:EDIT:NUMPoint .....	277
	:AWG<x>:ARBitrary:EDIT:FUNcTion.....	278
	:AWG<x>:ARBitrary:SAVe:WAVEform .....	278
	:AWG<x>:ARBitrary:LOAd:WAVEform .....	279
	:AWG<x>:ARBitrary:EDIT:COpy.....	279
	:AWG<x>:ARBitrary:EDIT:CLEar.....	280
	:AWG<x>:ARBitrary:EDIT:LINE .....	280
	:AWG<x>:ARBitrary:EDIT:SCALE.....	280
	:AWG<x>:ARBitrary:EDIT:POINt.....	281
	:AWG<x>:ARBitrary:EDIT:POINt:ADD.....	281
	:AWG<x>:ARBitrary:EDIT:POINt:DELEte.....	281
<hr/>		
Data Logging	:DATALOG:STATE .....	282
Commands	:DATALOG:SOURce .....	282
	:DATALOG:SAVe .....	283
	:DATALOG:INTerval .....	283
	:DATALOG:DURection.....	284
<hr/>		
Remote Disk	:REMOTEDisk:IPADdRes .....	285
Commands	:REMOTEDisk:PATHName .....	285
	:REMOTEDisk:USERName .....	285
	:REMOTEDisk:PASSWord .....	286
	:REMOTEDisk:MOUNT.....	286
	:REMOTEDisk:AUTOMount.....	286
<hr/>		
Spectrum	:SA:STATE .....	288
Analyzer	:SA:LIST.....	288

Commands	:SA:MEMory.....	288
	:SA:MEMory:SOURce.....	291
	:SA:SOURce.....	291
	:SA<x>:SPECTRUMTrace.....	292
	:SA<x>:NORMal.....	292
	:SA<x>:MAXHold.....	292
	:SA<x>:MINHold.....	293
	:SA<x>:AVErage.....	293
	:SA<x>:AVErage:NUMAVg.....	294
	:SA<x>:DETECTIonmethod:MODE.....	294
	:SA<x>:DETECTIonmethod:MAXHold.....	295
	:SA<x>:DETECTIonmethod:MINHold.....	295
	:SA<x>:DETECTIonmethod:NORMal.....	296
	:SA<x>:DETECTIonmethod:AVErage.....	296
	:SA<x>:FREQuency.....	297
	:SA<x>:SPAN.....	297
	:SA<x>:START.....	297
	:SA<x>:STOP.....	298
	:SA<x>:RBW:MODE.....	298
	:SA<x>:RBW.....	299
	:SA<x>:SPANRbwratio.....	299
	:SA<x>:WINDow.....	300
	:SA<x>:UNIts.....	301
	:SA<x>:SCALE.....	301
	:SA:POSition.....	302
	:SA<x>:INPut.....	302
	:SA:SPECTRogram:NUMSLICEs?.....	302
	:SA<x>:SPECTRogram:SLICESElect.....	303
	:SA<x>:SPECTRogram:SLICETIme?.....	303
	:SA:SPECTRogram:STATe.....	303
<hr/>		
USB Delay Command	:USBDelay.....	304
<hr/>		
FRA Commands	:FRA:RUN.....	305
	:FRA:STOP.....	305

	:FRA:SOURce:INPut .....	305
	:FRA:SOURce:OUTPut .....	306
	:FRA:FREQUency:STARt.....	306
	:FRA:FREQUency:STOP .....	306
	:FRA:AWG:LOAD.....	307
	:FRA:AWG:AMPlitude .....	307
	:FRA:POINT .....	308
	:FRA:SAVE .....	308
	:FRA:RECALL.....	308
	:FRA:DATA .....	309
	:FRA:SAVETOCsv.....	309
	:FRA:AWG:AMPlitude:PROFile .....	309
	:FRA:AWG:AMPlitude:INTERPolation .....	310
	:FRA:DATA:GMARgin .....	310
	:FRA:DATA:GMARgin:FREQUency.....	311
	:FRA:DATA:PMARgin.....	311
	:FRA:DATA:PMARgin:FREQUency.....	311
	:FRA:STATe .....	311
<hr/>		
Power supply	:POWERSupply:OUTPut<X> .....	312
Commands	:POWERSupply:OUTPut<X>:VOLTage .....	312
	:POWERSupply:OUTPut<X>:RECONFigure.....	312
	:POWERSupply:OUTPut<X>:OCP.....	313
	:POWERSupply:OUTPut<X>:CONFigure .....	313
<hr/>		
Python	:PYDEcode:TYPE.....	314
Commands	:PYDEcode:CANFD:SOURce.....	314
	:PYDEcode:CANFD:PROBe .....	315
	:PYDEcode:CANFD:SAMPLEpoint.....	315
	:PYDEcode:CANFD:BITRate .....	315
	:PYDEcode:CANFD:FASTBITRate .....	315
	:PYDEcode:FLEXRay:SOURce .....	316
	:PYDEcode:FLEXRay:BDIFF.....	316
	:PYDEcode:FLEXRay:CHTYPe .....	316
	:PYDEcode:FLEXRay:BITRate .....	317

:PYDEcode:USBPD:SOURce .....	317
:PYDEcode:USBPACKET:SIGnal .....	317
:PYDEcode:USBPACKET:DPlus.....	318
:PYDEcode:USBPACKET:DMinus .....	318
:PYDEcode:I2S:SCK .....	318
:PYDEcode:I2S:WS .....	319
:PYDEcode:I2S:SD .....	319

---

# C 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 page16.

---

Common Commands .....	37
Mask Commands .....	42
DMM Commands.....	56
Acquisition Commands.....	61
Autoscale Commands .....	69
Vertical Commands .....	70
Math Commands .....	75
Cursor Commands .....	83
Display Commands .....	95
Hardcopy Commands .....	100
Measure Commands .....	103
Measurement Commands .....	128
Reference Commands .....	137
Run Command .....	140
Stop Command .....	141
Single Command.....	142
Force Command.....	143
Timebase Commands.....	144
Trigger Commands.....	147
System Commands .....	182
Save/Recall Commands .....	183
Ethernet Command .....	188
Time Command .....	191
Bus Decode Commands .....	192

Mark Commands.....	205
Search Commands .....	207
Label Commands .....	239
Segment Commands .....	245
DVM Commands.....	255
Go_NoGo Commands .....	257
AWG Commands.....	263
Data Logging Commands.....	282
Remote Disk Commands.....	285
Spectrum Analyzer Commands .....	288
USB Delay Command .....	304
FRA Commands .....	305
Power Supply Commands.....	312
Python Commands.....	314

## Common Commands

### \*IDN?



**Description** Returns the manufacturer, model, serial number and version number of the unit.

**Syntax** \*IDN?

**Example** \*IDN?  
GW,MPO-2102B,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  
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;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

**\*SAV**

Set →

Description	Saves the current panel settings to the selected memory number (setup 1 ~ 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 MPO-2000 (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>
Query Syntax	*ESE?

Return parameter <NR1> 0~255

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?  
>4  
Indicates that there is a query error.

**\*ESR**



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

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  1            Returns 1 when all the outstanding commands have completed.

Set →

**\*SRE**

→ 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>

Query Syntax      \*SRE?

Parameter/  
Return parameter  <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**



**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> 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**      \*STB?  
 >16  
 Indicates that the MAV bit is set.

## Mask Commands

**:MASK:STATe** Set →  
→ Query

Description	Sets the mask state to on or off. Or returns the mask state.	
Syntax	:MASK:STATe {ON OFF} :MASK:STATe?	
Parameter	ON	Turn the mask function on.
	OFF	Turn the mask function off.
Example	:MASK:STATe ON Turn mask on.	

**:MASK:SOURce** Set →  
→ Query

Description	Sets or returns the compared source.	
Syntax	:MASK:SOURce {CH1 CH2 CH3 CH4} :MASK:SOURce?	
Parameter	CH1~CH4:	Channel 1 to Channel 4.
Example	:MASK:SOURce CH1 Set the compared source as channel 1.	

**:MASK:VIOLation** Set →  
→ Query

Description	Set or returns actions for the mask violations.	
Syntax	:MASK:VIOLation {STOP CONTInue} :MASK:VIOLation?	
Parameter	STOP	The waveform will be frozen.
	CONTInue	Ignore the violation.

Example :MASK:VIOLation STOP  
Sets the violation action to stop.

:MASK:VIOLation:SAVe (Set) →  
→ (Query)

Description Turns on/off the function of saving violation images. Or returns the state of saving violation images function.

Syntax :MASK:VIOLation:SAVe {ON|OFF}  
:MASK:VIOLation:SAVe?

Parameter ON Turns on this function.  
OFF Turns off this function.

Example :MASK:VIOLation:SAVe ON  
Turns the function of saving violation images on.

:MASK:AUTo (Set) →  
→ (Query)

Description Creates a mask fast in according to the reference source. Or returns the setting of the mask.

Syntax :MASK:AUTo {<Xmask>,<Ymask>}  
:MASK:AUTo?

Parameter <Xmask> Sets the horizontal range for the mask.  
<Ymask> Sets the vertical range for the mask.

Example :MASK:AUTo 0.2,1.2  
Sets the range of mask as {0.2,1.2} and creates a mask.

:MASK:AUTo:SOURce (Set) →  
→ (Query)

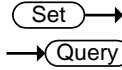
Description Sets or returns the reference source for the mask.

Syntax :MASK:AUTo:SOURce {CH1|CH2|CH3|CH4}  
:MASK:AUTo:SOURce?

Parameter	CH1~CH4	Channel 1 to Channel 4.
-----------	---------	-------------------------

Example :MASK:AUTO:SOURce CH1  
 Sets the reference source for the mask as channel 1.

**:MASK:AUTO:UNITs**

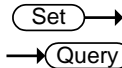


Description	Sets or returns the units for auto created mask.	
-------------	--	--

Syntax	:MASK:AUTO:UNITs {DIVIsions CURRent} :MASK:AUTO:UNITs?	
--------	---	--

Parameter	DIVIsions	Divisions units.
	CURRent	Current units.

Example :MASK:AUTO:UNITs DIVIsions  
 Sets the units for auto created mask as divisions units.



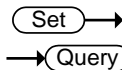
**:MASK:USER:UNITs**

Description	Sets or returns the units for the customized mask.	
-------------	--	--

Syntax	:MASK:USER:UNITs {DIVIsions CURRent} :MASK:USER:UNITs?	
--------	---	--

Parameter	DIVIsions	Divisions units.
	CURRent	Current units.

Example :MASK:USER:UNITs DIVIsions  
 Sets the units for the customized mask as divisions units.

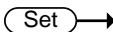



**:MASK:USER:AREa<x1>:POINT<x2>**

Description	Sets or returns the coordinates for the customized mask.	
-------------	--	--

Syntax	:MASK:USER:AREa<x1>:POINT<x2> {<XMASK>,<YMASK>} :MASK:USER:AREa<x1>:POINT<x2>?	
Parameter	<x1>	Number of the customized mask. <x1>:1~8.
	<x2>	Number of the points set up the mask area. <x2>:1~10.
	<XMASK>	Horizontal coordinates.<NRF>
	<YMASK>	Vertical coordinates.<NRF>

Example :MASK:USER:AREa1:POINT1 2,2  
Sets point 1 as (2,2) in area 1.

:MASK:USER:AREa<x1>:POINT<x2>:STATE  

Description Sets or returns the state of the point in the mask area.

Syntax :MASK:USER:AREa<x1>:POINT<x2>:STATE {ON|OFF}  
:MASK:USER:AREa<x1>:POINT<x2>:STATE?

Parameter	AREa<x1>	Number of the customized mask. <x1>:1~8.
	POINT<x2>	Number of the points set up the mask area. <x2>:1~10.
	ON	Enable the point in the mask area.
	OFF	Disable the point in the mask area.

Example :MASK:USER:AREa1:POINT1:STATE ON  
Set the state of the point 1 in area 1 to ON.

:MASK:USER:SAVE 

Description Saves a customized mask to the assigned file path with a specified filename.

Syntax	:MASK:USER:SAVe <file path> ("Disk:/xxx.MSK", "USB:/xxx.MSK")
Parameter	xxx.MSK      Filename.
Example	:MASK:USER:SAVe "Disk:/mask1.MSK" Saves a customized mask named mask1.MSK to root directory (Disk:/) of the scope.

**:MASK:USER:LOAD**

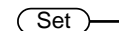


Description	Loads a customized mask from the assigned file path with a specified filename.
Syntax	:MASK:USER:LOAD <file path> ("Disk:/xxx.MSK", "USB:/xxx.MSK")
Parameter	xxx.MSK      Filename
Example	:MASK:USER:LOAD "Disk:/mask1.MSK" Loads a customized mask named mask1.MSK from root directory (Disk:/) of the scope.

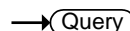
**:MASK:USER:CREATE**



Description	Create or removes the customized mask.
Syntax	:MASK:USER:CREATE {ON OFF}
Parameter	ON              Creates the mask. OFF             Removes the mask.
Example	:MASK:USER:CREATE ON Create a customized mask.



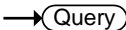
**:MASK:RATio**



Description	Resets the mask violation ratio.
Syntax	:MASK:RATio {RESET} :MASK:RATio?

Parameter	RESET	Reset
Example	:MASK:RATio RESET Resets the ratio.	

:MASK:VIOLation:THREShold 



Description	Sets or queries the violation threshold for mask.	
Syntax	:MASK:VIOLation:THREShold {<NR1>   INFInite} :MASK:VIOLation:THREShold?	
Parameter	<NR1>	Sets the violation threshold. Range:1~1000000
	INFInite	Sets violation threshold as infinite.
Example	:MASK:VIOLation:THREShold 1000 :MASK:VIOLation:THREShold? 1000	

:MASK:STOPAFter 



Description	Sets or queries the mask test stop after waveform or time violation.	
Syntax	:MASK:STOPAFter {WAVEform   TIME} :MASK:STOPAFter?	
Parameter	WAVEform	Sets the mask test stop after waveform violation.
	TIME	Sets the mask test stop after time violation.
Example	:MASK:STOPAFter TIME :MASK:STOPAFter? TIME	

Set →

→ Query

**:MASK:STOPAFter:WAVeform**

**Description** Sets or queries the numbers of waveform for stop test.

**Syntax** :MASK:STOPAFter:WAVeform {<NR1> | INFInite}  
:MASK:STOPAFter:WAVeform?

**Parameter**

<NR1>	Sets the numbers of waveform for stop test. Range:1~1000000
INFInite	Sets the numbers of waveform for stop test as infinite.

**Example** :MASK:STOPAFter:WAVeform 20000  
:MASK:STOPAFter:WAVeform?  
20000

Set →

→ Query

**:MASK:STOPAFter:TIME**

**Description** Sets or queries the numbers of time for stop test.

**Syntax** :MASK:STOPAFter:TIME{<NR1> | INFInite}  
:MASK:STOPAFter:TIME?

**Parameter**

<NR1>	Sets the numbers of time for stop test. Range:1~1000000
INFInite	Sets the numbers of time for stop test as infinite.

**Example** :MASK:STOPAFter:TIME 20000  
:MASK:STOPAFter:TIME?  
20000

Set →

→ Query

**:MASK:FAILACtion:STOPACQuisition**

**Description** Sets or queries the state of action, stop the test when test fail.

Syntax	:MASK:FAILACTion:STOPACQuisition {ON OFF} :MASK:FAILACTion:STOPACQuisition?	
Parameter	ON	Turns on the fail action: stop acuquisition.
	OFF	Turns off the fail action: stop acuquisition.
Example	:MASK:FAILACTion:STOPACQuisition ON :MASK:FAILACTion:STOPACQuisition? ON	

Set →

→ Query

**:MASK:FAILACTion:SAVEIMAge**

Description	Sets or queries the state of action, saving the screen image when test fail.	
Syntax	:MASK:FAILACTion:SAVEIMAge {ON OFF} :MASK:FAILACTion:SAVEIMAge?	
Parameter	ON	Turns on the fail action: saving the screen image.
	OFF	Turns off the fail action: saving the screen image.
Example	:MASK:FAILACTion:SAVEIMAge ON :MASK:FAILACTion:SAVEIMAge? ON	

Set →

→ Query

**:MASK:FAILACTion:SAVEWAVEform**

Description	Sets or queries the state of action, saving the waveform file when test fail.	
Syntax	:MASK:FAILACTion:SAVEWAVEform {ON OFF} :MASK:FAILACTion:SAVEWAVEform?	
Parameter	ON	Turns on the fail action: saving the waveform file.
	OFF	Turns off the fail action: saving the waveform file.

Example :MASK:FAILACTion:SAVEWAVEform ON  
 :MASK:FAILACTion:SAVEWAVEform?  
 ON

Set →

→ Query

:MASK:FAILACTion:HARDCopy

Description Sets or queries the state of action, hardcopy when test fail.

Syntax :MASK:FAILACTion:HARDCopy {ON|OFF}  
 :MASK:FAILACTion:HARDCopy?

Parameter	ON	Turns on the fail action:hardcopy.
	OFF	Turns off the fail action:hardcopy.

Example :MASK:FAILACTion:HARDCopy ON  
 :MASK:FAILACTion:HARDCopy?  
 ON

Set →

→ Query

:MASK:FAILACTion:GONogo

Description Sets or queries the state of action, output a signal from Go/NoGo port when test fail.

Syntax :MASK:FAILACTion:GONogo {ON|OFF}  
 :MASK:FAILACTion:GONogo?

Parameter	ON	Turns on the fail action:output a signal from Go/NoGo port.
	OFF	Turns off the fail action:output a signal from Go/NoGo port.

Example :MASK:FAILACTion:GONogo ON  
 :MASK:FAILACTion:GONogo?  
 ON

**:MASK:COMPLETEAction:GONogo** 


**Description** Sets or queries the state of action, output a signal from Go/NoGo port when test complete.

**Syntax** :MASK:COMPLETEAction:GONogo {ON|OFF}  
:MASK:COMPLETEAction:GONogo?

<b>Parameter</b>	ON	Turns on the complete action:output a signal from Go/NoGo port.
	OFF	Turns off the complete action:output a signal from Go/NoGo port.

**Example** :MASK:COMPLETEAction:GONogo ON  
:MASK:COMPLETEAction:GONogo?  
ON

**:MASK:COMPLETEAction:TESTRepeat** 


**Description** Sets or queries the state of action, repeat the test when test complete.

**Syntax** :MASK:COMPLETEAction:TESTRepeat {ON|OFF}  
:MASK:COMPLETEAction:TESTRepeat?

<b>Parameter</b>	ON	Turns on the complete action:repeat the test.
	OFF	Turns off the complete action:repeat the test.

**Example** :MASK:COMPLETEAction:TESTRepeat ON  
:MASK:COMPLETEAction:TESTRepeat?  
ON

**:MASK:COMPLETEAction:TESTDelay** 


**Description** Sets or queries the delay time of test repeat.

**Syntax** :MASK:COMPLETEAction:TESTDelay {<NRF>}  
:MASK:COMPLETEAction:TESTDelay?

Parameter <NRF> Sets the times of delay. Range:0~200(sec)

Example :MASK:COMPLETEAction:TESTDelay 5  
 :MASK:COMPLETEAction:TESTDelay?  
 5.0

:MASK:RESults:DISPlay Set →  
→ Query

Description Sets or queries the display state of mask results menu.

Syntax :MASK:RESults:DISPlay {ON|OFF}  
 :MASK:RESults:DISPlay?

Parameter ON Turns on the mask results menu.  
 OFF Turns off the mask results menu.

Example :MASK:RESults:DISPlay ON  
 :MASK:RESults:DISPlay?  
 ON

:MASK:RESults:DISPlay:DETail Set →  
→ Query

Description Sets or queries the display state of mask detail results menu.

Syntax :MASK:RESults:DISPlay:DETail {ON|OFF}  
 :MASK:RESults:DISPlay:DETail?

Parameter ON Turns on the mask detail results menu.  
 OFF Turns off the mask detail results menu.

Example :MASK:RESults:DISPlay:DETail ON  
 :MASK:RESults:DISPlay:DETail?  
 ON

**:MASK:RESults:CURRent:STATUS** → Query

Description	Queries the current status of mask test.	
Syntax	:MASK:RESults:CURRent:STATUS?	
Parameter	OFF	Mask test is off.
	Running (Failing)	Mask test is running, and the test is failed.
	Running (Passing)	Mask test is running, and the test is passed.
	Failed	Mask test stopped, and the test is failed.
	Passed	Mask test stopped, and the test is passed.
	Delay	Mask test stopped, and waiting for repeat test.
Example	:MASK:RESults:CURRent:STATUS? Passed	

**:MASK:RESults:CURRent:WAVEform** → Query

Description	Queries the current numbers of tested waveforms.	
Syntax	:MASK:RESults:CURRent:WAVEform?	
Example	:MASK:RESults:CURRent:WAVEform? 638	

**:MASK:RESults:CURRent:VIOLation** → Query

Description	Queries the current numbers of violations.	
Syntax	:MASK:RESults:CURRent:VIOLation?	
Example	:MASK:RESults:CURRent:VIOLation? 176	

:MASK:RESults:CURRent:TIME → Query

Description	Queries the current elapsed time(sec).
Syntax	:MASK:RESults:CURRent:TIME?
Example	:MASK:RESults:CURRent:TIME? 5

:MASK:RESults:CURRent:HITSPERSEGment<x> → Query

Description	Queries the current detailed results of hits per segment mask.
Syntax	:MASK:RESults:CURRent:HITSPERSEGment<x>?
Parameter	<x> 1~8
Example	:MASK:RESults:CURRent:HITSPERSEGment2? 255

:MASK:RESults:TOTal:WAVEform → Query

Description	Queries the total numbers of tested waveforms.
Syntax	:MASK:RESults:TOTal:WAVEform?
Example	:MASK:RESults:TOTal:WAVEform? 1256

:MASK:RESults:TOTal:VIOLation → Query

Description	Queries the total numbers of violations.
Syntax	:MASK:RESults:TOTal:VIOLation?
Example	:MASK:RESults:TOTal:VIOLation? 562

:MASK:RESults:TOTal:TESTNUMBER → Query

Description	Queries the total numbers of completed mask test.
Syntax	:MASK:RESults:TOTal:TESTNUMBER?
Example	:MASK:RESults:TOTal:TESTNUMBER? 10

:MASK:RESults:TOTal:FTESTNUMBER → Query

Description	Queries the total numbers of failed mask test.
Syntax	:MASK:RESults:TOTal:FTESTNUMBER?
Example	:MASK:RESults:TOTal:FTESTNUMBER? 3

:MASK:RESults:TOTal:TIME → Query

Description	Queries the total elapsed time(sec).
Syntax	:MASK:RESults:TOTal:TIME?
Example	:MASK:RESults:TOTal:TIME? 150

:MASK:RESults:TOTal:HITSPERSEGment<x> → Query

Description	Queries the total detailed results of hits per segment mask.
Syntax	:MASK:RESults:TOTal:HITSPERSEGment<x>?
Parameter	<x> 1~8
Example	:MASK:RESults:TOTal:HITSPERSEGment1? 36

## DMM Commands

**:DMM** → Query

Description	Returns the DMM status.	
Syntax	:DMM?	
Related commands	:MEASUREMENT:DISPLAY	
Parameter/Return parameter	<string>	Returns the mode, current measurement, max measurement, minimum measurement, Hold state.
Example	:DMM? Mode:ACV,Value:0.000,Max Value:0.000,Min Value:0.000,Hold:OFF	

**:DMM:STATE** Set →  
→ Query

Description	Sets or queries the DMM function .	
Syntax	:DMM: STATE { ON   OFF   ?}	
Parameter/Return parameter	ON	Turns the DMM function on.
	OFF	Turns the DMM function off.
Example	:DMM:STATE ON Turns the DMM function on.	

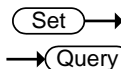
**:DMM:VALue** → Query

Description	Returns the measurement value.	
Syntax	:DMM:VALue?	
Related commands	:MEASUREMENT:DISPLAY	

Return parameter	<code>&lt;string&gt;</code>	Returns the measurement or value on the display as a string.
------------------	-----------------------------	--

Example           :DMM:VALue?  
0.000

Returns the value on the DMM display.



**:DMM:HOLD**

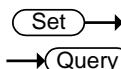
Description	Sets or queries the Hold function status.	
-------------	---	--

Syntax	:DMM:HOLD { ON   OFF   ? }	
--------	----------------------------	--

Parameter/Return parameter	ON	Turns the Hold function on.
	OFF	Turns the Hold function off.

Example           :DMM:HOLD ON

Turns the Hold function on.



**:DMM:MMIN**

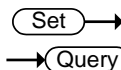
Description	Sets or queries the maximum and minimum status.	
-------------	---	--

Syntax	:DMM: MMIN { ON   OFF   ? }	
--------	-----------------------------	--

Parameter/Return parameter	ON	Turns the maximum and minimum function on.
	OFF	Turns the maximum and minimum function off.

Example           :DMM: MMIN ON

Turns the maximum and minimum function on.



**:DMM:MODE**

Description	Sets or queries the DMM mode.	
-------------	-------------------------------	--

Syntax	:DMM:MODE { DCV   DCMV   ACV   ACMV   DCA   DCMA   ACA   ACMA   OHM   DIODE   BEEP   TEMPerature   ? }	
--------	---	--

Parameter/Return parameter	DCV	DCV mode
	DCMV	DCMV mode
	ACV	ACV mode
	ACMV	ACMV mode
	DCA	DCA mode
	DCMA	DCMA mode
	ACA	ACA mode
	ACMA	ACMA mode
	OHM	Resistance measurement mode
	DIODE	Diode tester
	BEEP	Continuity tester
	TEMPerature	Temperature measurement mode

**Example**            :DMM:MODE DCV  
                              Sets the measurement mode to DCV.

Set →

**:DMM:MODE:RANGe**

→ Query

<b>Description</b>	Sets or queries the DMM measurement range.
<b>Syntax</b>	:DMM:MODE:RANGe (AUTo <NRf>) :DMM:MODE:RANGe?
<b>Related commands</b>	:DMM:MODE

Parameter/Return parameter	AUTo	Auto range
	<NRf>	ACV: 5,50,750
		DCV: 5,50,500,1000
		ACmV: 0.5,0.05
		DCmV: 0.5,0.05
		ACmA: 0.5,0.05
		DCmA: 0.5,0.05

	ACA: 10
	DCA: 10

**Example** :DMM:MODE ACV  
 :DMM:MODE:RANGe AUTO  
 Sets the ACV measurement to auto range.

Set →

**:DMM:TEMPerature:UNITs**

→ Query

**Description** Sets the units for the temperature measurement function.

**Syntax** :DMM:TEMPerature:UNITs { Celsius | Fahrenheit | ? }

<b>Parameter/Return parameter</b>	Celsius	Degrees Celsius
	Fahrenheit	Degrees Fahrenheit

**Example** :DMM:TEMPerature:TYPE Celsius  
 Sets the temperature measurement to °C.

Set →

**:DMM:TEMPerature:TYPE**

→ Query

**Description** Sets the type of thermocouple used for the temperature measurement function.

**Syntax** :DMM:TEMPerature:TYPE { TYPEB | TYPEE | TYPEJ | TYPEK | TYPEN | TYPER | TYPES | TYPET | ? }

<b>Parameter/Return parameter</b>	TYPEB	B
	TYPEE	E
	TYPEJ	J
	TYPEK	K
	TYPEN	N
	TYPER	R
	TYPES	S
	TYPET	T

Example :DMM:TEMPerature:TYPe K  
 Sets the temperature measurement function to use the K type thermocouple.

Set →

:DMM:TEMPerature:SIM

→ Query

Description Set or returns the environment temperature when temperature measurement selected.



Syntax :DMM:TEMPerature:SIM {<NRf>}  
 :DMM:TEMPerature:SIM?



Related commands :DMM:MODE  
 :DMM:TEMPerature:UNITs

Parameter/Return parameter	<NRf>	0.0~50.0 for Celsius degrees; 32~ 122.0 for Fahrenheit degrees
----------------------------	-------	---

Example :DMM:MODE TEMPerature  
 :DMM:TEMPerature:UNITs Celsius  
 :DMM:TEMPerature:SIM 23.5  
 Sets the environment temperature to 23.5 Celsius degrees.

## Acquisition Commands

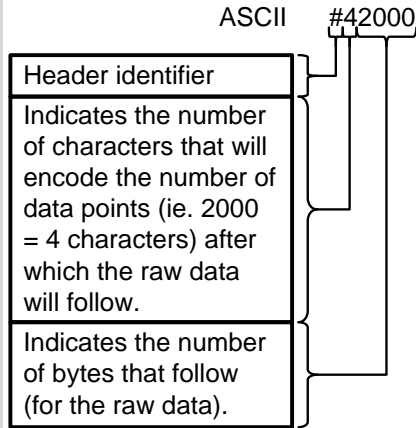
:ACQuire:AVERage	
	 
Description	Selects or returns the number of waveform acquisitions that are averaged in the average acquisition mode.
Syntax	:ACQuire:AVERage {<NR1>  ?}
Related Commands	:ACQuire:MODE
Parameter	<NR1> 2, 4, 8, 16, 32, 64, 128, 256
Note	Before using this command, select the average acquisition mode. See the example below.
Example	<pre>:ACQuire:MODE AVERage :ACQuire:AVERage 2</pre> <p>Selects the average acquisition mode, and sets the average number to 2.</p>

:ACQuire:MODE							
	 						
Description	Selects or returns the acquisition mode.						
Syntax	:ACQuire:MODE {SAMPlE   PDETECT   AVERage   ?}						
Related Commands	:ACQuire:AVERage						
Parameter	<table border="0"> <tr> <td>SAMPlE</td> <td>Sample mode sampling</td> </tr> <tr> <td>PDETECT</td> <td>Peak detect sampling</td> </tr> <tr> <td>AVERage</td> <td>Average sampling mode</td> </tr> </table>	SAMPlE	Sample mode sampling	PDETECT	Peak detect sampling	AVERage	Average sampling mode
SAMPlE	Sample mode sampling						
PDETECT	Peak detect sampling						
AVERage	Average sampling mode						
Example	<pre>:ACQuire:MODE PDETECT</pre> <p>Sets the sampling mode to peak detection.</p>						

**:ACQuire<X>:MEMory?**

→ Query

Description	Returns the data in acquisition memory for the selected channel as a header + raw data.	
Syntax	:ACQuire<X>:MEMory?	
Related Commands	:ACQuire:RECOrdlength :HEADer	
Parameter	<X>	Channel number (1 to 4)
Return parameter	<string> <waveform block data>	<p>Returns acquisition settings followed by raw waveform block data.</p> <p>&lt;string&gt; Returns the acquisition settings for the selected channel.</p> <p>Format: parameter(1),setting(1);parameter(2),setting(2)...parameter(n),setting(n);Waveform Data;</p> <p>&lt;waveform block data&gt; Header followed by the raw waveform data.</p> <p>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.</p>



**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 ~ 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:**


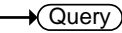
#42000 28 27 26 26 27.....

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.....follows waveform block  
 data in hex.....

 →  
 → 

**:ACquire:FILTER:SOURce**

Description	Returns the source of the filter.
Syntax	:ACquire:FILTER:SOURce {CH1 CH2 CH3 CH4 ?}
Parameter/ Return parameter	CH1 ~ CH4 Source channel
Example	:ACquire:FILTER:SOURce? CH1 Sets the filter source to CH1.

 →  
 → 

**:ACquire:FILTER**

Description	Turns the filter on/off or queries its status.
Syntax	:ACquire:FILTER {ON OFF ?}
Parameter/ Return parameter	ON Filter on. OFF Filter off.

Example :ACquire:FILTer?  
 OFF  
 Indicates that the filter is turned off.

:ACquire:FILTer:FREQuency:UPPER (Set) →  
 → (Query)

Description Sets or returns the filter upper frequency.

Syntax :ACquire:FILTer:FREQuency:UPPER {DEfault}  
 :ACquire:FILTer:FREQuency:UPPER <NRf>  
 :ACquire:FILTer:FREQuency:UPPER?

Parameter	DEfault	Sets the frequency to default.
	<NRf>	Sets the frequency to user.(Range:1Hz~500MHz)

Example :ACquire:FILTer:FREQuency:UPPER 4.95e+07  
 :ACquire:FILTer:FREQuency:UPPER?  
 4.950000e+07

:ACquire:FILTer:FREQuency:LOWER (Set) →  
 → (Query)

Description Sets or returns the filter lower frequency.

Syntax :ACquire:FILTer:FREQuency:LOWER {DEfault}  
 :ACquire:FILTer:FREQuency:LOWER <NRf>  
 :ACquire:FILTer:FREQuency:LOWER?

Parameter	DEfault	Sets the frequency to default.
	<NRf>	Sets the frequency to user.(Range:1Hz~500MHz)

Example :ACquire:FILTer:FREQuency:LOWER 1.25e+07  
 :ACquire:FILTer:FREQuency:LOWER?  
 1.250000e+07

Set →  
 → Query

**:ACQUIRE:FILTER:TYPE**

---

Description	Sets or returns the filter type.	
Syntax	:ACQUIRE:FILTER:TRACKING {LOWPASS   HIGHPASS   BANDPASS} :ACQUIRE:FILTER:TYPE?	
Parameter	LOWPASS	Lowpass Type.
	HIGHPASS	Highpass Type.
	BANDPASS	Bandpass Type.
Example	:ACQUIRE:FILTER:TYPE? >LOWPASS Returns low pass type as present filter type	

Set →  
 → Query

**:ACQUIRE:FILTER:TRACKING**

---

Description	Turns filter tracking on/off or queries its state.	
Syntax	:ACQUIRE:FILTER:TRACKING {ON OFF ?}	
Parameter/ Return parameter	OFF	Tracking off
	ON	Tracking on
Example	:ACQUIRE:FILTER:TRACKING ON Turns filter tracking on.	


→ Query



**:ACQUIRE<X>:STATE?**

---

Description	Returns the status of waveform data.	
Syntax	:ACQUIRE<X>:STATE?	
Parameter	<X>	Channel number (1 to 4)
Return parameter	0	Raw data is not ready
	1	Raw data is ready


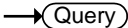
Example :ACquire1:STATE?  
0  
Returns 0. Channel 1's raw data is not ready.

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

**:ACquire:INTERpolation** 
 →  
 → 

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

Example :ACquire:INTERpolation?  
>SINC  
Returns SINC as the interpolation mode.

**:ACquire:RECOrdlength** 
 →  
 → 

Description	Sets or queries the record length.	
Syntax	:ACquire:RECOrdlength {<NRf>  ?}	
Parameter/Return parameter	<NRf>	Record length. Settable record length: (1e+3   1e+4   1e+5   1e+6   1e+7)

Example :ACquire:RECOrdlength 1e+3  
Sets the record length to 1000 points.

**:ACquire:SAMPLerate?** → 

Description	Queries the value of sample rate.
Syntax	: ACquire:SAMPLerate?

Example : ACQuire:SAMPLerate?  
1.00000E+09

:HEADer (Set) →  
→ (Query)

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?

Parameter	ON	Add header information.
	OFF	Don't add header information.

Return parameter Returns the configuration (ON, OFF) for the selected channel.

Example :HEADer ON

## Autoscale Commands

:AUTOSet .....	69
:AUTORSET:MODE .....	69

### :AUTOSet

Set →

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

**Syntax** :AUTOSet

Set →

### :AUTORSET:MODE

→ Query

**Description** Sets the Autoset mode or queries its state.

**Syntax** :AUTORSET:MODE {FITScreen | ACPriority | ?}

**Related Commands** :AUTOSet

<b>Parameter/Return parameter</b>	FITScreen	Fit Screen mode
	ACPriority	AC priority mode

**Example** :AUTORSET?  
FITSCREEN

## Vertical Commands

**:CHANnel<X>:BWLimit** (Set) →  
→ (Query)

Description	Sets or returns the bandwidth limit on/off.	
Syntax	:CHANnel<X>:BWLimit {FULL   <NR3>   ?}	
Parameter	<X>	Channel 1, 2, 3, 4
	FULL	Full bandwidth
	<NR3>	Sets the bandwidth limit to a pre-defined bandwidth.
		100E+6: 100MHz
		20E+6: 20MHz
Return Parameter	<NR3>	Returns the bandwidth.
	Full	Full bandwidth
Example	:CHANnel1:BWLimit 2.000E+07 Sets the channel 1 bandwidth to 20MHz.	

**:CHANnel<X>:COUPling** (Set) →  
→ (Query)

Description	Selects or returns the coupling mode.	
Syntax	CHANnel<X>:COUPling {AC   DC   GND   ?}	
Parameter	<X>	Channel 1, 2, 3, 4
	AC	AC coupling
	DC	DC coupling
	GND	Ground coupling
Return parameter	Returns the coupling mode.	
Example	:CHANnel1:COUPling DC Sets the coupling to DC for Channel 1.	

Set →  
 → Query

**:CHANnel<X>:DESKew**

Description	Sets the deskew time in seconds.	
Syntax	:CHANnel<X>:DESKew { <NR3>   ?}	
Parameter	<X>	Channel 1, 2, 3, 4
	<NR3>	Deskew time: -5.00E-11 to 5.00E-11 -50ns to 50 ns. (10 ps /step)
Return parameter	<NR3>	Returns the deskew time.
Example	:CHANnel1:DESKew 1.300E-9 Sets the deskew time to 1.3 nano seconds.	

Set →  
 → Query

**:CHANnel<X>:DISPlay**

Description	Turns a channel on/off or returns its status.	
Syntax	:CHANnel<X>:DISPlay {OFF   ON   ?}	
Parameter	<X>	Channel 1, 2, 3, 4
	OFF	Channel off
	ON	Channel on
Return Parameter	ON	Channel is on
	OFF	Channel is off
Example	:CHANnel1:DISPlay ON Turns on Channel 1	

Set →  
 → Query

**:CHANnel<X>:EXPanD**

Description	Sets Expand By Ground or Expand By Center for a channel or queries its status.	
Syntax	:CHANnel<X>:EXPanD {GND   CENTer   ?}	
Parameter	<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>:IMPedance?** → Query

Description	Returns the impedance of the oscilloscope. (The impedance of the MPO-2000 is fixed at 1MΩ)	
Syntax	:CHANnel<X>:IMPedance?	
Parameter	<x>	Channel
	1/2/3/4	CH1/2/3/4
Return parameter	<NR3>	Returns the impedance value.
Example	:CHANnel1:IMPedance? 1.000000E+06 The impedance is 1M ohms.	


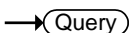
**:CHANnel<X>:INVert** Set →  
→ Query

Description	Inverts a channel or returns its status.	
Syntax	:CHANnel<X>:INVert {OFF   ON   ?}	
Parameter	<X>	Channel 1, 2, 3, 4
	OFF	Invert off
	ON	Invert on
Return parameter	ON	Invert on
	OFF	Invert off
Example	:CHANnel1:INVert ON Inverts Channel 1	


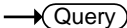
**:CHANnel<X>:POSition** 


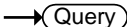


Description	Sets or returns the position level for a channel.	
Note	<p>The vertical position will only be set to closest allowed value. The position level range depends on the vertical scale.</p> <p>The scale must first be set before the position can be set.</p>	
Syntax	:CHANnel<X>:POSition { <NRf>   ?}	
Parameter	<X>	Channel 1, 2, 3, 4
	<NRf>	Position. Range depends on the vertical scale.
Return parameter	<NR3>	Returns the position value.
Example 1	<p>:CHANnel1:POSition 2.4E-3</p> <p>Sets the Channel 1 position to 2.4mV/mA</p>	
Example 2	<p>:CHANnel1:POSition?</p> <p>2.4E-3</p> <p>Returns 2.4mV as the vertical position.</p>	

**:CHANnel<X>:PROBe:RATio** 



Description	Sets or returns the probe attenuation factor.	
Syntax	:CHANnel<X>:PROBe:RATio { <NRf>   ?}	
Related Commands	:CHANnel<X>:PROBe:TYPE	
Parameter	<X>	Channel 1, 2, 3, 4
	<NRf>	Probe attenuation factor
Return parameter	<NR3>	Returns the probe factor
Example	<p>:CHANnel1:PROBe:RATio 1.00E+0</p> <p>Sets the Channel 1 probe attenuation factor to 1x</p>	

		 
<b>:CHANnel&lt;X&gt;:PROBe:TYPe</b>		
Description	Sets or returns the probe type (voltage/current).	
Syntax	:CHANnel<X>:PROBe:TYPe { VOLTage   CURRent   ?}	
Related Commands	:CHANnel<X>:PROBe:RATio	
Parameter	<X>	Channel 1, 2, 3, 4
	VOLTage	Voltage
	CURRent	Current
Return parameter	Returns the probe type.	
Example	:CHANnel1:PROBe:TYPe VOLTage Sets the Channel 1 probe type to voltage.	

		 
<b>:CHANnel&lt;X&gt;:SCALE</b>		
Description	Sets or returns the vertical scale. The scale depends on the probe attenuation factor. Note the probe attenuation factor should be set before the scale.	
Syntax	:CHANnel<X>:SCALE { <NRf>   ?}	
Parameter	<X>	Channel 1, 2, 3, 4
	<NRf>	Vertical scale: 2e-3 to 1e+1 2mV to 10V (Probe x1)
Return parameter	<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** 



**Description** Turns the math display on or off on the screen.

**Syntax** :MATH:DISP {OFF|ON|?}

<b>Parameter/ Return parameter</b>	OFF	Math is not displayed on screen
	ON	Math is displayed on screen

**Example** :MATH:DISP OFF  
Math is off.

**:MATH:TYPE** 



**Description** Queries or sets the Math type to FFT, Advanced Math or to dual channel math operations

**Syntax** :MATH:TYPE { DUAL | ADVanced | FFT | ? }

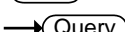
**Related Commands** :MATH:DISP

<b>Parameter</b>	DUAL	Dual channel operations
	ADVanced	Advanced math operations
	FFT	FFT operations

**Return parameter** Returns the math type.

**Example** :MATH:TYPE DUAL  
Sets the Math type to dual channel math operation.

**:MATH:DUAL:SOURce<X>** 



**Description** Sets the dual math source for source 1 or 2.

**Syntax** :MATH:DUAL:SOURce<X> { CH1 | CH2 | CH3 | CH4 | REF1 | REF2 | REF3 | REF4 | ? }

Parameter	<X>	Source number 1 or 2
	CH1~4	Channel 1 to 4
	REF1~4	Reference waveforms 1 to 4

Return parameter Returns the source for the source 1 or 2.

Example :MATH:DUAL:SOURce1 CH1  
Sets source1 as channel 1.

Set →

→ Query

**:MATH:DUAL:OPERator**

Description Sets the math 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 operator type.

Example :MATH:DUAL:OPERator PLUS  
Sets the math operator as plus (+).

Set →

→ Query

**:MATH:DUAL:POSition**

Description Sets the vertical position of the displayed math result expressed by unit/division.

Syntax :MATH:DUAL:POSition {<NRf> | ? }

Parameter	<NRf>	Vertical position
		Depends on the vertical scale (Unit/Div)


Return parameter <NR3> Returns the vertical position.

Example :MATH:DUAL:POSition 1.0E+0  
Sets the vertical position to 1.00 unit/div.


**:MATH:DUAL:SCALE**

Description	Sets the vertical scale of the displayed math result.	
Syntax	:MATH:DUAL:SCALE {<NRf>   ?}	
Parameter	<NRf>	Vertical scale
Return parameter	<NR3>	Returns the scale.
Example	:MATH:DUAL:SCALE 2.0E-3 Sets the vertical scale to 2mV/2mA.	


**:MATH:FFT:SOURce**

Description	Sets and queries the FFT math source.	
Syntax	:MATH:FFT:SOURce { CH1   CH2   CH3   CH4   REF1   REF2   REF3   REF4   ? }	
Related commands	:MATH:ADVanced:EDIT:SOURce<X> :MATH:ADVanced:EDIT:OPERator	
Parameter	CH1~4 REF1~4	Channel 1 to 4 Reference waveform 1 to 4
Return parameter	Returns the FFT source.	
Example	:MATH:FFT:SOURce CH1 Sets the FFT math source as channel 1.	


**:MATH:FFT:MAG**

Description	Sets FFT vertical units as linear or decibels.	
Syntax	:MATH:FFT:MAG {LINEAR   DB   ?}	
Parameter	LINEAR DB	Linear units (Vrms) Logarithmic units (dB)
Return parameter	Returns the FFT vertical units.	

Example :MATH:FFT:MAG DB  
Sets FFT vertical units to dB.

Set →

:MATH:FFT:WINDow

→ Query

Description Sets the windowing filter used for the FFT function.

Syntax :MATH:FFT:WINDow  
{RECTangular|HAMming|HANning|BLAckman|?}

Parameter	RECTangular	Rectangular window
	HAMming	Hamming window
	HANning	Hanning window
	BLAckman	Blackman window

Return parameter Returns the FFT window.

Example :MATH:FFT:WINDow HAMming  
Sets the FFT window filter to hamming.

Set →

:MATH:FFT:POSition

→ Query

Description Sets the vertical position of the displayed FFT result.

Syntax MATH:FFT:POSition { <NRf> | ? }

Parameter	<NRf>	Vertical position: -12e+0 to +12e+0 (12 units/division to +12 units/division.)
-----------	-------	---

Return parameter	<NR3>	Returns the vertical position.
------------------	-------	--------------------------------

Example :MATH:FFT:POSition -2e-1  
Sets the FFT position to -0.2 divisions.

Set →

:MATH:FFT:SCALE

→ Query

Description Sets the vertical scale of the displayed FFT result.

Syntax :MATH:FFT:SCALE {<NRf> | ?}

Parameter	<NRf>	Vertical scale: Linear: 2e-3 to 1e+3 (2mV~1kV) dB: 1e+0 to 2e+1 (1~20dB)
-----------	-------	--

Return parameter	<NR3>	Returns vertical scale.
------------------	-------	-------------------------

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

:MATH:FFT:HORizontal:SCALE (Set) →  
→ (Query)

Description	Sets or queries the zoom scale for FFT math.	
-------------	--	--

Syntax	:MATH:FFT:HORizontal:SCALE {<NRf>   ?}	
--------	--	--

Parameter	<NRf>	Zoom scale: 1 to 20 times
-----------	-------	---------------------------

Return parameter	<NR3>	Returns zoom scale.
------------------	-------	---------------------

Example :MATH:FFT:HORizontal:SCALE 5  
Sets the zoom scale to 5X.

:MATH:FFT:HORizontal:POSITION (Set) →  
→ (Query)

Description	Sets the horizontal position of the displayed FFT result.	
-------------	---	--

Syntax	MATH:FFT:HORizontal:POSITION { <NRf>   ? }	
--------	--	--

Parameter	<NRf>	Horizontal position: 0Hz ~ 999.9kHz
-----------	-------	-------------------------------------

Return parameter	<NR3>	Returns the vertical position.
------------------	-------	--------------------------------

Example :MATH:FFT:HORizontal:POSITION 6.0e5  
Sets the FFT horizontal position to 600kHz.

:MATH:DEFine (Set) →  
→ (Query)

Description	Sets or queries the advanced math expression as a string.	
-------------	---	--

Syntax	:MATH:DEFine {<string>  ?}	
--------	----------------------------	--

Related :MATH:DISP  
:MATH:TYPE

Parameter	<string>	An expression enclosed in double quotes. Note that ensure parentheses are used correctly in the expression. The expression can contain the following parts:
	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
	Measurement	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?**

→ Query

Description	Returns the value of the VAR1 and VAR2 variables.
Syntax	MATHVAR?
Related Commands	MATHVAR:VAR<X> MATH:DEFine
Return parameter	<string> VAR1 <NR3>; VAR2 <NR3>
Example	MATHVAR? VAR1 1.000000E+06; VAR2 1.0E+1 Returns the value of both variables.

Set →

**:MATHVAR:VAR<X>**

→ Query

Description	Sets or returns the VAR1 or VAR2 variables.
Syntax	MATHVAR:VAR<x> {<NRf>   ?}
Related Commands	MATH:DEFine
Parameter	<X> 1, 2 (VAR1 or VAR2) <NRf> Value of VAR1/VAR2
Return parameter	<NR3> Returns the value of VAR1/VAR2
Example	:MATH:VAR1 6.0e4 Sets VAR1 to 60000.

Set →

**:MATH:ADVanced:POSition**

→ Query

Description	Sets the vertical position of the advanced math result, expressed in unit/div.
Syntax	MATH:ADVanced:POSition { <NRf>   ? }




Parameter	<NRf>	Vertical position: -12e+0 to +12e+0 (12 units/division to +12 units/division.)
Return parameter	<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>   ?}	
Parameter	<NRf>	Vertical scale
Return parameter	<NR3>	Returns the vertical scale.
Example	:MATH:ADVanced:SCALE 2.0E-3 Sets the vertical scale to 2mV/Div.	

## Cursor Commands

		 
<b>:CURSor:MODE</b>		
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.
	H	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.	

		 
<b>:CURSor:SOURce</b>		
Description	Sets or queries the cursor source.	
Syntax	:CURSor:SOURce {CH1   CH2   CH3   CH4   REF1   REF2   REF3   REF4   MATH   LOGic   BUS1   ? }	
Parameter	CH1~CH4	Channel 1 to 4
	REF1~4	Reference waveform 1 to 4
	MATH	Math source
	LOGic	Logic source
	BUS1	Bus source
Return parameter	Returns the cursor source.	
Example	:CURSor:SOURce CH1 Turns the cursor source as channel 1.	

Set →

→ Query


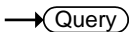
**:CURSor:HUNI**

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	Sets the current cursor position as the phase or ratio reference for the Percent or Degrees (horizontal) cursors.	
Note	This command can only be used when :CURSor:HUNI is set to DEGrees or PERcent.	
Syntax	:CURSor:HUSE {CURRent}	
Related Commands	:CURSor:MODE :CURSor:HUNI	
Parameter	CURRent	Uses the current horizontal position
Example	:CURSor:HUSE CURRent.	

**:CURSor:VUNI**

---


Description	Sets or queries the units for the vertical bar cursors.	
Syntax	:CURSor:VUNI {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 unit type.	
Example	:CURSor:VUNI BASE Sets the units to the base units.	



**:CURSor:VUSE**

---

Description	Sets the current cursor position as the ratio reference for 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:VUSE CURRent.	



**:CURSor:DDT**

---

Description	Returns the deltaY/deltaT (dy/dT) readout. This function is only supported if the source channels are CH1~4, Ref1~4 or Math.	
Syntax	:CURSor:DDT{?}	

Related Commands :CURSor:MODE

Return Parameter <NR3> Returns the readout in <NR3> format.

Example :CURSor:DDT?  
4.00E-05

Set →

→ Query

### :CURSor:H1Position

Description Sets or returns the first horizontal cursor (H1) position.

Syntax :CURSor:H1Position {<NRf>| ?}

Related Commands :CURSor:H2Position

Parameter <NRf> Horizontal position

Return parameter Returns the cursor position.

Example :CURSor:H1Position?  
-1.34E-3  
Returns the H1 cursor position as -1.34ms.

Set →

→ Query

### :CURSor:H2Position

Description Sets or returns the second horizontal cursor (H2) position.

Syntax :CURSor:H2Position {<NRf>| ?}

Related Commands :CURSor:H1Position

Parameter <NRf> Horizontal Position

Return parameter Returns the cursor position.

Example :CURSor:H2Position 1.5E-3  
Sets the H2 cursor position to 1.5ms.

**:CURSor:HDELta**

→ Query

---

Description	Returns the delta of H1 and H2.
Syntax	:CURSor:HDELta{?}
Return Parameter	<NR3> Returns the distance between two horizontal cursors.
Example	:CURSor:HDELta? 5.0E-9 Returns the horizontal delta as 5ns.

Set →

**:CURSor:V1Position**

→ Query

---

Description	Sets the first vertical cursor (V1) position.
Syntax	:CURSor:V1Position {<NRf>  ?}
Parameter	<NRf> Vertical position. Depends on the vertical scale.
Return parameter	<NR3> Returns the cursor position.
Example	:CURSor:V1Position 1.6E -1 Sets the V1 cursor position to 160mA.

Set →

**:CURSor:V2Position**

→ Query

---

Description	Sets the first vertical cursor (V2) position.
Syntax	:CURSor:V2Position {<NRf>  ?}
Parameter	<NRf> Vertical position. Depends on the vertical scale.
Return parameter	<NR3> Returns the cursor position.
Example	:CURSor:V2Position 1.1E-1 Sets the V2 cursor position to 110mA.

**:CURSor:VDELta** → Query

Description	Returns the delta of V1 and V2.	
Syntax	:CURSor:VDELta{?}	
Return Parameter	<NR3>	Returns the difference between two vertical cursors.
Example	:CURSor:VDELta? 4.00E+0 Returns the vertical delta as 4 volts.	

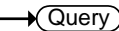
**:CURSor:XY:RECTangular:X:POSition<X>** Set →  
→ Query

Description	Sets or queries the horizontal position in XY mode for the X rectangular coordinates for cursor 1 or 2.	
Syntax	:CURSor:XY:RECTangular:X:POSition<X> {<NRf> ?}	
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 40mV/mV.	

**:CURSor:XY:RECTangular:X:DELta** → Query

Description	Returns the delta value of cursor 1 and 2 on the X coordinate.	
Syntax	:CURSor:XY:RECTangular:X:DELta{?}	
Return Parameter	<NR3>	Returns the delta value of cursor 1 and 2 as <NR3>.

Example :CURSor:XY:RECTangular:X:DELta?  
80.0E-3  
Returns the horizontal delta as 80mV.

:CURSor:XY:RECTangular:Y:POSition<X>  

Description Sets or queries the vertical position in XY mode for the Y rectangular coordinates for cursor 1 or 2.

Syntax :CURSor:XY:RECTangular:Y:POSition<X> {<NRf>|?}

Parameter	<X>	Cursor 1, 2
	<NRf>	Vertical position co-ordinates

Return parameter <NR3> Returns the cursor position.

Example :CURSor:XY:RECTangular:Y:POSition1 4.0E-3  
Sets the Y-coordinate cursor 1 position to 40mV/mV.

:CURSor:XY:RECTangular:Y:DELta 

Description Returns the delta value of cursor 1 and 2 on the Y coordinate.

Syntax :CURSor:XY:RECTangular:Y:DELta{?}

Return Parameter	<NR3>	Returns the delta value of cursor 1 and 2 as <NR3>.
------------------	-------	---

Example :CURSor:XY:RECTangular:Y:DELta?  
80.0E-3  
Returns the horizontal delta as 80mV.

:CURSor:XY:POLar:RADIUS:POSition<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>{?}	
Parameter	<X>	1, 2 (cursor 1, cursor 2)
Return Parameter	<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** → Query

Description	Returns the radius delta value of cursor 1 and 2.	
Syntax	:CURSor:XY:POLar:RADIUS:DELta{?}	
Return Parameter	<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>** → Query

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>{?}	
Parameter	<X>	1, 2 (Cursor 1, Cursor 2)
Return parameter	<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** → Query

Description	Queries the polar angle delta between cursor1 and cursor2.	
Syntax	:CURSor:XY:POLar:THETA:DELta{?}	

Return parameter	<code>&lt;NR3&gt;</code>	Returns the theta delta between cursor1 and cursor2.
------------------	--------------------------	--

Example           :CURSor:XY:POLar:THETA:DELta?  
                   9.10E+0  
                   Returns the delta as 9.1°.

**:CURSor:XY:PRODUct:POSItion<X>** → Query

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>{?}
--------	-----------------------------------

Parameter	<code>&lt;X&gt;</code>	1, 2 (Cursor 1, Cursor 2)
-----------	------------------------	---------------------------

Return parameter	<code>&lt;NR3&gt;</code>	Returns the product value of the Cursor1 or Cursor2.
------------------	--------------------------	--

Example           :CURSor:XY:PRODUct:POSItion1?  
                   9.44E-5  
                   Returns the product of cursor1 as 94.4uVV.

**:CURSor:XY:PRODUct:DELta** → Query

Description	Queries the product delta in XY mode.
-------------	---------------------------------------

Syntax	:CURSor:XY:PRODUct:DELta{?}
--------	-----------------------------

Return parameter	<code>&lt;NR3&gt;</code>	Returns the product delta.
------------------	--------------------------	----------------------------

Example           :CURSor:XY:PRODUct:DELta?  
                   1.22E-5  
                   Returns the product delta as 12.2uVV.

**:CURSor:XY:RATio:POSItion<X>** → Query

Description	Queries the ratio in XY mode for the specified cursor, where x can be either cursor 1 or 2.
-------------	---

Syntax	:CURSor:XY:RATio:POSItion<X>{?}
--------	---------------------------------

Parameter	<X>	1, 2 (Cursor 1, Cursor 2)
Return parameter	<NR3>	Returns the ratio.

Example :CURSor:XY:RATio:POSition?  
 6.717E+1  
 Returns the ratio value as 6.717V/V.

Set →

→ Query

**:CURSor:HTRACking**

Description Sets or queries the state of horizontal cursor track.

Syntax :CURSor:HTRACking {ON|OFF}  
 :CURSor:HTRACking?

Example :CURSor:HTRACking ON  
 :CURSor:HTRACking?  
 ON

Set →

→ Query

**:CURSor:VTRACking**

Description Sets or queries the state of vertical cursor track.

Syntax :CURSor:VTRACking {ON|OFF}  
 :CURSor:VTRACking?

Example :CURSor:VTRACking ON  
 :CURSor:VTRACking?  
 ON

Set →

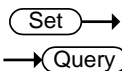
→ Query

**:CURSor:MARK**

Description Sets or queries the state of cursor mark.

Syntax :CURSor:MARK {ON|OFF}  
 :CURSor:MARK?

Example :CURSor:MARK ON  
 :CURSor:MARK?  
 ON



**:CURSor:SA:SOURce**

Description	Sets or queries the SA source of cursor.
Syntax	:CURSor:SA:SOURce {SA1 SA2} :CURSor:SA:SOURce?
Related Commands	:CURSor:SOURce :CURSor:HUNI :CURSor:HUSE :CURSor:DDT :CURSor:H1Position :CURSor:H2Position :CURSor:HDELta :CURSor:V1Position :CURSor:V2Position When in SA Mode, the SA source affects the target of the above commands.

Parameter	SA1	Sets cursor SA source as SA1.
	SA2	Sets cursor SA source as SA2.

Example CURSor:SA:SOURce SA2  
 CURSor:SA:SOURce?  
 SA2

**:CURSor:XY:RATio:DELta**



Description	Queries the ratio delta in XY mode.
Syntax	:CURSor:XY:RATio:DELta{?}
Return parameter	<NR3> Returns the ratio delta.

Example           :CURSor:XY:RATio:DELta?  
                  5.39E+1  
                  Returns the ratio delta as 53.9V/V.

## Display Commands

**:DISPlay:INTensity:WAVEform** 
 Set →  
 Query

**Description** Sets or queries the waveform intensity level.

**Syntax** :DISPlay:INTensity:WAVEform {<NRf> | ?}

**Parameter** <NRf> 0.0E+0 to 1.0E+2 (0% to 100%)

**Return Parameter** <NR3> Returns the intensity.

**Example** :DISPlay:INTensity:WAVEform 5.0E+1  
Sets the waveform intensity to 50%.

**:DISPlay:INTensity:GRATICule** 
 Set →  
 Query

**Description** Sets or queries the graticule intensity level.

**Syntax** :DISPlay:INTensity:GRATICule {<NRf> | ?}

**Parameter** <NRf> 1.0E+0 to 1.0E+2 (10% to 100%)

**Return Parameter** <NR3> Returns the graticule intensity.

**Example** :DISPlay:INTensity:GRATICule 5.0E+1  
Sets the graticule intensity to 50%.

**:DISPlay:INTensity:BACKLight** 
 Set →  
 Query

**Description** Sets or queries the intensity of the backlight display.

**Syntax** :DISPlay:INTensity:BACKLight {<NRf> | ?}

**Parameter** <NRf> 1.0E+0 to 1.0E+2 (10% to 100%)

**Return Parameter** <NR3> Returns the backlight intensity.

**Example** :DISPlay:INTensity:BACKLight 5.0E+1  
Sets the backlight intensity to 50%.

:DISPlay:INTensity:BACKLight:AUTODim      (Set) →  
 :ENable      → (Query)

Description	Sets or queries the display auto-dim function.	
Syntax	:DISPlay:INTensity:BACKLight:AUTODim:ENable {OFF   ON   ?}	
Parameter/ Return parameter	OFF	Turn auto-dim on.
	ON	Turn auto-dim off.
Example	:DISPlay:INTensity:BACKLight:AUTODim:ENable ON Turns the auto-dim function on.	

:DISPlay:INTENSITy:BACKLight:AUTODim      (Set) →  
 :TIME      → (Query)

Description	Sets or queries the display auto-dim time.	
Syntax	:DISPlay:INTensity:BACKLight:AUTODim:TIME { <NR1>   ? }	
Parameter/ Return parameter	<NR1>	1 ~ 180 minutes. Time in minutes.
Example	:DISPlay:INTensity:BACKLight:AUTODim:TIME 10 Sets the auto-dim time to 10 minutes.	

:DISPlay:PERsistence      (Set) →  
    → (Query)

Description	Sets or queries the waveform persistence level.	
Syntax	:DISPlay:PERsistence { INFINite   OFF   <NRf>   ? }	
Parameter	<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>	Returns the persistence time.
	INFIinite	Infinite persistence
	OFF	No persistence

Example :DISPlay:PERSistence 2.0E+0  
Sets the persistence to 2 seconds.





Set →

→ Query


### :DISPlay:GRATicule

Description Sets or queries graticule display type.

Syntax :DISPlay:GRATicule {FULL | GRID|CROSs | FRAMe | ?}

Parameter	FULL		CROSs	
	FRAMe		GRID	

Return parameter Returns the graticule type.

Example :DISPlay:GRATicule FULL  
Sets the graticule to .

Set →

→ Query

### :DISPlay:WAVEform

Description Sets or queries whether the waveforms are drawn as vectors or dots.

Syntax :DISPlay:WAVEform {VECTor | DOT | ?}

Parameter	VECTor	Vectors
	DOT	Dots

Return parameter Returns VECTOR or DOT.

Example :DISPlay:WAVEform VECTor  
Sets the waveform to vectors.

**:DISPlay:PNGOutput?**

→ **Query**

Description	Return the current screen image as PNG image data.
Syntax	:DISPlay:PNGOutput?
Example	:DISPlay:PNGOutput? #516643\89PNG\r\n\1A\n\00\00\00\rIHDR...

**Set** →

**:DISPlay:RULER**

→ **Query**

Description	Sets or queries the state of ruler display.
Syntax	:DISPlay:RULER {ON OFF} :DISPlay:RULER?
Parameter	ON Turns on the ruler display. OFF Turns off the ruler display.
Example	:DISPlay:RULER ON :DISPlay:RULER? ON

**Set** →

**:DISPlay:TRANSREADouts**

→ **Query**

Description	Sets or queries the state of transparent display.
Syntax	:DISPlay:TRANSREADouts {ON OFF} :DISPlay:TRANSREADouts?
Parameter	ON Turns on the transparent display. OFF Turns off the transparent display.
Example	:DISPlay:TRANSREADouts ON :DISPlay:TRANSREADouts? ON

---

**:DISPlay:OUTPut**

→ Query

---

Description	Returns the screen image as a 16 bit RGB run length encoded image.
-------------	--

---

Syntax	:DISPlay:OUTPut{?}
--------	--------------------

---

Return parameter	Format: header+data+LF
------------------	------------------------

For example assuming the image data size is 60072 bytes then the following would be returned:

#560072<[count] [color] [count] [color]..... ><LF>

Where #560072 is the header, each [count] and [color] data are 2 bytes and <LF> is a line feed character.

---

## Hardcopy Commands

### :HARDcopy:START

Set →

**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

### :HARDcopy:MODE

Set →

→ Query

**Description** Sets or queries whether hardcopy is set to print or save.

**Syntax** :HARDcopy:MODE { PRINT | SAVE | ? }

**Related Commands** :HARDcopy:START

<b>Parameter</b>	PRINT	Print mode
	SAVE	Save mode

**Return parameter** Returns the mode.(PRINT/SAVE)

**Example** :HARDcopy:MODE PRINT  
Sets hardcopy to print.

### :HARDcopy:PRINTINKSaver


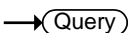
Set →

→ Query


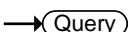
**Description** Sets Inksaver On or Off for printing.

**Syntax** :HARDcopy:PRINTINKSaver { OFF | ON | ? }

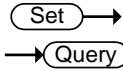
Related Commands	:HARDcopy:START :HARDcopy:MODE	
Parameter	ON	Inksaver ON
	OFF	Inksaver OFF
Return parameter	Returns the print Ink Saver mode.(ON/OFF)	
Example	:HARDcopy:PRINTINKSaver ON Sets Ink Saver to ON for printing.	

**: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:START :HARDcopy:MODE	
Parameter	PNG	PNG file format
	BMP	BMP file format
Return parameter	Returns the image file format (PNG/BMP).	
Example	:HARDcopy:SAVEFORMat PNG Sets the file format to PNG.	



**:HARDcopy:ASSIGN**

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** 



Description	Sets or queries 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:GATing OFF Turns gating off (full record).	

**:MEASure:SOURce** 



Description	Sets or queries the measurement source for source1 or source2.	
Syntax	:MEASure:SOURce<X> { CH1   CH2  CH3   CH4   MATH   ? }	
Parameter	<X>	Source1 or source2
	CH1~CH4	Channel 1 to 4
	MATH	Math
Return parameter	Returns the source (CH1, CH2, CH3, CH4, MATH)	
Example	:MEASure:SOURce1 CH1 Sets source1 to channel 1.	

**:MEASure:METHod** 



Description	Sets or queries the method used to determine the High-Low measurement 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.

Set →

:MEASUrement:REFLevel:PERCent:HIGH → Query

Description Sets or queries the high reference level as a percentage.

Syntax :MEASUrement:REFLevel:PERCent:HIGH {<NRF> | ?}

Parameter	<NRF>	0 - 100%
-----------	-------	----------

Return parameter Returns the high reference level

Example :MEASUrement:REFLevel:PERCent:HIGH 50.1  
Set the high reference level to 50.1%.

Set →

:MEASUrement: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%
-----------	-------	----------

Return parameter Returns the low reference level.

Example :MEASUrement:REFLevel:PERCent:LOW 40.1  
Set the low reference level to 40.1%.

**:MEASUrement:REFLevel:PERCent:MiD** 



**Description** Sets or queries the first mid reference level as a percentage.

**Syntax** :MEASUrement:REFLevel:PERCent:MiD {<NRf> | ?}

**Parameter** <NRf> 0 - 100%

**Return parameter** Returns the mid reference level.

**Example** :MEASUrement:REFLevel:PERCent:MiD 50  
Set the mid reference level to 50%.

**:MEASUrement:REFLevel:PERCent:MiD2** 



**Description** Sets or queries the second mid reference level as a percentage.

**Syntax** :MEASUrement:REFLevel:PERCent:MiD2 {<NRf> | ?}

**Parameter** <NRf> 0 - 100%

**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:FALL** 


**Description** Returns the fall time measurement result.

**Syntax** :MEASure:FALL{?}

**Related Commands** :MEASure:SOURce<X>

**Return parameter** <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 measures the fall time.

**:MEASure:FOVShoot**



Description

Returns the fall overshoot amplitude.

Syntax

```
:MEASure:FOVShoot{?}
```

Related

```
:MEASure:SOURce<X>
```

Commands

Return parameter

<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


```
:MEASure:SOURce<X>
```

Commands

Returns

Returns the fall preshoot as <NR3>.

Return parameter	<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 Channel 1, and then measures the fall preshoot.


**:MEASure:FREQuency** → Query

**Description** Returns the frequency value.

**Syntax** :MEASure:FREQuency{?}

**Related Commands** :MEASure:SOURce<X>

Return parameter	<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** → Query

**Description** Returns the first negative pulse width timing.

**Syntax** :MEASure:NWIDth{?}

Related Commands :MEASure:SOURce<X>

Return parameter	<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>


Return parameter	<NR3>	Returns the positive duty ratio.
	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:PDUTy?  
 5.000E+01  
 Selects Channel 1, and then measures the positive duty cycle.


**:MEASure:PERiod**

→ Query

Description	Returns the period.	
Syntax	:MEASure:PERiod{?}	
Related Commands	:MEASure:SOURce<X>	
Return parameter	<NR3>	Returns the period.
	Chan Off	Indicates the source channel is not activated.
 Note	Before using this command, select the measurement channel. See the example below.	
Example	<pre>:MEASure:SOURce1 CH1 :MEASure:PERiod? 1.0E-3</pre> Selects Channel 1, and then measures the period.	

**:MEASure:PWIDth**


→ Query

Description	Returns the first positive pulse width.	
Syntax	:MEASure:PWIDth{?}	
Related Commands	:MEASure:SOURce<X>	
Return parameter	<NR3>	Returns the positive pulse width.
	Chan Off	Indicates the source channel is not activated.
 Note	Before using this command, select the measurement channel. See the example below.	
Example	<pre>:MEASure:SOURce1 CH1 :MEASure:PWIDth? 5.0E-6</pre> Selects Channel 1, and then measures the positive pulse width.	

**:MEASure:RISe**

→ Query

Description	Returns the first pulse rise time.	
Syntax	:MEASure:RISe{?}	
Related Commands	:MEASure:SOURce<X>	
Return parameter	<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**

→ Query

Description	Returns the rising overshoot over the entire waveform in percentage.	
Syntax	:MEASure:ROVShoot{?}	
Related Commands	:MEASure:SOURce<X>	
Return parameter	<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** → Query

Description       Returns rising preshoot over the entire waveform in percentage.

Syntax            :MEASure:RPReshoot{?}

Related Commands   :MEASure:SOURce<X>

Return parameter	<NR3>	Returns the rising preshoot.
	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:PPULSE** → Query

Description       Returns the number of positive pulses.

Syntax            :MEASure:PPULSE{?}

Related Commands   :MEASure:SOURce<X>

Return parameter	<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>

Return parameter <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:PEDGE{?}

Related Commands :MEASure:SOURce<X>

Return parameter <NR3> Returns the number of positive 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: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

```
:MEASure:SOURce<X>
```

Commands

Return parameter

```
<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  $V_{high}$ - $V_{low}$ .

Syntax

```
:MEASure:AMPLitude{?}
```

Related Commands :MEASure:SOURce<X>

Return parameter	<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>

Return parameter	<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:MEAN?  
 1.82E-3  
 Selects Channel 1, and then measures the mean value.

**:MEASure:CMEan**

→ Query

**Description** Returns the mean voltage/current of one full period.

**Syntax** :MEASure:CMEan{?}

**Related Commands** :MEASure:SOURce<X>

<b>Return parameter</b>	<b>&lt;NR3&gt;</b>	Returns the cyclic mean.
	<b>Chan Off</b>	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:HIGH**

→ Query

**Description** Returns the global high voltage/current.

**Syntax** :MEASure:HIGH{?}

**Related Commands** :MEASure:SOURce<X>

<b>Return parameter</b>	<b>&lt;NR3&gt;</b>	Returns the high value.
	<b>Chan Off</b>	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** → Query

Description Returns the global low voltage/current.

Syntax :MEASure:LOW{?}

Related Commands :MEASure:SOURce<X>

Return parameter	<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 low current/voltage.

**:MEASure:MAX** → Query

Description Returns the maximum amplitude.

Syntax :MEASure:MAX{?}

Related Commands :MEASure:SOURce<X>

Return parameter	<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

```
:MEASure:SOURce<X>
```

Commands

Return parameter

```
<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** → **Query**

**Description** Returns the root-mean-square voltage/current of one or more full periods.

**Syntax** :MEASure:RMS{?}

**Related Commands** :MEASure:SOURce<X>

Return parameter	<NR3>	Returns the RMS 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:RMS?  
 1.31E-3  
 Selects Channel 1, and then measures the RMS voltage/current.

**:MEASure:CRMS**


→ **Query**

**Description** Returns the root-mean-square voltage/current of one full periods.

**Syntax** :MEASure:CRMS{?}

**Related Commands** :MEASure:SOURce<X>

<b>Return parameter</b>	<b>&lt;NR3&gt;</b>	Returns the CRMS value.
	<b>Chan Off</b>	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:CRMS?  
 1.31E-3  
 Selects Channel 1, and then measures the CRMS voltage/current.

**:MEASure:AREa**


→ **Query**

**Description** Returns the voltage/current area over one or more full periods.

**Syntax** :MEASure:AREa{?}

**Related Commands** :MEASure:SOURce<X>


<b>Return parameter</b>	<b>&lt;NR3&gt;</b>	Returns the area value.
	<b>Chan Off</b>	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**


Description	Returns the voltage/current area over one full period.	
Syntax	:MEASure:CARea{?}	
Related Commands	:MEASure:SOURce<X>	
Return parameter	<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:FRRDelay{?}	
Related Commands	:MEASure:SOURce<X>	
Return parameter	<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** → 

**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>

<b>Return parameter</b>	<b>&lt;NR3&gt;</b>	Returns the delay.
	<b>Chan Off</b>	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 Commands :MEASure:SOURce<X>

Return parameter	<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?  
 -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>

Return parameter	<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:FFFDelay?  
 -8.89E-6  
 Select channel 1 and 2 as delay source1/2, and then measure 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>

Return parameter	<b>&lt;NR3&gt;</b>	Returns the delay.
	Chan Off	Indicates the source channel is not activated.

**Note**

Select the two source channels before entering this command.

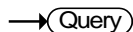
Example :MEASure:SOURce1 CH1

:MEASure:SOURce2 CH2

:MEASure:LRRDelay?

-8.89E-6

Select channel 1 and 2 as delay source1/2, and then measure LRR.

**:MEASure:LRFDelay**

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>

Return parameter	<b>&lt;NR3&gt;</b>	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?  
 -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>

Return parameter	<b>&lt;NR3&gt;</b>	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  
 Select channel 1 and 2 as delay source1/2, and then measure LFR.


**:MEASure:LFFDelay** → Query

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>

Return parameter	<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:LFFDelay?
-9.99E-6
```

Select channel 1 and 2 as delay source1/2, and then measure LFF.

**:MEASure:PHase** → 

Description	Returns the phase between source 1 and source 2.
Syntax	:MEASure:PHase{?}
Related Commands	:MEASure:SOURce<X>

Return parameter	<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.

**:MEASure:PFLI**

→ Query

Description	Returns the % flicker of times.
Syntax	:MEASure:PFLI?
Related Commands	:MEASure:SOURce<x>
Return parameter	<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:PFLI ?  
                   5.950E+01  
 Selects Channel 1 as the source, and then measures the % flicker of times.

**:MEASure:FLI**

→ Query

Description	Returns the flicker idx of times.
Syntax	:MEASure:FLI?
Related Commands	:MEASure:SOURce<x>
Return parameter	<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:FLI ?  
                  2.870E-01  
                  Selects Channel 1 as the source, and then measures  
                  the flicker idx of times.

## Measurement Commands

:MEASUrement:MEAS<X>:SOURCE<X> (Set) →  
→ (Query)

**Description** Sets or queries the measurement source for a selected automatic measurement. This is a statistics related command.

**Syntax** :MEASUrement:MEAS<X>:SOURCE<X> { CH1 | CH2 | CH3 | CH4 | ? }

**Related commands** :MEASUrement:MEAS<X>:TYPE

<b>Parameter</b>	MEAS<X>	The automatic measurement number from 1 to 8.
	SOURCE<X>	SOURCE1: the source for all single channel measurements.
	SOURCE<X>	SOURCE2: the source for all delay or phase measurements.
	CH1 to CH4	Channel 1, 2, 3, 4
	MATH	Math source

<b>Return parameter</b>	CH1 to CH4	Channel 1, 2, 3, 4
	MATH	Math source

**Example** :MEASUrement:MEAS1:SOURCE1?>CH1  
Returns the (first) source for measurement 1.

:MEASUrement:MEAS<X>:TYPE (Set) →  
→ (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 | CRM s | AREa | CARea |  
 ROVShoot | FOVShoot | RPReshoot | FPReshoot |  
 FREQuency | PERIod | RISE | FALL | PWIdth | NWIdth |  
 PDUTy | PPULSE | NPULSE | PEDGE | NEDGE |  
 PFLIcker | FLIcker | FRRDelay | FRFDelay | FFRDelay |  
 FFFDelay | LRRDelay | LRFDelay | LFRDelay | LFFDelay  
 | PHAse | ?}

Related commands :MEASUrement:MEAS<X>:SOURCE<X>

Parameter MEAS<X> The automatic measurement number from 1 to 8.

Return parameter Returns the measurement type

Example :MEASUrement:MEAS1:TYPe RMS  
 Sets measurement 1 to RMS measurement.

Set →

:MEASUrement:MEAS<X>:STATE → 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 | ? }

Related commands :MEASUrement:MEAS<X>:SOURce<X>  
 :MEASUrement:MEAS<X>:TYPe


Parameter MEAS<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.

**:MEASUrement:MEAS<X>:VALue** → **Query**

Description	Returns the measurement results for the selected measurement. This is a statistics related command.	
Syntax	:MEASUrement:MEAS<X>:VALue?	
Related Commands	:MEASUre:SOURce<X>	
Parameter	MEAS<X>	The automatic measurement number from 1 to 8.
Return parameter	<NR3>	Returns the measurement for the selected measurement number.

 **Note** The measurement source(s), measurement number, measurement type and measurement state must first be set before a measurement result can be returned.

**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** → **Query**

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?	
Related Commands	:MEASUrement:STATIstics:MODE	

Parameter	MEAS<X>	The automatic measurement number from 1 to 8.
Return parameter	<NR3>	Returns the measurement for the selected measurement number.
Example	<pre>:MEASUrement:MEAS3:SOUrce1 CH1 :MEASUrement:MEAS3:TYPe PK2PK :MEASUrement:MEAS3:STATE ON :MEASUrement:STATIstics:MODE ON :MEASUrement:MEAS3:MAXimum? 2.800E-02</pre> <p>Returns the maximum measurement result for measurement number 3.</p>	

### :MEASUrement:MEAS<X>:MEAN



**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** :MEASUrement:MEAS<X>:MEAN?

**Related Commands** :MEASUrement:STATIstics:MODE

Parameter	MEAS<X>	The automatic measurement number from 1 to 8.
-----------	---------	---

Return parameter	<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:MEAN?
2.090E-02
```

Returns the mean measurement result for measurement number 3.

**:MEASUrement:MEAS<X>:MINIumum** → **Query**

**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>:MINIumum?

**Related Commands** :MEASUrement:STATIstics:MODe

<b>Parameter</b>	MEAS<X>	The automatic measurement number from 1 to 8.
------------------	---------	---

<b>Return parameter</b>	<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:MINIumum?
1.600E-02
```

Returns the minimum measurement result for measurement number 3.

**:MEASUrement:MEAS<X>:STDdev** → **Query**

**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?

**Related Commands** :MEASUrement:STATIstics:MODe

Parameter	MEAS<X>	The automatic measurement number from 1 to 8.
Return parameter	<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:STDdev?
1.530E-03
```

Returns the standard deviation for measurement number 3.

**:MEASUrement:STATIstics:MODE** 



Description Puts the statics measurement results on the display or queries whether the statistics are displayed.

Syntax :MEASUrement: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

```
:MEASUrement:STATIstics:MODE ON
```

Displays statistics on the screen.

**:MEASUrement:STATIstics:WEIghting** 



Description Sets and queries the number of samples (weighting) used for the statistics calculations.

Syntax :MEASUrement:STATIstics:WEIghting { <NR1> | ? }

Parameter/	<NR1>	Number of samples (2~1000)
Return parameter		

Example :MEASUrement:STATistics:WEIghting 5  
Sets the number of samples to 5.

Set →

:MEASUrement:INDICators:STAT

→ Query

Description Set or query the measurement indicator's state.

Syntax :MEASUrement:INDICators:STAT {OFF|MEAS<x>}  
:MEASUrement:INDICators:STAT?

Parameter	OFF	Turn off the indicator.
	MEAS<x>	Set the indicator source. <x>:1~8.

Example :MEASUrement:INDICators:STAT OFF  
:MEASUrement:INDICators:STAT?  
OFF  
:MEASUrement:INDICators:STAT MEAS2  
:MEASUrement:INDICators:STAT?  
MEAS2

:MEASUrement:INDICators:HORIZ<x>?

→ Query

Description Query the position of measurement indicator's horizontal track.

Syntax :MEASUrement:INDICators:HORIZ<x>? {PRECise}

Parameter	HORIZ<x>	The horizontal track. <x>:1 or 2.
	PRECise	Display more digit for the return value.

Example :MEASUrement:INDICators:HORIZ1?  
3.120e+00  
:MEASUrement:INDICators:HORIZ1? PRECise  
3.120000e+00

**:MEASUrement:INDICators:VERT<x>?** → **Query**

Description	Query the position of measurement indicator's vertical track.
Syntax	:MEASUrement:INDICators:VERT<x>? {PRECise}
Parameter	<p>VERT&lt;x&gt; The vertical track. &lt;x&gt;:1 or 2.</p> <p>PRECise Display more digit for the return value.</p>
Example	<pre>:MEASUrement:INDICators:VERT1? -2.135e-02 :MEASUrement:INDICators:VERT1? PRECise -2.135000e-02</pre>

**:MEASUrement:INDICators:NUMHORZ?** → **Query**

Description	Query the number of measurement indicator's horizontal tracks currently being displayed.
Syntax	:MEASUrement:INDICators:NUMHORZ?
Example	<pre>:MEASUrement:INDICators:NUMHORZ? 2</pre>

**:MEASUrement:INDICators:NUMVERT?** → **Query**

Description	Query the number of measurement indicator's vertical tracks currently being displayed.
Syntax	:MEASUrement:INDICators:NUMVERT?
Example	<pre>:MEASUrement:INDICators:NUMVERT? 1</pre>

**:MEASUrement:SHORTCut:SElect** (Set) →  
→ (Query)

Description	Set or query the measurement shortcut selection.	
Syntax	:MEASUrement:SHORTCut:SElect {USER SHORTCUT1 SHORTCUT2  SHORTCUT3  SHORTCUT4} :MEASUrement:SHORTCut:SElect?	
Parameter	USER	Make measurement select the user settings.
	SHORTCUT1~ SHORTCUT4	Make measurement select the shortcut settings.
Example	:MEASUrement:SHORTCut:SElect SHORTCUT2 :MEASUrement:SHORTCut:SElect? SHORTCUT2	

**:MEASUrement:SHORTCut:SAVE** (Set) →

Description	Set user measurement setting to the shortcut.	
Syntax	:MEASUrement:SHORTCut:SAVE {SHORTCUT1 SHORTCUT2  SHORTCUT3  SHORTCUT4}	
Parameter	SHORTCUT1~ SHORTCUT4	Measurement saving destination.
Example	:MEASUrement:SHORTCut:SAVE SHORTCUT1	

**:MEASUrement:STATIstics** (Set) →

Description	Resets the statics calculations. This command will clear all the currently accumulated measurements.	
Syntax	:MEASUrement:STATIstics {RESET}	

## Reference Commands

**:REF<X>:DISPlay** 


**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>:DISPlay { OFF| ON| ? }

<b>Parameter</b>	<b>&lt;X&gt;</b>	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:DISPlay ON  
Turns on reference1 (REF 1) on the display.

**:REF<X>:TIMebase:POSition** 


**Description** Sets or returns the selected reference waveform time base position.


**Syntax** :REF<X>:TIMebase:POSition { <NRf> | ? }

**Related commands** :REF<X>:DISPlay

<b>Parameter</b>	<b>&lt;X&gt;</b>	Reference waveform 1, 2, 3, 4.
	<b>&lt;NRf&gt;</b>	Horizontal co-ordinates

**Return parameter** <NR3> Returns the reference waveform position

**Example** :REF1:TIMEbase:POSition -5.000E-5  
 Selects reference 1, and then sets the horizontal position to -50us.

**:REF<X>:TIMEbase:SCALE** 



<b>Description</b>	Sets or returns the selected reference waveform time base scale.	
<b>Syntax</b>	:REF<X>:TIMEbase:SCALE { <NRf>   ?}	
<b>Related commands</b>	:REF<X>:DISPlay	
<b>Parameter</b>	<X>	Reference waveform 1, 2, 3, 4.
	<NRf>	Horizontal scale
<b>Return parameter</b>	<NR3>	Returns the reference waveform scale.


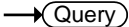
**Example** :REF1:TIMEbase:SCALE 5.00E-4  
 Selects reference 1, and then sets the horizontal scale to 500us/div.

**:REF<X>:OFFSet** 



<b>Description</b>	Sets or returns the selected reference waveform vertical position (offset).	
<b>Syntax</b>	:REF<X>:OFFSet { <NRf>   ?}	
<b>Related commands</b>	:REF<X>:DISPlay	
<b>Parameter</b>	<X>	Reference waveform 1, 2, 3, 4.
	<NRf>	Vertical offset
<b>Return parameter</b>	<NR3>	Returns the reference waveform vertical position.

**Example** :REF1:OFFSet -5.000E-2  
 Selects reference 1, and then sets the vertical position to -50mV/mA.

		 
<b>:REF&lt;x&gt;:SCALE</b>		
Description	Sets or returns the selected reference waveform vertical scale.	
Syntax	:REF<X>:SCALE { <NRf>   ? }	
Related commands	:REF<X>:DISPlay	
Parameter	<X>	Reference waveform 1, 2, 3, 4.
	<NRf>	Vertical scale
Return parameter	<NR3>	Returns the reference waveform vertical scale.
Example	:REF1:SCALE 5.000E-2 Selects reference 1, and then sets the vertical scale to 50mV   mA/div.	

## Run Command

---

:RUN



---

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



---

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



---

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



---

Description	The Force command forces an acquisition (equivalent to pressing the Force-Trig key on the front panel).
-------------	---

---

Syntax	:FORCe
--------	--------

---

## Timebase Commands

**:TIMEbase:EXPand** (Set) →  
→ (Query)

Description	Sets or queries the horizontal expansion mode.	
Syntax	:TIMEbase:EXPand {CENTer TRIGger ?}	
Parameter/Return parameter	CENTer	Expand from the center of the display.
	TRIGger	Expand from the trigger point.
Example	:TIMEbase:EXPand TRIGger Sets the expansion point to the trigger point.	

**:TIMEbase:POSition** (Set) →  
→ (Query)

Description	Sets or queries the horizontal position.	
Syntax	:TIMEbase:POSition {<NRf>   ?}	
Parameter	<NRf>	Horizontal position
Return parameter	<NR3>	Returns the horizontal position
Example	:TIMEbase:POSition 5.00E-4 Sets the horizontal position as 500us.	

**:TIMEbase:SCALE** (Set) →  
→ (Query)

Description	Sets or queries the horizontal scale.	
Syntax	:TIMEbase:SCALE {<NRf>   ?}	
Parameter	<NRf>	Horizontal scale
Return parameter	<NR3>	Returns the horizontal scale.
Example	:TIMEbase:SCALE 5.00E-2 Sets the horizontal scale to 50ms/div.	

Set →  
 → Query

<b>:TIMebase:MODE</b>							
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	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; padding: 5px;">MAIN</td> <td style="padding: 5px;">Sets the time base mode to the main screen.</td> </tr> <tr> <td style="padding: 5px;">WINDow</td> <td style="padding: 5px;">Sets the time base mode to the zoom window.</td> </tr> <tr> <td style="padding: 5px;">XY</td> <td style="padding: 5px;">Sets the time base mode to the XY display.</td> </tr> </table>	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.
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 →  
 → Query

<b>:TIMebase:WINDow:POSition</b>	
Description	Sets or queries the zoom horizontal position.
Syntax	:TIMebase:WINDow:POSition {<NRf>   ?}
Related commands	:TIMebase:MODE
Parameter	<NRf> Horizontal position for zoom window
Return parameter	<NR3> Returns the zoom horizontal position.
Example	:TIMebase:WINDow:POSition 2.0E-3 Sets the zoom horizontal position as 20ms.

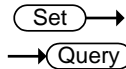
Set →

<b>:PLAYStop</b>	
Description	Set the zoom play/stop or play the current segment in segments mode.

Syntax :PLAYStop {ON|OFF}  
:PLAYStop?

Parameter ON Play  
OFF Stop

Example :PLAYStop ON  
:PLAYStop?  
ON



**:TIMebase:WINDow:SCALe**

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>   ?}	
Related commands	:TIMebase:MODE	
Parameter	<NRf>	Zoom horizontal scale. The range will depend on the time base.
Return parameter	<NR3>	Returns the zoom horizontal scale.
Example	:TIMebase:WINDow:SCALe 2.0E-3 Sets the zoom horizontal scale to 2ms.	

## Trigger Commands

:TRIGger:FREQuency

→ Query

Description	Queries the trigger frequency.
Syntax	:TRIGger:FREQuency{?}
Return parameter	<NR3> Returns the trigger frequency.
Example	:TRIGger:FREQuency? 1.032E+3 Returns the 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	<table border="0"> <tr> <td>EDGE</td> <td>Edge trigger</td> </tr> <tr> <td>DELay</td> <td>Delay trigger</td> </tr> <tr> <td>PULSEWidth</td> <td>Pulse width trigger</td> </tr> <tr> <td>VIDeo</td> <td>Video trigger</td> </tr> <tr> <td>RUNT</td> <td>Runt trigger</td> </tr> <tr> <td>RISEFall</td> <td>Rise and fall trigger</td> </tr> <tr> <td>BUS</td> <td>Bus trigger</td> </tr> <tr> <td>TIMEOut</td> <td>Timeout trigger</td> </tr> </table>	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
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:TYPe EDGE Sets the trigger type to edge.																

Set →  
 → Query

**:TRIGger:SOURce**

---

Description	Sets or queries the trigger source.						
Syntax	:TRIGger:SOURce { CH1   CH2   CH3   CH4   EXT   LINe   ? }						
Parameter	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; background-color: #e0e0e0;">CH1 to CH4</td> <td>Channel 1 to channel 4</td> </tr> <tr> <td style="background-color: #e0e0e0;">EXT</td> <td>External source</td> </tr> <tr> <td style="background-color: #e0e0e0;">LINe</td> <td>AC Line</td> </tr> </table>	CH1 to CH4	Channel 1 to channel 4	EXT	External source	LINe	AC Line
CH1 to CH4	Channel 1 to channel 4						
EXT	External source						
LINe	AC Line						
Return parameter	Returns the trigger source.						
Example	:TRIGger:SOURce CH1 Sets the trigger source to channel 1.						

Set →  
 → Query

**:TRIGger:COUple**

---

Description	Sets or queries the trigger coupling.								
Note	Applicable for edge and delay triggers only.								
Syntax	:TRIGger:COUple {AC   DC   HF   LF   ?}								
Parameter	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; background-color: #e0e0e0;">AC</td> <td>AC mode</td> </tr> <tr> <td style="background-color: #e0e0e0;">DC</td> <td>DC mode</td> </tr> <tr> <td style="background-color: #e0e0e0;">HF</td> <td>High frequency rejection</td> </tr> <tr> <td style="background-color: #e0e0e0;">LF</td> <td>Low frequency rejection</td> </tr> </table>	AC	AC mode	DC	DC mode	HF	High frequency rejection	LF	Low frequency rejection
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 →  
 → Query

**:TRIGger:NRE]**

---

Description	Sets or queries noise rejection status.
Syntax	:TRIGger:NRE] {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.

Set →

→ Query

### :TRIGger:MODE

Description Sets or queries the trigger mode.

Syntax :TRIGger:MODE {AUTo | NORMal | ?}

Parameter	AUTo	Auto trigger (Untriggered roll)
	NORMal	Normal trigger

Return parameter Returns the trigger mode.

Example :TRIGger:MODE NORMAL  
Sets the trigger mode to normal.

Set →

→ Query

### :TRIGger:HOLDoff

Description Sets or queries the holdoff time.

Syntax :TRIGger:HOLDoff {<NRf> | ?}

Parameter <NRf> Holdoff time

Return parameter <NR3> Returns the trigger holdoff time.

Example :TRIGger:HOLDoff 1.00E-8  
Sets the trigger holdoff time to 10ns.

Set →

→ Query

### :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> | ?}

Related commands :TRIGger:TYPe

Parameter	<NRf>	Trigger level value.
	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> Returns the trigger level.

Example1 :TRIGger:LEVel TTL  
Sets the trigger to TTL.

Example2 :TRIGger:LEVel 3.30E-1  
Sets the trigger level to 330mV/mA.

Set →

→ Query

### :TRIGger:HLEVel

Description Sets or queries the high trigger level.

Note Applicable for Rise and Fall/Pulse Runt triggers.

Syntax :TRIGger:HLEVel {<NRf> | ?}

Related commands :TRIGger:TYPe

Parameter	<NRf>	High level value.
-----------	-------	-------------------

Return parameter <NR3> Returns the trigger high level.

Example :TRIGger:HLEVel 3.30E-1  
Sets the trigger high level to 330mV/mA.

Set →

→ Query

### :TRIGger:LLEVel

Description Sets or queries the low trigger level.

Note Applicable for Rise and Fall/Pulse Runt triggers.

Syntax	:TRIGger:LLEVel {<NRf>   ?}	
Related commands	:TRIGger:TYPe	
Parameter	<NRf>	Low level value.
Return parameter	<NR3>	Returns the trigger low level.
Example	:TRIGger:LLEVel -3.30E-3 Sets the trigger low level to -330mV/mA.	


**:TRIGger:EDGE:SLOP**

Description	Sets or queries the trigger slope.	
Syntax	:TRIGger:EDGE: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:EDGE:SLOP FALL Sets the trigger slope to falling.	

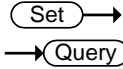
  


**:TRIGger:DELAy:SLOP**

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  
Sets the trigger slope to falling.



**:TRIGger:DElay:TYPE**

Description Sets or queries the trigger delay type.

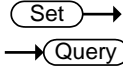
Syntax :TRIGger:DElay:TYPE {TIME | EVENT | ?}

Related commands :TRIGger:TYPE

Parameter	TIME	Sets the delay type to time.
	EVENT	Sets the delay type to event.

Return parameter Returns the trigger delay type.

Example :TRIGger:DElay:TYPE TIME  
Sets the delay type to time delay.



**:TRIGger:DElay:TIME**

Description Sets or queries the delay time value.

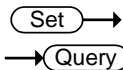
Syntax :TRIGger:DElay:TIME {<NRf> | ?}

Related commands :TRIGger:DElay:TYPE

Parameter	<NRf>	Delay time (1.00E-8~1.00E+1)
-----------	-------	------------------------------

Return parameter	<NR3>	Returns the delay time.
------------------	-------	-------------------------

Example :TRIGger:DElay:TIME 1.00E-6  
Sets the delay time to 1us.



**:TRIGger:DElay:EVENT**

Description Sets or queries the number of events for the event delay trigger.

Syntax	:TRIGger:DElay:EVENT {<NR1>   ?}	
Related commands	:TRIGger:DElay:TYPe	
Parameter	<NR1>	1~65535 events
Return parameter	<NR1>	Returns the number of events.
Example	:TRIGger:DElay:EVENT 2 Sets the number of events to 2.	

Set →  
 → Query

**:TRIGger:DElay:LEVel**

Description	Sets or queries the trigger delay level.	
Syntax	:TRIGger:DElay:LEVel {<NRf>   ?}	
Parameter	<NRf>	Delay trigger level
Return parameter	<NR3>	Returns the delay trigger.
Example	:TRIGger:DElay:LEVel 5.00E-3 Sets the delay trigger level to 5mV/mA.	

Set →  
 → Query

**:TRIGger:PULSEWidth:POLarity**

Description	Sets or queries the pulse width trigger polarity.	
Syntax	:TRIGger:PULSEWidth:POLarity {POSitive   NEGative   ?}	
Related commands	:TRIGger:TYPe	
Parameter	POSitive	Positive polarity
	NEGative	Negative polarity
Return parameter	Returns the pulse width polarity.	
Example	:TRIGger:PULSEWidth:POLarity POSitive Sets the pulse width polarity to positive.	

Set →  
→ Query

**:TRIGger:RUNT:POLarity**

---

Description	Sets or queries the Pulse Runt trigger polarity.	
Syntax	:TRIGger:RUNT:POLarity { POSitive   NEGative   EITher   ? }	
Related commands	:TRIGger:TYPe	
Parameter	POSitive	Positive polarity
	NEGative	Negative polarity
	EITher	Positive or negative polarity
Return parameter	Returns the pulse runt trigger polarity.	
Example	:TRIGger:RUNT:POLarity POSitive Sets the Pulse Runt trigger polarity to positive.	

Set →  
→ Query

**:TRIGger:RUNT:WHEn**

---

Description	Sets or queries 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 Pulse Runt trigger condition to unequal (≠).	

Set →  
→ Query

---

:TRIGger:RUNT:TIME	
Description	Sets or queries the Pulse Runt trigger time.
Syntax	:TRIGger:RUNT:TIME {<NRf>   ? }
Related commands	:TRIGger:TYPE :TRIGger:RUNT:WHEn
Parameter	<NRf> Pulse runt time (4nS to 10S)
Return Parameter	<NR3> Returns the runt time in seconds.
Example	:TRIGger:RUNT:TIME 4.00E-5 Sets the runt time to 40.0uS.

Set →  
→ Query

---

: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	Returns the rise & fall slope.
Example	:TRIGger:RISEFall:SLOP RISe Sets the Rise & Fall slope to rising.

Set →  
→ Query

---

: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 (Set) →  
 → (Query)

Description Sets or queries the Rise and Fall time.

Syntax :TRIGger:RISEFall:TIME {<NRf> | ? }

Related commands :TRIGger:TYPe  
 :TRIGger:RISEFall:WHEn

Parameter	<NRf>	Rise and Fall time (4nS to 10S)
Return Parameter	<NR3>	Returns the rise and fall time in seconds.

Example :TRIGger:RISEFall:TIME 4.00E-5  
 Sets the trigger rise & fall to 40.0us.

:TRIGger:VIDeo:TYPe (Set) →  
 → (Query)

Description Sets or queries the video trigger type.

Syntax :TRIGger:VIDeo:TYPe {NTSC | PAL | SECam | EDTV480P | EDTV576P | HDTV720P | HDTV1080I | HDTV1080P | ? }

Related commands :TRIGger:TYPe

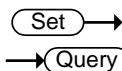
Parameter	NTSC	NTSC
	PAL	PAL
	SECam	SECAM

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 video trigger to NTSC.

**:TRIGger:VIDeo:FIELD**



**Description** Sets or queries the video trigger field.

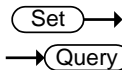
**Syntax** :TRIGger:VIDeo:FIELD { FIELD1 | FIELD2 | ALLFields | ALLLines | ? }

**Related commands** :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.



**:TRIGger:VIDeo:LINE**

**Description** Sets or queries the video trigger line.

**Syntax** :TRIGger:VIDeo:LINE {<NR1> | ?}

**Related commands** :TRIGger:TYPe

Parameter	<NR1>	Video line
Return parameter	<NR3>	Returns the video trigger line.

Example :TRIGger:VIDeo:LINE 1  
Sets the video trigger to line 1.

:TRIGger:VIDeo:POLarity (Set) →  
→ (Query)

Description	Sets or queries the video trigger polarity.
Syntax	:TRIGger:VIDeo:POLarity { POSitive   NEGative   ? }

Related commands :TRIGger:TYPe

Parameter	POSitive	Positive polarity
	NEGative	Negative polarity

Return parameter Returns the video trigger polarity.

Example :TRIGger:VIDeo:POLarity POSitive  
Sets the video trigger polarity to positive.

:TRIGger:PULSe:WHEn (Set) →  
→ (Query)

Description	Sets or queries the pulse width trigger conditions.
Syntax	:TRIGger:PULSe:WHEn { MOREthan   LESSthan   EQual   UNEQual   ? }

Related commands :TRIGger:TYPe  
:TRIGger:PULSe:TIME

Parameter	MORE than	>
	LESSthan	<
	EQual	=
	UNEQual	≠

Return parameter Returns the pulse width trigger conditions.


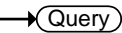
Example :TRIGger:PULSe:WHEn UNEQual  
 Sets the trigger pulse width conditions to not equal to (≠).

:TRIGger:PULSe:TIME 



Description	Sets or queries the pulse width time.	
Syntax	:TRIGger:PULSe:TIME {<NRf>   ?}	
Related commands	:TRIGger:TYPe :TRIGger:PULSe:WHEn	
Parameter	<NRf>	Pulse width time (4ns~10s)
Return parameter	<NR3>	Returns the pulse width time in seconds.

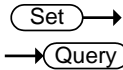
Example :TRIGger:PULSe:TIME 4.00E-5  
 Sets the trigger pulse width to 40.0us.

:TRIGger:TIMEOut:WHEn 



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 timeout condition to low.



**:TRIGger:TIMEOut:TIMER**

Description	Sets or returns timeout trigger time.	
Syntax	:TRIGger:TIMEOut:TIMER {<NRf>   ? }	
Related commands	:TRIGger:TIMEOut:WHEn	
Parameter	<NRf>	Timeout time. (4nS to 10S).
Return parameter	Returns the timeout time as <NR3>.	
Example	:TRIGger:TIMEOut:TIMER? 8.960e-05	

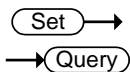
**:TRIGger:STATe**



Description	Returns the current state of the triggering system.	
Syntax	:TRIGger:STATe?	
Return parameter	*ARMED	Indicates that the oscilloscope is acquiring pretrigger information.
	*AUTO	Indicates that the oscilloscope is in the automatic mode and acquires data even in the absence of a trigger.
	*READY	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:STATe?  
 AUTO  
 The trigger is in auto mode.

**:TRIGger:EXTERnAl:PROBE:TYPE**

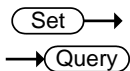


Description	Sets or queries the external probe type.	
Syntax	:TRIGger:EXTERnAl:PROBE:TYPE { VOLTage   CURRent   ? }	
Related commands	:TRIGger:EXTERnAl:PROBE:RATio	
Parameter	VOLTage	Voltage
	CURRent	Current

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>   ?}	
Related commands	:TRIGger:EXTERnAl:PROBE:TYPE	
Parameter	<NRf>	External probe attenuation factor.
Return parameter	<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	I2C	I <sup>2</sup> C mode
	UART	UART mode
	CAN	CAN mode
	LIN	LIN mode

Example :TRIGger:BUS:TYPe?  
UART

:TRIGger:BUS:THReshold:CH<x> (Set) →  
→ (Query)

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

Syntax :TRIGger:BUS:THReshold:CH<X> {<NR3> | ?}

<X>	CH1 ~ CH4
<NR3>	Threshold level

Return Parameter <NR3> Returns the threshold level

Example :TRIGger:BUS:THReshold:CH1 1  
Sets the CH1 threshold to 1V.

:TRIGger:BUS:B1:I2C:CONDition (Set) →  
→ (Query)

Description Sets or queries the I<sup>2</sup>C trigger conditions.

Syntax :TRIGger:BUS:B1:I2C:CONDition  
{START | STOP | REPEATstart | ACKMISS | ADDRess | DATA | ADDRANDDATA | ? }

Parameter	START	Set Start as the I <sup>2</sup> C trigger condition.
	STOP	Set Stop as the I <sup>2</sup> C trigger condition.
	REPEATstart	Set Repeat of Start as the I <sup>2</sup> C trigger condition.

ACKMISS	Set Missing Acknowledgement as the I <sup>2</sup> C trigger condition.
ADDRess	Set Address as the I <sup>2</sup> C trigger condition.
DATA	Set Data as the I <sup>2</sup> C trigger condition.
ADDRANDDATA	Set Address and Data as the I <sup>2</sup> C trigger condition.

Return parameter Returns the I<sup>2</sup>C bus trigger condition.

Example :TRIGger:BUS:B1:I2C:CONDition ADDRess  
Set Address as the I2C trigger condition.

:TRIGger:BUS:B1:I2C:ADDRess:MODE  

Description Sets or queries the I<sup>2</sup>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 addressing
	ADDR10	10 bit addressing

Return Parameter	0	7 bit addressing
	1	10 bit addressing

Example :TRIGger:BUS:B1:I2C:ADDRess:MODE?  
0  
The addressing mode is currently set to 7 bits.

:TRIGger:BUS:B1:I2C:ADDRess:TYPE  

Description Sets the I<sup>2</sup>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<sup>2</sup>C bus address value when the I<sup>2</sup>C bus is set to trigger on Address or Address/Data.

Syntax :TRIGger:BUS:B1:I2C:ADDRess:VALue {<string> | ?}

Related commands :TRIGger:BUS:B1:I2C:ADDRess:MODE

Parameter	<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

:TRIGger:BUS:B1:I2C:ADDRess:DIRection  

Description Sets or queries the address bit as read write or don't care.

Note This setting only applies when the I<sup>2</sup>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: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.

:TRIGger:BUS:B1:I2C:DATA:SIZE  

Description Sets or queries the data size in bytes for the I<sup>2</sup>C bus.

Note This setting only applies when the I<sup>2</sup>C trigger is set to trigger on Data or Address/Data

Syntax :TRIGger:BUS:B1:I2C:DATA:SIZE {<NR1> | ? }

Related commands :TRIGger:BUS:B1:I2C:CONDition

Parameter	<NR1>	Number of data bytes (1 to 5).
-----------	-------	--------------------------------

Return parameter	<NR1>	Returns the number of bytes.
------------------	-------	------------------------------

Example :TRIGger:BUS:B1:I2C:DATA:SIZE 3  
Sets the number of bytes to 3.

:TRIGger:BUS:B1:I2C:DATA:VALue (Set) →  
← (Query)

Description Sets or queries the triggering data value for the I<sup>2</sup>C bus when the I<sup>2</sup>C bus is set to trigger on Data or Address/Data.

Syntax :TRIGger:BUS:B1:I2C:DATA:VALue {<string> | ? }

Related commands :TRIGger:BUS:B1:I2C:DATA:SIZE

Parameter	<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:B1:I2C:DATA:VALue "1x1x0101"  
Sets the value to XXX0101

Example 2 :TRIGger:BUS:B1:I2C:DATA:VALue?  
1X1X0101

:TRIGger:BUS:B1:UART:CONDition (Set) →  
← (Query)

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.
RXPARTyerr	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.
TXPARTyerr	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> | ?}

**Related commands** :TRIGger:BUS:B1:UART:CONDition

**Parameter** <NR1> Number of bytes (1 to 10).

**Return parameter** <NR1> Returns the number of bytes.

**Example** :TRIGger:BUS:B1:UART:RX:DATA:SIZE 5  
Sets the number of bytes to 5.

:TRIGger:BUS:B1:UART:RX:DATA:VALue 


**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> | ? }

**Related commands** :TRIGger:BUS:B1:UART:RX:DATA:SIZE

<b>Parameter</b>	<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** :TRIGger:BUS:B1:UART:CONDition RXDATA  
 :TRIGger:BUS:B1:UART:RX:DATA:SIZE 1  
 :TRIGger:BUS:B1:UART:RX:DATA:VALue "1x1x0101"  
 Sets the value to 1x1x0101

**Example 2** :TRIGger:BUS:B1:UART:RX:DATA:VALue?  
 1X1X0101

: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 trigger is set to trigger on Tx Data

**Syntax** :TRIGger:BUS:B1:UART:TX:DATA:SIZE {<NR1> | ? }

**Related commands** :TRIGger:BUS:B1:UART:CONDition

Parameter	<code>&lt;NR1&gt;</code>	Number of bytes (1 to 10).
-----------	--------------------------	----------------------------

Return parameter	<code>&lt;NR1&gt;</code>	Returns the number of bytes.
------------------	--------------------------	------------------------------

Example	:TRIGger:BUS:B1:UART:TX:DATA:SIZE 5 Sets the number of bytes to 5.	
---------	---	--

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>   ? }	
--------	--	--

Related commands	:TRIGger:BUS:B1:UART:TX:DATA:SIZE	
------------------	-----------------------------------	--

Parameter	<code>&lt;sting&gt;</code>	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	
-----------	---	--

Description	Sets or returns the CAN trigger condition.	
-------------	--	--

Syntax	:TRIGger:BUS:B1:CAN:CONDition {SOF FRAMEType IDentifier DATA IDANDDATA EOF ACKMISS STUFFERR ?}	
Parameter/ Return parameter	SOF	Triggers on a start of frame
	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 start of frame.

Example2 :TRIGger:BUS:B1:CAN:CONDition?  
>SOF

:TRIGger:BUS:B1:CAN:FRAMEType 


Description Sets or returns the frame type for a CAN FRAMEType trigger.

Syntax :TRIGger:BUS:B1:CAN:FRAMEType  
{DATA|REMote|ERRor|OVERLoad|?}

Parameter/ Return parameter	DATA	Sets the frame type to data frame
	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 →
		→ Query
<b>:TRIGger:BUS:B1:CAN:IDentifier:MODE</b>		
Description	Sets or returns the CAN identifier mode for the bus.	
Syntax	:TRIGger:BUS:B1:CAN:IDentifier:MODE {STANDard EXTended ?}	
Parameter/ Return parameter	STANDard	Standard addressing mode
	EXTended	Extended addressing mode
Example	:TRIGger:BUS:B1:CAN:IDentifier:MODE? >STANDARD  Returns the addressing mode.	

		Set →
		→ Query
<b>:TRIGger:BUS:B1:CAN:IDentifier:VALue</b>		
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:BUS:B1:CAN:IDentifier:VALue {<string> ?}	
Related Commands	:TRIGger:BUS:B1:CAN:IDentifier:MODE	
Parameter/ Return parameter	<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:B1:CAN:CONDition ID
              :TRIGger:BUS:B1:CAN:IDentifier:MODE STANDARD
              :TRIGger:BUS:B1:CAN:IDentifier:VALue
              "01100X1X01X"
              :TRIGger:BUS:B1:CAN:IDentifier:VALue?
              >01100X1X01X
```

Set →

```
:TRIGger:BUS:B1:CAN:IDentifier: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|?}

<b>Parameter/ Return parameter</b>	READ	Sets read as the data direction
	WRITE	Sets write as the data direction
	NOCARE	Sets either as the data direction

**Example1** :TRIGger:BUS:B1:CAN:IDentifier:DIRection? >WRITE

**Example2** :TRIGger:BUS:B1:CAN:IDentifier:DIRection READ  
:TRIGger:BUS:B1:CAN:IDentifier:DIRection? > READ

Set →

```
:TRIGger:BUS:B1:CAN:DATA:QUALifier → Query
```

**Description** Sets or returns the CAN data qualifier.  
Note: Only applicable when the triggering condition is set to DATA or IDANDDATA.

**Syntax** :TRIGger:BUS:B1:CAN:DATA:QUALifier {LESSthan|MOREthan|Equal|UNEQual|LESSEQual|MOREEQual|?}

<b>Parameter/ Return parameter</b>	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:B1:CAN:DATA:QUALifier?  
 >EQUAL  
 :TRIGger:BUS:B1:CAN:DATA:QUALifier MOREthan  
 :TRIGger:BUS:B1:CAN:DATA:QUALifier?  
 >MOREthan


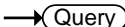
:TRIGger:BUS:B1:CAN:DATA:SIZE (Set) →  
 → (Query)

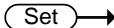
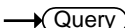
Description Sets or returns the length of the data string in bytes for a CAN trigger.  
 Note: Only applicable when the condition is set to DATA or IDANDDATA.

Syntax :TRIGger:BUS:B1:CAN:DATA:SIZE {<NR1>|?}

Parameter/ Return parameter <NR1> 1~8 (bytes)

Example :TRIGger:BUS:B1:CAN:DATA:SIZE?  
 >1  
 :TRIGger:BUS:B1:CAN:DATA:SIZE 2  
 :TRIGger:BUS:B1:CAN:DATA:SIZE?  
 >2

		 
<b>:TRIGger:BUS:B1:CAN:DATA:VALue</b>		
Description	Sets or returns the binary data string to be used for a CAN trigger.  Note: Only applicable when the condition is set to DATA or IDANDDATA.	
Related Commands	:TRIGger:BUS:B1:CAN:DATA:SIZE	
Syntax	:TRIGger:BUS:B1:CAN:DATA:VALue {<string> ?}	
Parameter/Return parameter	<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	<pre>:TRIGger:BUS:B1:CAN:DATA:SIZE 1 :TRIGger:BUS:B1:CAN:DATA:VALue "01010X1X" :TRIGger:BUS:B1:CAN:DATA:VALue? &gt;01010X1X</pre>	

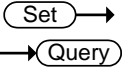
		 
<b>:TRIGger:BUS:B1:LIN:CONDition</b>		
Description	Sets or returns the LIN trigger condition.	
Syntax	:TRIGger:BUS:B1:LIN:CONDition {SYNCField IDentifier DATA IDANDDATA WAKEup SLEEP ERRor ?}	
Parameter/Return parameter	SYNCField	Sets the LIN trigger condition to the sync field.
	IDentifier	Sets the LIN 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:B1:LIN:DATA:QUALifier 

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 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:B1:LIN:DATA:QUALifier?  
>EQUAL  
:TRIGger:BUS:B1:LIN:DATA:QUALifier MOREthan  
:TRIGger:BUS:B1:LIN:DATA:QUALifier?  
>MORETHAN

:TRIGger:BUS:B1:LIN:DATA:SIZE (Set) →  
← (Query)

Description Sets or returns the length of the data string in bytes for the LIN trigger.  
Note: Only applicable when the condition is set to DATA or IDANDDATA.

Syntax :TRIGger:BUS:B1:LIN:DATA:SIZE {<NR1>|?}

Parameter/ Return parameter <NR1> 1~8 (bytes)

Example :TRIGger:BUS:B1:LIN:DATA:SIZE?  
>1  
:TRIGger:BUS:B1:LIN:DATA:SIZE 2  
:TRIGger:BUS:B1:LIN:DATA:SIZE?  
>2

:TRIGger:BUS:B1:LIN:DATA:VALue (Set) →  
← (Query)

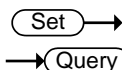
Description Sets or returns the binary data string to be used for the LIN trigger.  
Note: Only applicable when the condition is set to DATA or IDANDDATA.

Related Commands :TRIGger:BUS:B1:LIN:DATA:SIZE

Syntax :TRIGger:BUS:B1:LIN:DATA:VALue {<string>|?}

Parameter/ Return parameter	<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:B1:LIN:DATA:SIZE 1  
                   :TRIGger:BUS:B1:LIN:DATA:VALue "01010X1X"  
                   :TRIGger:BUS:B1:LIN:DATA:VALue?  
                   >01010X1X



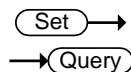
**:TRIGger:BUS:B1:LIN:ERRTYPE**

Description       Sets or returns the error type be used for the LIN trigger.

Syntax            :TRIGger:BUS:B1:LIN:ERRTYPE  
                   {SYNC|PARItY|CHecksum|?}

Parameter/ Return parameter	SYNC	Sets the LIN error type to SYNC.
	PARItY	Sets the LIN error type to parity.
	CHecksum	Sets the LIN error type to checksum.

Example           :TRIGger:BUS:B1:LIN:ERRTYPE?  
                   >SYNC  
                   :TRIGger:BUS:B1:LIN:ERRTYPE CHECKSUM  
                   :TRIGger:BUS:B1:LIN:ERRTYPE?  
                   >CHECKSUM



**:TRIGger:ALTErnate**

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.	


Set →  
 → Query

**:TRIGger:BUS:B1:SPI:CONDition**

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 triggering condition.	
Example	:TRIGger:BUS:B1:SPI:CONDition MISO Sets the SPI bus to trigger on MISO.	

Set →  
 → Query

**:TRIGger:BUS:B1:SPI:DATA:SIZE**

Description	Sets or queries 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>   ?}	
Related commands	:TRIGger:BUS:B1:SPI:CONDition	

Parameter	<NR1>	Number of words (1 to 32).
Return parameter	<NR1>	Returns the number of words.
Example	:TRIGger:BUS:B1:SPI:DATA:SIZE 10 Sets the number of words to 10.	
	<div style="text-align: right;"> </div>	
	:TRIGger:BUS:B1:SPI:DATA:MISO:VALue	
Description	Sets or queries the triggering data value for the SPI bus when the bus is set to trigger on MISO or MISO/MOSI.	
Syntax	:TRIGger:BUS:B1:SPI:DATA:MISO:VALue {<string>   ? }	
Related commands	:TRIGger:BUS:B1:SPI:DATA:SIZE	
Parameter	<sting>	<p>The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string".</p> <p>x = don't care 1 = binary 1 0 = binary 0</p>
Return Parameter	Returns the data value.	
Example1	:TRIGger:BUS:B1:SPI:CONDition MISO :TRIGger:BUS:B1:SPI:DATA:SIZE 2 :TRIGger:BUS:B1:SPI:DATA:MISO:VALue "1x1x0101" Sets the value to 1x1x0101	
Example 2	:TRIGger:BUS:B1:SPI:DATA:MISO:VALue? 1X1X0101	

:TRIGger:BUS:B1:SPI:DATA:MOSI:VALue (Set) →  
→ (Query)

**Description** Sets or queries the triggering data value for the SPI bus when the bus is set to trigger on MOSI or MISO/MOSI.

**Syntax** :TRIGger:BUS:B1:SPI:DATA:MOSI:VALue {<string> | ? }

**Related commands** :TRIGger:BUS:B1:SPI:DATA:SIZE

<b>Parameter</b>	<code>&lt;string&gt;</code>	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 MOSI  
:TRIGger:BUS:B1:SPI:DATA:SIZE 2  
:TRIGger:BUS:B1:SPI:DATA:MOSI:VALue "1x1x0101"  
Sets the value to 1x1x0101

**Example2** :TRIGger:BUS:B1:SPI:DATA:MOSI:VALue?  
1X1X0101

:TRIGger:BUS:B1:LIN:IDentifier:VALue (Set) →  
→ (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:BUS:B1:LIN:IDentifier:VALue {<string>|?}

---


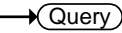
Parameter/ Return parameter	<string>	The size of the string depends on the data size setting. The string must be enclosed in double quotes, "string".  String contents: x = don't care 1 = binary 1 0 = binary 0
--------------------------------	----------	--


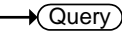
---

Example

```
:TRIGger:BUS:B1:LIN:CONDition ID  
:TRIGger:BUS:B1:LIN:IDentifier:VALue "00X1X01X"  
:TRIGger:BUS:B1:LIN:IDentifier:VALue?  
>01100X1X01X  
:TRIGger:LOGic:INPut:CLOCK:SOURce
```

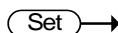
## System Commands

		
<code>:SYSTem:LOCK</code>		
<hr/>		
Description	Turns the panel lock on off.	
Syntax	<code>:SYSTem:LOCK {OFF   ON   ? }</code>	
Parameter	OFF	System lock off
	ON	System lock on
Return parameter	Returns the status of the panel lock (ON, OFF).	
Example	<code>:SYSTem:LOCK ON</code> Turns the panel lock on.	

		
<code>:SYSTem:ERRor</code>		
<hr/>		
Description	Queries the error queue. See the appendix on page 320 for details.	
Syntax	<code>:SYSTem:ERRor?</code>	
Return parameter	Returns the last message in the error queue.	
Example	<code>:SYSTem:ERRor?</code> +0, "No error."	

## Save/Recall Commands

### :RECALL:SETUp



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

### :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>	
Parameter	n	1~20 (Wave1~wave20)
	<file page>	Filename in file path. Example: "Disk:/xxx.LSF", "USB:/xxx.LSF", "Disk:/xxx.CSV", "USB:/xxx.CSV"
	<X>	1,2,3,4 (REF1, REF2, REF3, REF4)
Example	<p>:RECALL:WAVEform W1, REF1</p> <p>Recalls the waveform stored in Wave1 to reference 1.</p>	

**: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”)}	
Related commands	:SAVe:IMAGe:FILEFormat :SAVe:IMAGe:INKSaver	
Parameter	xxx.PNG or xxx.BMP	File name (8 characters max)
Example	:SAVe:IMAGe “Disk:/pic1.PNG” Saves a screen image named pic1.png to the root directory (Disk:/) 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.	

**:SAVe:IMAGe:FILEFormat**

Set →

→ Query

Description	Sets the file format for image.	
Syntax	:SAVe:IMAGe:FILEFormat {PNG   BMP   ?}	
Related commands	:SAVe:IMAGe :SAVe:IMAGe:INKSaver	
Parameter	PNG BMP	Sets the file format to PNG Sets the file format to BMP
Return parameter	Returns the file format (PNG, BMP).	
Example	:SAVe:IMAGe:FILEFormat PNG Sets the image file format to PNG.	

**:SAVe:IMAGe:INKSaver**

Set →

→ Query

Description	Turns Ink Saver on or off.	
Syntax	:SAVe:IMAGe:INKSaver {OFF   ON  ?}	
Related commands	:SAVe:IMAGe :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	:SAVe:SETUp {<file path> ("Disk:/xxx.SET", "USB:/xxx.SET)   S1~S20}	
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:WAVEform**

Description	Saves a waveform to internal memory or to a designated file path.	
Related commands	:SAVe:WAVEform:FILEFormat	

Syntax	:SAVE:WAVEform {CH1~REF4, REF<X> }   {CH1~REF4, W1~W20}   {CH1~ALL, file path}	
Parameter	CH1~REF4,	CH1~CH4, Math, REF1~4
	<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 channel1 waveform to REF2.

Example 2 :SAVE:WAVEform ALL, "Disk:/ALL001"  
Creates a folder which named "ALL001" and saves all displayed waveform to the "ALL001" directory with LSF format

Example 3 :SAVE:WAVEform ALL, "Disk:/ALL002.CSV"  
Save the all channels waveform to root directory (Disk:/) of the internal flash disk with CSV format.

Example 4 :SAVE:WAVEform CH2, "Disk:/DS0003.LSF"  
Save the channel 2's waveform to root directory (Disk:/) of the internal flash disk with LSF format.

Set →

→ Query

**:SAVE:WAVEform:FILEFormat**

Description Sets the waveform savefile format.

Syntax :SAVE:WAVEform:FILEFormat {LSF | DCSV | FCSV | ?}

Parameter	LSF	Sets the file format to the MPO-2000'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 Command

:ETHERnet:DNS:IPADdress Set →  
→ Query

Description	Sets or queries ETHERnet DNS IP address.
Syntax	:ETHERnet:DNS:IPADdress {string } :ETHERnet:DNS:IPADdress?
Parameter	string IP number
Example	:ETHERnet:DNS:IPADdress "172.16.1.251" :ETHERnet:DNS:IPADdress? 172.16.1.251

:ETHERnet:DOMAINname Set →  
→ Query

Description	Sets or queries ETHERnet domain name.
Syntax	:ETHERnet:DOMAINname {string} :ETHERnet:DOMAINname?
Parameter	string IP number
Example	:ETHERnet:DOMAINname "GW01" :ETHERnet:DOMAINname? GW01

:ETHERnet:ENET:ADDRess? Set →  
→ Query

Description	Queries ETHERnet MAC address.
Syntax	:ETHERnet:ENET:ADDRess?
Example	:ETHERnet:ENET:ADDRess? 01:02:03:04

 →  
 → 

**:ETHERnet:GATEWay:IPADdress**

Description	Sets or queries ETHERnet Gateway IP.
Syntax	:ETHERnet:GATEWay:IPADdress{string} :ETHERnet:GATEWay:IPADdress?
Parameter	string IP number
Example	:ETHERnet:GATEWay:IPADdress "172.16.0.252" :ETHERnet:GATEWay:IPADdress? 172.16.0.252

 →  
 → 

**:ETHERnet:NAME**

Description	Sets or queries ETHERnet Name.
Syntax	:ETHERnet:NAME {string} :ETHERnet:NAME?
Parameter	string Instrument name
Example	:ETHERnet:NAME "GDS3000A" :ETHERnet:NAME? GDS3000A

 →  
 → 

**:ETHERnet:PASSWord**

Description	Sets or queries ETHERnet Password.
Syntax	ETHERnet:PASSWord {string} :ETHERnet:PASSWord?
Parameter	string Password
Example	:ETHERnet:PASSWord "GW1234" :ETHERnet:PASSWord? GW1234

Set →  
→ Query

---

<b>:ETHERnet:SUBNETMask</b>	
Description	Sets or queries ETHERnet Subnet mask.
Syntax	ETHERnet:SUBNETMask {string} :ETHERnet:SUBNETMask?
Parameter	string      Mask IP number
Example	:ETHERnet:SUBNETMask "255.255.126.0" :ETHERnet:SUBNETMask? 255.255.126.0

Set →  
→ Query

---

<b>:ETHERnet:DHCP</b>	
Description	Sets or queries the DHCP settings.
Syntax	:ETHERnet:DHCP { OFF   ON   ? }
Parameter	ON              Turns DHCP on. OFF             Turns DHCP off.
Example	:ETHERnet:DHCP ON Turns DHCP on.

## Time Command

---

:DATE



Description	Sets the system date and time.
Syntax	:DATE {<string>}
Parameter	<string> "YYYYMMDDhhmmss" Where: YYYY: year MM: month DD: day hh: hour mm: minute ss: second
Example	:DATE "20140802142830" Sets the time and date as: Year: 2014, Month: 08, Day: 02, Hour: 14 (2PM), Minute: 28, Second: 30.

## Bus Decode Commands

**:BUS1** → Query

Description	Returns the supported BUS types.
Syntax	:BUS1?
Return Parameter	Returns the supported bus types.
Example	BUS1? I2C, SPI, UART, Parallel, CAN, LIN

Set →

**:BUS1:STATE** → Query

Description	Sets or queries the state of the bus.	
Syntax	:BUS1:STATE { OFF   ON   ? }	
Related commands	:BUS1:TYPE	
Parameter/Return parameter	OFF	Turns the bus off.
	ON	Turns the bus on.
Example	:BUS1:STATE ON Turns the bus on.	

Set →

**:BUS1:TYPE** → Query

Description	Sets or queries the type of bus.	
Syntax	:BUS1:TYPE { UART   I2C   SPI   Parallel   CAN   LIN   ? }	
Related commands	:BUS1:STATE	
Parameter/Return parameter	UART	Set to UART mode.
	I2C	Sets to I2C mode.

	SPI	Set to SPI mode
	Parallel	Set to Parallel mode
	CAN	Sets to CAN mode.
	LIN	Sets to LIN mode.

Example :BUS1:TYPe LIN  
Sets the bus to LIN mode.

**:BUS1:INPut**

Set →

→ Query

Description	Sets or returns the bus source.	
Syntax	:BUS1:INPut {ANALog   ?}	
Parameter/Return parameter	ANALog	Sets the bus source as analog inputs.
Example1	:BUS1:INPut? >ANALOG	

Set →

→ Query

**:BUS1:I2C:ADDRes:RWINClude**

Description	Sets or queries whether the read/write bit is included in the I <sup>2</sup> C address.	
Syntax	:BUS1:I2C:ADDRes: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:ADDRes:RWINClude ON Includes the R/W bit in the I <sup>2</sup> C address.	

Set →  
→ Query

<b>:BUS1:I2C:SCLK:SOURce</b>	
Description	Sets or queries which channel is used for the I <sup>2</sup> C SCLK source.
Syntax	:BUS1:I2C:SCLK:SOURce { CH1   CH2   CH3   CH4 ? }
Parameter/Return parameter	CH1 to CH4 Analog channels 1 ~ 4.
Example	:BUS1:I2C:SCLK:SOURce CH1 Sets channel 1 as the SCLK source.

Set →  
→ Query

<b>:BUS1:I2C:SDA:SOURce</b>	
Description	Sets or queries which channel is used for the I <sup>2</sup> C SDA source.
Syntax	:BUS1:I2C:SDA:SOURce{ CH1   CH2   CH3   CH4  ? }
Parameter/Return parameter	CH1 to CH4 Analog channels 1 ~ 4.
Example	:BUS1:I2C:SDA:SOURce CH1 Sets channel 1 as the SDA source.

Set →  
→ Query

<b>:BUS1:UART:BITRate</b>	
Description	Sets or queries the UART bit rate.
Syntax	:BUS1:UART:BITRate { <NR1>   ? }
Parameter/Return parameter	<NR1> UART bit rate in bps
Example	:BUS1:UART:BITRate? >2400 :BUS1:UART:BITRate 50 :BUS1:UART:BITRate? >50

Set →  
 → Query

**:BUS1:UART:DATABits**

**Description**      Sets or queries the number UART data for bus 1.

**Syntax**            :BUS1:UART:DATABits { 5 | 6 | 7 | 8 | 9 | ? }

<b>Parameter/Return parameter</b>	5	5 data bits in the UART frame.
	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  
 Sets the UART frame to 7 bits.

Set →  
 → Query

**:BUS1:UART:PARItY**

**Description**      Sets or queries the UART bus parity.

**Syntax**            :BUS1:UART:PARItY { <NR1> | ? }

<b>Parameter/Return parameter</b>	<NR1>	0: None
		1: Odd parity
		2: Even parity

**Example**            :BUS1:UART:PARItY 1  
 Sets the parity to odd.

Set →  
 → Query

**:BUS1:UART:PACKEt**

**Description**      Sets or queries the UART packet setting.

**Syntax**            :BUS1:UART:PACKEt { <NR1> | ? }

<b>Parameter/Return parameter</b>	<NR1>	0: Off
		1: On

**Example**            :BUS1:UART:PACKEt 1  
 Turns UART packets on.

**:BUS1:UART:POLARity** (Set) →  
→ (Query)

Description	Sets or returns the UART polarity.	
Syntax	:BUS1:UART: POLARity {NORMAl INVerted} :BUS1:UART: POLARity?	
Parameter	NORMAl	Sets normal UART polarity.
	INVerted	Sets inverted UART polarity.
Example	:BUS1:UART:POLARity NORMAl :BUS1:UART:POLARity? NORMAL	

**:BUS1:UART:EOFPAcket** (Set) →  
→ (Query)

Description	Sets or queries the EOF character for the UART packet setting.	
Syntax	:BUS1:UART:EOFPAcket <NR1>	
Parameter/Return parameter	<NR1>	0: NULL 1: LF (line feed) 2: CR (carriage return) 3: SP (space character) 4: FF
Example	:BUS1:UART:EOFPAcket 2 Sets the OEF character to CR.	

**:BUS1:UART:TX:SOURce** (Set) →  
→ (Query)

Description	Sets or queries which channel is used for the UART Tx source.	
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.

:BUS1:UART:RX:SOURce Set →  
← 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 parameter OFF Off, no Rx source  
CH1 to CH4 Analog channels CH1 to CH4

Example :BUS1:UART:RX:SOURce CH1  
Sets channel 1 as the Rx source.

:BUS1:DISPlay:FORMAt Set →  
← 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 Binary format  
HEXadecimal Hexadecimal format

Example :BUS1:DISPlay:FORMAt BINary  
Sets the display format to binary.

:LISTer:DATA ← Query

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** (Set) →  
→ (Query)

**Description** Sets or returns the CAN input source.

**Syntax** :BUS1:CAN:SOURce { CH1 | CH2 | CH3 | CH4 | ? }

<b>Parameter/Return parameter</b>	CH1 ~ CH4	Analog channel source
-----------------------------------	-----------	-----------------------

**Example** :BUS1:CAN:SOURCE?  
>CH1  
  
Returns the CAN source.

**:BUS1:CAN:PROBe** (Set) →  
→ (Query)

**Description** Sets or returns the signal type of the CAN bus.

**Syntax** :BUS1:CAN:PROBe {CANH | CANL | TX | RX | ? }

<b>Parameter/Return parameter</b>	CANH	CAN-High
	CANL	CAN-Low
	TX	Transmit
	RX	Receive

**Example** :BUS1:CAN:PROBe?  
>CANH  
  
:BUS1:CAN:PROBe CANL  
:BUS1:CAN:PROBe?  
>CANL

**:BUS1:CAN:SAMPLEpoint** → (Query)

**Description** Returns the sample point of the CAN 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.

Set →

:BUS1:CAN:BITRate

→ Query

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

Syntax :BUS1:CAN:BITRate  
{RATE10K|RATE20K|RATE50K|RATE125K|RATE250K|  
RATE500K|RATE800K|RATE1M | <NR1> | ?}

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
	<NR1>	CAN bit rate in bps

Example :BUS1:CAN:BITRate?  
>1000000  
  
:BUS1:CAN:BITRate rate800k  
:BUS1:CAN:BITRate?  
>800000  
  
:BUS1:CAN:BITRate 25000  
:BUS1:CAN:BITRate?  
>25000

Set →

:BUS1:LIN:BITRate

→ Query

---

Description	Sets or returns the bit rate of the LIN bus.	
Syntax	:BUS1:LIN:BITRate {<NR1>   ?}	
Parameter/Return parameter	<NR1>	LIN bit rate in bps.
Example	:BUS1:LIN:BITRate 9600 Sets the LIN bit rate to 9600bps.	

Set →

:BUS1:LIN:IDFORmat

→ Query

---

Description	Sets or returns the LIN ID format.	
Syntax	:BUS1:LIN:IDFORmat {NOPARity PARity ?}	
Parameter/Return parameter	NOPARity	Don't include parity bits with Id.
	PARity	Include parity bits with Id.
Example	:BUS1:LIN:IDFORmat? NOPARITY Returns the ID format.	

Set →

:BUS1:LIN:POLARity

→ Query

---

Description	Sets or returns the LIN polarity.	
Syntax	:BUS1:LIN:POLARity {NORMal INVerted ?}	
Parameter/Return parameter	NORMal	Normal LIN polarity
	INVerted	Inverted LIN polarity
Example	:BUS1:LIN:POLARity? NORMAL Returns the LIN polarity.	

**:BUS1:LIN:SAMPLEpoint** → Query

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.

Set →

**:BUS1:LIN:SOURce** → Query

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 Returns the LIN source.

Set →

**:BUS1:SPI:SCLK:POLARity** → Query

Description	Sets or queries the polarity of the SCLK line for the SPI bus.
Syntax	:BUS1:SPI:SCLK:POLARity { FALL   RISE   ? }
Parameter/Return parameter	FALL Sets the polarity to falling edge. RISE Sets the polarity to rising edge.
Example	:BUS1:SPI:SCLK:POLARity FALL Sets the polarity to falling edge.

Set →  
 → Query

**:BUS1:SPI:SS:POLARity**

Description	Sets or queries the polarity of the SS line for the SPI bus.	
Syntax	:BUS1:SPI:SS:POLARity { LOW   HIGH   ? }	
Parameter/Return parameter	LOW	Active low polarity
	HIGH	Active high polarity
Example	:BUS1:SPI:SS:POLARity LOW Sets the SS line to active low.	

Set →  
 → Query

**:BUS1:SPI:WORDSize**

Description	Sets the number of bits per word for the SPI bus.	
Syntax	:BUS1:SPI:WORDSize {<NR1>   ? }	
Parameter/Return parameter	<NR1>	Bits per word (4~32)
Example	:BUS1:SPI:WORDSize 4 Sets the word size to 4 bits per word.	

Set →  
 → Query

**:BUS1:SPI:BITORder**

Description	Sets or queries the bit order for the SPI bus.	
Syntax	:BUS1:SPI:BITORder {<NR1>   ? }	
Parameter/Return parameter	<NR1>	0: MSB bit first 1: LSB bit first
Example	:BUS1:SPI:BITORder? 0 The bit order is currently set as MSB bit first.	

→  
 →

**:BUS1:SPI:SCLK:SOURce**

Description	Sets or queries which channel is used for the SPI SCLK source.
Syntax	:BUS1:SPI:SCLK:SOURce { CH1   CH2   CH3   CH4   D0   D1   D2   D3   D4   D5   D6   D7   D8   D9   D10   D11   D12   D13   D14   D15   ? }
Parameter/Return parameter	CH1 to CH4 Analog channels CH1 to CH4 D0 to D15 Digital channels D0 to D15
Example	:BUS1:SPI:SCLK:SOURce CH1 Sets channel 1 as the SPI SCLK source.

→  
 →

**:BUS1:SPI:SS:SOURce**

Description	Sets or queries which channel is used for the SPI SS source.
Syntax	:BUS1:SPI:SS:SOURce { CH1   CH2   CH3   CH4   D0   D1   D2   D3   D4   D5   D6   D7   D8   D9   D10   D11   D12   D13   D14   D15   ? }
Parameter/Return parameter	CH1 to CH4 Analog channels CH1 to CH4 D0 to D15 Digital channels D0 to D15
Example	:BUS1:SPI:SS:SOURce CH1 Sets channel 1 as the SPI SS source.

→  
 →

**:BUS1:SPI:MOSI:SOURce**

Description	Sets or queries which channel is used for the SPI MOSI source.
Syntax	:BUS1:SPI:MOSI:SOURce { OFF   CH1   CH2   CH3   CH4   D0   D1   D2   D3   D4   D5   D6   D7   D8   D9   D10   D11   D12   D13   D14   D15   ? }

Parameter/Return parameter	CH1 to CH4	Analog channels CH1 to CH4
	D0 to D15	Digital channels D0 to D15
	OFF	No MOSI source.

Example :BUS1:SPI:MOSI:SOURce CH1  
Sets channel 1 as the SPI MOSI source.

Set →

:BUS1:SPI:MISO:SOURce

→ Query

Description Sets or queries which channel is used for the SPI MISO source.

Syntax :BUS1:SPI:MISO:SOURce { OFF | CH1 | CH2 | CH3 | CH4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | ? }

Parameter/Return parameter	CH1 to CH4	Analog channels CH1 to CH4
	D0 to D15	Digital channels D0 to D15
	OFF	No MISO source.

Example :BUS1:SPI:MISO:SOURce CH1  
Sets channel CH1 as the SPI MISO source.

Set →

:BUS1:LIN:STANDARD

→ Query

Description Sets or returns the LIN standard.

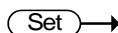
Syntax :BUS1:LIN:STANDARD {V1X|V2X|BOTH|?}

Parameter/Return parameter	V1X	Lin standard version 1.x
	V2X	Lin standard version 2.x
	BOTH	Both standards

Example :BUS1:LIN:STANDARD?  
>BOTH  
Returns the LIN standard.

## Mark Commands

### :MARK



Description	Move to next or previous event mark.
Syntax	:MARK { NEXT   PREVIOUS }
Related commands	:MARK:CREATE :MARK:DELEte
Parameter	NEXT Move to next mark PREVIOUS Move to previous mark
Example	:MARK NEXT Moves to the next event mark.

### :MARK:CREATE



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

### :MARK:DELEte



Description	Deletes the current mark or all the marks on a waveform.
-------------	--

Syntax	:MARK:DELEte { CURRent   ALL }	
Related commands	:MARK	
	: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

Set →

**Description** Copies the search settings to the trigger settings or copies the trigger settings to the search settings.

**Syntax** :SEARCH:COPY {SEARCHtotrigger|TRIGgertosearch}

<b>Parameter</b>	SEARCHtotrigger	Copy the search setting to the trigger settings.
	TRIGgertosearch	Copy the trigger settings to the search settings.

**Example** :SEARCH:COPY SEARCHtotrigger  
Copies the search settings to the trigger settings.

Set →

### :SEARCH:STATE

→ Query

**Description** Sets or queries whether the Search function is on or off.

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

<b>Parameter/Return parameter</b>	OFF	Turn the Search function on.
	ON	Turn the Search function off.

**Example** :SEARCH:STATE ON  
Turn Search on.

### :SEARCH:TOTAL

→ Query

**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

Set →

→ Query

:SEARCH:TRIGger:TYPE

Description Sets or queries the search trigger type.

Syntax :SEARCH:TRIGger:TYPE { EDGe | PULSEWidth | RUNT | RISEFall | FFTPeak | BUS | ? }

Parameter/Return parameter	EDGe	Edge trigger
	PULSEWidth	Pulse width trigger
	RUNT	Runt trigger
	RISEFall	Rise and Fall trigger
	FFTPeak	FFT Peak trigger
	BUS	Bus trigger

Example :SEARCH:TRIGger:TYPE EDGe  
Sets the search trigger to the edge type.

Set →

→ Query



:SEARCH:TRIGger:SOURce

Description Sets or queries the search trigger source.

Syntax :SEARCH:TRIGger:SOURce {CH1 | CH2 |CH3 | CH4|? }

Parameter/Return parameter	CH1 to CH4	Channel 1 to Channel 4
----------------------------	------------	------------------------


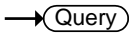
Example :SEARCH:TRIGger:SOURce CH1  
Sets the search trigger source as CH1.

---

**:SEARCH:TRIGger:EDGE:SLOP**

Description	Sets or queries 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 trigger slope.	
Example	:SEARCH:TRIGger:EDGE:SLOP FALL Sets the search trigger slope to falling.	


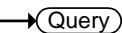

  


---

**:SEARCH:TRIGger:LEVel**

Description	Sets or queries the search trigger level.	
Syntax	:SEARCH:TRIGger:LEVel {TTL   ECL  SETTO50   <NRf>   ?}	
Related commands	:SEARCH:TRIGger:TYPE	
Parameter	<NRf>	Trigger level value
	TTL	Sets the search trigger level to TTL.
	ECL	Sets the search trigger level to ECL.
	SETTO50	Sets the search trigger level to the User level (50% by default).
Return parameter	<NR3>	Returns the trigger.
Example1	:SEARCH:TRIGger:LEVel TTL Sets the search trigger level to TTL.	



**:SEARCH:TRIGger:PULSEWidth:POLarity** 
 →  
 → 

**Description** Sets or queries the pulse width search trigger polarity.

**Syntax** :SEARCH:TRIGger:PULSEWidth:POLarity {POSitive | NEGative | ?}

**Related commands** :SEARCH:TRIGger:TYPe

<b>Parameter</b>	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:TRIGger:RUNT:POLarity** 
 →  
 → 

**Description** Sets or queries the Pulse Runt search trigger polarity.

**Syntax** :SEARCH:TRIGger:RUNT:POLarity {POSitive | NEGative | EITHer | ?}

**Related commands** :SEARCH:TRIGger:TYPe

<b>Parameter</b>	POSitive	Positive polarity
	NEGative	Negative polarity
	EITHer	Positive or negative polarity

**Return parameter** Returns the pulse runt search trigger polarity.

**Example** :SEARCH:TRIGger:RUNT:POLarity POSitive  
Sets the Pulse Runt search trigger polarity to positive.

Set →  
 → Query

**:SEARCH:TRIGger:RISEFall:SLOP**

---

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 Sets the Rise & Fall search trigger slope to rising.	

Set →  
 → Query

**:SEARCH:TRIGger:PULSe:WHEN**

---

Description	Sets or queries the pulse width search trigger conditions.	
Syntax	:SEARCH:TRIGger:PULSe:WHEN { MOREthan   LESSthan   EQual   UNEQual   ? }	
Related commands	:SEARCH:TRIGger:TYPE :SEARCH:TRIGger:PULSe:TIME	
Parameter	MOREthan	>
	LESSthan	<
	EQual	=
	UNEQual	≠
Return parameter	Returns the pulse width search trigger conditions.	
Example	:SEARCH:TRIGger:PULSe:WHEN UNEQual Sets the pulse width search trigger conditions to not equal to (≠).	

Set →  
 → Query

---

**:SEARCH:TRIGger:PULSe:TIME**

Description	Sets or queries the pulse width search trigger time.	
Syntax	:SEARCH:TRIGger:PULSe:TIME {<NRf>   ?}	
Related commands	:SEARCH:TRIGger:TYPe :SEARCH:TRIGger:PULSe:WHEn	
Parameter	<NRf>	Pulse width time (4ns~10s)
Return parameter	<NR3>	Returns the pulse width time in seconds.
Example	:SEARCH:TRIGger:PULSe:TIME 4.00E-5 Sets the pulse width search trigger to 40.0us.	

Set →  
 → Query

---

**:SEARCH:TRIGger:RUNT:WHEn**

Description	Sets or queries the pulse runt search trigger conditions.	
Syntax	:SEARCH:TRIGger:RUNT:WHEn {MOREthan   LESSthan   EQual   UNEQual   ?}	
Related commands	:SEARCH:TRIGger:TYPe :SEARCH:TRIGger:RUNT:TIME	
Parameter	MOREthan	>
	LESSthan	<
	EQual	=
	UNEQual	≠
Return parameter	Returns the pulse runt search trigger conditions.	
Example	:SEARCH:TRIGger:RUNT:WHEn UNEQual Sets the pulse runt search trigger condition to unequal (≠).	

Set →  
 → Query

**:SEARCH:TRIGger:RUNT:TIME**

---

Description	Sets or queries the pulse runt search trigger time.	
Syntax	:SEARCH:TRIGger:RUNT:TIME {<NRf>   ? }	
Related commands	:SEARCH:TRIGger:TYPE :SEARCH:TRIGger:RUNT:WHEn	
Parameter	<NRf>	Pulse runt time (4nS to 10S)
Return Parameter	<NR3>	Returns the runt time in seconds.
Example	:SEARCH:TRIGger:RUNT:TIME 4.00E-5 Sets the pulse runt time to 40.0uS.	

Set →  
 → Query

**:SEARCH:TRIGger:RISEFall:WHEn**

---

Description	Sets or queries the rise and fall search trigger conditions.	
Syntax	:SEARCH:TRIGger:RISEFall:WHEn {MOREthan   LESSthan   Equal   UNEQual  ? }	
Related commands	:SEARCH:TRIGger:TYPE :SEARCH:TRIGger:RISEFall:TIME	
Parameter	MOREthan	>
	LESSthan	<
	Equal	=
	UNEQual	≠
Return parameter	Returns the rise and fall search trigger condition.	
Example	:SEARCH:TRIGger:RISEFall:WHEn UNEQual Sets the rise and fall search trigger condition to unequal (≠).	

**:SEARCH:TRIGger:RISEFall:TIME** 



Description	Sets or queries the rise and fall time.	
Syntax	:SEARCH:TRIGger:RISEFall:TIME {<NRf>   ? }	
Related commands	:SEARCH:TRIGger:TYPE :SEARCH:TRIGger:RISEFall:WHEn	
Parameter	<NRf>	Rise and Fall time (4nS to 10S)
Return Parameter	<NR3>	Returns the rise and fall time in seconds.
Example	:SEARCH:TRIGger:RISEFall:TIME 4.00E-5 Sets the trigger rise and fall time to 40.0us.	

**:SEARCH:TRIGger:BUS:TYPE** 

Description	Returns the current bus type.	
Syntax	:SEARCH:TRIGger:BUS:TYPE?	
Return parameter	I2C	I2C mode
	UART	UART mode
	CAN	CAN mode
	LIN	LIN mode
Example	:SEARCH:TRIGger:BUS:TYPE? UART	

**:SEARCH:TRIGger:BUS:B1:I2C:CONDition** 


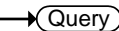


Description	Sets or queries the I <sup>2</sup> C search trigger conditions.	
Syntax	:SEARCH:TRIGger:BUS:B1:I2C:CONDition {START   STOP   REPEATstart   ACKMISS   ADDRess   DATA   ADDRANDDATA   ? }	
Parameter	START	Set Start as the I <sup>2</sup> C search trigger condition.

STOP	Set Stop as the I <sup>2</sup> C search trigger condition.
REPEATstart	Set Repeat of Start as the I <sup>2</sup> C search trigger condition.
ACKMISS	Set Missing Acknowledgement as the I <sup>2</sup> C search trigger condition.
ADDRess	Set Address as the I <sup>2</sup> C search trigger condition.
DATA	Set Data as the I <sup>2</sup> C search trigger condition.
ADDRANDDATA	Set Address and Data as the I <sup>2</sup> C search trigger condition.

Return parameter Returns the I<sup>2</sup>C bus search trigger condition.

Example :SEARCH:TRIGger:BUS:B1:I2C:CONDition ADDRess  
Set Address as the I<sup>2</sup>C search trigger condition.

:SEARCH:TRIGger:BUS:B1:I2C:ADDRess   
:MODE 

Description Sets or queries the I<sup>2</sup>C addressing mode (7 or 10 bits) for the search trigger.

Syntax :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:MODE {ADDR7 | ADDR10 | ? }

Related commands :SEARCH:TRIGger:BUS:B1:I2C:CONDition

Parameter	ADDR7	7 bit addressing
	ADDR10	10 bit addressing

Return Parameter	0	7 bit addressing
	1	10 bit addressing

Example :SEARCH:TRIGger:BUS:B1:I2C:ADDRess:MODE? 0  
The addressing mode is current set to 7 bits.

**:SEARCH:TRIGger:BUS:B1:I2C:ADDRes** (Set) →  
**:TYPE** → (Query)

**Description** Sets the I<sup>2</sup>C bus address type, or queries what the setting is for the search trigger.

**Syntax** :SEARCH:TRIGger:BUS:B1:I2C:ADDRes:TYPE  
 {GENERALcall | STARTbyte | HSmode | EEPROM | CBUS  
 | ?}

**Related commands** :SEARCH:TRIGger:BUS:B1:I2C:CONDition

<b>Parameter</b>	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:TRIGger:BUS:B1:I2C:ADDRes:TYPE?  
 CBUS

**:SEARCH:TRIGger:BUS:B1:I2C:ADDRes** (Set) →  
**:VALue** → (Query)

**Description** Sets or queries the I<sup>2</sup>C bus address value when the I<sup>2</sup>C search trigger is set to trigger on Address or Address/Data.

**Syntax** :SEARCH:TRIGger:BUS:B1:I2C:ADDRes:VALue  
 {<string> | ? }

**Related commands** :SEARCH:TRIGger:BUS:B1:I2C:ADDRes:MODE

Parameter	<string>	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 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:TRIGger:BUS:B1:I2C:ADDRess     Set →  
:DIRection                                   → Query

**Description**       Sets or queries the address bit as read write or don't care for the search function.

 **Note**           This setting only applies when the I2C 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 Parameter** 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 (Set) →  
→ (Query)

**Description** Sets or queries the data size in bytes for the I<sup>2</sup>C bus.

 **Note** This setting only applies when the I2C search trigger is set to trigger on Data or Address/Data

**Syntax** :SEARCH:TRIGger:BUS:B1:I2C:DATa:SIZE {<NR1> | ? }

**Related commands** :SEARCH:TRIGger:BUS:B1:I2C:CONDition

**Parameter** <NR1> Number of data bytes (1 to 5).

**Return parameter** <NR1> Returns the number of bytes.

**Example** :SEARCH:TRIGger:BUS:B1:I2C:DATa:SIZE 3  
Sets the number of bytes to 3.

:SEARCH:TRIGger:BUS:B1:I2C:DATa:VALue (Set) →  
→ (Query)

**Description** Sets or queries the triggering data value for the I<sup>2</sup>C bus when the I<sup>2</sup>C search trigger is set to trigger on Data or Address/Data.

**Syntax** :SEARCH:TRIGger:BUS:B1:I2C:DATa:VALue {<string> | ? }

**Related commands** :SEARCH:TRIGger:BUS:B1:I2C:DATa:SIZE

**Parameter** <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


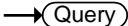
Example 2 :SEARCH:TRIGger:BUS:B1:I2C:DATA:VALue?  
 1X1X0101

:SEARCH:TRIGger:BUS:B1:UART (Set) →  
 :CONDition ← (Query)


Description	Sets or queries the UART search triggering condition.	
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:   
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 Rx Data


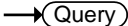
**Syntax**                :SEARCH:TRIGger:BUS:B1:UART:RX:DATA:SIZE  
{<NR1> | ?}

**Related commands**    :SEARCH:TRIGger:BUS:B1:UART:CONDition

**Parameter**           <NR1>           Number of bytes (1 to 10).

**Return parameter**   <NR1>           Returns the number of bytes.

**Example**               :SEARCH:TRIGger:BUS:B1:UART:RX:DATA:SIZE 5  
Sets the number of bytes to 5.

:SEARCH:TRIGger:BUS:B1:UART:RX:DATA:   
VALue 

**Description**           Sets or queries the search triggering data value for the UART bus when the bus is set to trigger on Rx Data.

**Syntax**                :SEARCH:TRIGger:BUS:B1:UART:RX:DATA:VALue  
{<string> | ?}

**Related commands**    :SEARCH:TRIGger:BUS:B1:UART:RX:DATA:SIZE


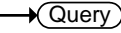
**Parameter**           <string>        The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string".

x = don't care  
 1 = binary 1  
 0 = binary 0


**Return Parameter** Returns the data value.

**Example 1** :SEARCH:TRIGger:BUS:B1:UART:CONDition RXDATA  
 :SEARCH:TRIGger:BUS:B1:UART:RX:DATA:SIZE 1  
 :SEARCH:TRIGger:BUS:B1:UART:RX:DATA:VALue  
 "1x1x0101"  
 Sets the value to 1x1x0101

**Example 2** :SEARCH:TRIGger:BUS:B1:UART:RX:DATA:VALue?  
 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


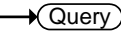
**Syntax** :SEARCH:TRIGger:BUS:B1:UART:TX:DATA:SIZE  
 {<NR1> | ?}

**Related commands** :SEARCH:TRIGger:BUS:B1:UART:CONDition

**Parameter** <NR1> Number of bytes (1 to 10).

**Return parameter** <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:   
 VALue 

**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> | ? }

Related commands :SEARCH:TRIGger:BUS:B1:UART:TX:DATA:SIZE

Parameter	<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:TRIGger:BUS:B1:UART:CONDition TXDATA  
:SEARCH:TRIGger:BUS:B1:UART:TX:DATA:SIZE 1  
:SEARCH:TRIGger:BUS:B1:UART:TX:DATA:VALue "1x1x0101"  
  
Sets the value to 1x1x0101

Example 2 :SEARCH:TRIGger:BUS:B1:UART:TX:DATA:VALue? 1X1X0101

(Set) →

:SEARCH:TRIGger:BUS:B1: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:TRIGger:BUS:B1:CAN:CONDition SOF  
Triggers search on a start of frame.

Example2 :SEARCH:TRIGger:BUS:B1:CAN:CONDition?  
>SOF

Set →

:SEARCH:TRIGger:BUS:B1:CAN:FRAMEType → Query

Description Sets or returns the frame type for the CAN FRAMEType search trigger.

Syntax :SEARCH:TRIGger:BUS:B1:CAN:FRAMEType {DATA|REMote|ERRor|OVERLoad|?}

Parameter/ Return parameter	DATA	Sets the frame type to data frame
	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:TRIGger:BUS:B1:CAN:FRAMEType DATA  
Sets the frame type to DATA.


:SEARCH:TRIGger:BUS:B1:CAN:IDentifier: MODE Set →  
→ Query

Description Sets or returns the CAN identifier mode for the bus.

Syntax	:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODE {STANDard EXTended ?}	
Parameter/ Return parameter	STANDard	Standard addressing mode
	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:TRIGger:BUS:B1:CAN:IDentifier: VALue Set → Query

Description	Sets or returns the identifier string used for the CAN search trigger.
-------------	--

 Note	Only applicable when the search trigger condition is set to ID or IDANDDATA.
--	--

Syntax	:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:VALue {<string> ?}
--------	--


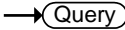
Related Commands	:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODE
------------------	--

Parameter/ Return parameter	<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:CAN:CONDition ID	
	:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODE STANDARD	
	:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:VALue	


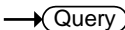
“01100X1X01X”


:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:VALue?  
>01100X1X01X

:SEARCH:TRIGger:BUS:B1:CAN:IDentifier   
:DIRection 

Description	Sets or queries the address bit as read, write or don't care.	
Syntax	:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:DIRection {READ WRITE NOCARE ?}	
Parameter/ Return parameter	READ	Sets read as the data direction
	WRITE	Sets write as the data direction
	NOCARE	Sets either as the data direction

Example2 :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:DIRection?  
>WRITE  
  
:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:DIRection  
READ  
  
:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:DIRection?  
>READ

:SEARCH:TRIGger:BUS:B1:CAN:DATA:   
QUALifier 

Description	Sets or returns the CAN data qualifier.	
 Note	Only applicable when the search triggering condition is set to DATA or IDANDDATA.	
Syntax	:SEARCH:TRIGger:BUS:B1:CAN:DATA:QUALifier {LESSthan MOREthan EQual UNEQual LESSEQual MOREEQual ?}	
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:TRIGger:BUS:B1:CAN:DATA:QUALifier?
>EQUAL
:SEARCH:TRIGger:BUS:B1:CAN:DATA:QUALifier
MOREthan
:SEARCH:TRIGger:BUS:B1:CAN:DATA:QUALifier?
>MORETHAN
```

Set →

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:TRIGger:BUS:B1:CAN:DATA:SIZE {<NR1>|?}
```

Parameter/ Return parameter	<NR1>	1~8 (bytes)
--------------------------------	-------	-------------

Example

```
:SEARCH:TRIGger:BUS:B1:CAN:DATA:SIZE?
>1
:SEARCH:TRIGger:BUS:B1:CAN:DATA:SIZE 2
:SEARCH:TRIGger:BUS:B1:CAN:DATA:SIZE?
>2
```

:SEARCH:TRIGger:BUS:B1:CAN:DATA: VALue (Set) →  
→ (Query)

**Description** Sets or returns the binary data string to be used for the CAN search trigger.

**Related Commands** :SEARCH:TRIGger:BUS:B1:CAN:DATA:SIZE

**Syntax** :SEARCH:TRIGger:BUS:B1:CAN:DATA:VALue {<string>|?}

<b>Parameter/ Return parameter</b>	<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:CAN:DATA:SIZE 1  
 :SEARCH:TRIGger:BUS:B1:CAN:DATA:VALue "01010X1X"  
 :SEARCH:TRIGger:BUS:B1:CAN:DATA:VALue? >01010X1X

(Set) →  
→ (Query)

:SEARCH:TRIGger:BUS:B1:LIN:CONDition

**Description** Sets or returns the LIN search trigger condition.

**Syntax** :SEARCH:TRIGger:BUS:B1:LIN:CONDition {SYNCField|IDentifier|DATA|IDANDDATA|WAKEup|SL EEP|ERRor|?}



<b>Parameter/ Return parameter</b>	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:TRIGger:BUS:B1:LIN:DATA:    
 QUALifier 

Description

Sets or returns the LIN data qualifier.  
 Note: Only applicable when the search trigger condition is set to DATA or IDANDDATA.

Syntax

```
:SEARCH:TRIGger:BUS:B1:LIN:DATA:QUALifier
{LESSthan|MOREthan|EQual|UNEQual|LESSEQual|M
OREEQual|?}
```

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:TRIGger:BUS:B1:LIN:DATa:QUALifier?  
>EQUAL  
:SEARCH:TRIGger:BUS:B1:LIN:DATa:QUALifier?  
MOREthan  
:SEARCH:TRIGger:BUS:B1:LIN:DATa:QUALifier?  
>MORETHAN

(Set) →

:SEARCH:TRIGger: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:TRIGger:BUS:B1:LIN:DATa:SIZE {<NR1>|?}

Parameter/ Return parameter	<NR1>	1~8 (bytes)
--------------------------------	-------	-------------

Example :SEARCH:TRIGger:BUS:B1:LIN:DATa:SIZE?  
>1  
:SEARCH:TRIGger:BUS:B1:LIN:DATa:SIZE 2  
:SEARCH:TRIGger:BUS:B1:LIN:DATa:SIZE?  
>2

(Set) →

:SEARCH:TRIGger:BUS:B1:LIN:DATa:VALue → (Query)

Description Sets or returns the binary data string to be used for the LIN search trigger.



Note

Only applicable when the condition is set to DATA or IDANDDATA.


Related Commands	:SEARCH:TRIGger:BUS:B1:LIN:DATA:SIZE	
Syntax	:SEARCH:TRIGger:BUS:B1:LIN:DATA:VALue {<string> ?}	
Parameter/Return parameter	<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:TRIGger:BUS:B1:LIN:DATA:VALue? >01010X1X	

:SEARCH:TRIGger:BUS:B1:LIN:ERRTYPE  

Description	Sets or returns the error type to be used for the LIN search trigger.	
Syntax	:SEARCH:TRIGger:BUS:B1:LIN:ERRTYPE {SYNC PARItY CHecksum ?}	
Parameter/Return parameter	SYNC	Sets the LIN error type to SYNC.
	PARItY	Sets the LIN error type to parity.
	CHecksum	Sets the LIN error type to checksum.
Example	:SEARCH:TRIGger:BUS:B1:LIN:ERRTYPE? >SYNC :SEARCH:TRIGger:BUS:B1:LIN:ERRTYPE CHECKSUM :SEARCH:TRIGger:BUS:B1:LIN:ERRTYPE? >CHECKSUM	

:SEARCH:TRIGger:BUS:B1:LIN:IDentifier: VALue (Set) →  
→ (Query)

**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>|?}

<b>Parameter/ Return parameter</b>	<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"
:SEARCH:TRIGger:BUS:B1:LIN:IDentifier:VALue?
>01100X1X01X
```

:SEARCH:FFTPeak:METhod (Set) →  
→ (Query)

**Description** Sets or returns the FFT peak method type.

**Related Commands** :SEARCH:TRIGger:TYPE  
:SEARCH:FFTPeak:METhod:MPEak  
:SEARCH:TRIGger:LEVel

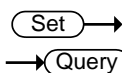
**Syntax** :SEARCH:FFTPeak:METhod {MPEak | LEVel | ?}

Parameter/ Return parameter	MPEak	Sets the peak method to the Max Peak type.
	LEVel	Sets the peak methods to the Level type.

Example

```
:SEARCH:FFTPeak:METhod LEVEL
:SEARCH:FFTPeak:METhod?
>LEVEL
:SEARCH:TRIGger:LEVel?
>1.000E+00
:SEARCH:TRIGger:LEVel 2
:SEARCH:TRIGger:LEVel?
>2.000E+00
```

:SEARCH:FFTPeak:METhod:MPEak



Description

Sets the active peak number (1 ~ 10) or return the frequency of the active peak number.

Related Commands

```
:SEARCH:TRIGger:TYPe
:SEARCH:FFTPeak:METhod
```

Syntax

```
:SEARCH:FFTPeak:METhod:MPEak {<NR1> | ?}
```

Parameter

<NR1>	Active peak number.
-------	---------------------

Return parameter

<NR3>	Frequency of the active peak.
-------	-------------------------------

Example

```
:SEARCH:FFTPeak:METhod MPEak
:SEARCH:FFTPeak:METhod?
>MPEAK
:SEARCH:FFTPeak:METhod:MPEak?
>1.000E+00
:SEARCH:FFTPeak:METhod:MPEak 2
:SEARCH:FFTPeak:METhod:MPEak?
>2.000E+00
```

Set →  
 → Query

**:SEARCH:FFTPeak:SINFo**

---


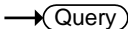
Description	Sets or returns the State Info to “Mark” or “Peak”.	
Related Commands	:SEARCH:TRIGger:TYPE	
Syntax	:SEARCH:FFTPeak:SINFo {MARK   PEAK   ?}	
Parameter/ Return parameter	MARK	Sets the State Info to Mark.
	PEAK	Sets the State Info to Peak.
Example	<pre>:SEARCH:FFTPeak:SINFo? &gt;PEAK :SEARCH:FFTPeak:SINFo mark :SEARCH:FFTPeak:SINFo? &gt;MARK</pre>	

Set →  
 → Query

**:SEARCH:TRIGger:BUS:B1:SPI:CONDition**


---



Description	Sets or queries 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	<pre>:SEARCH:TRIGger:BUS:B1:SPI:CONDition MISO Sets the SPI bus to trigger on MISO.</pre>	

**:SEARCH:TRIGger:BUS:B1:SPI:DATA:SIZE**

---

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>   ?}	
Related commands	:SEARCH:TRIGger:BUS:B1:SPI:CONDition	
Parameter	<NR1>	Number of words (1 to 32).
Return parameter	<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:VALue**

---

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>   ?}	
Related commands	:SEARCH:TRIGger:BUS:B1:SPI:DATA:SIZE	
Parameter	<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:SPI:CONDition MISO  
                       :SEARCH:TRIGger:BUS:B1:SPI:DATA:SIZE 2  
                       :SEARCH:TRIGger:BUS:B1:SPI:DATA:MISO:VALue  
                       "1x1x0101"  
                       Sets the value to 1x1x0101

Example 2           :SEARCH:TRIGger:BUS:B1:SPI:DATA:MISO:VALue?  
                       1X1X0101

(Set) →

:SEARCH:TRIGger:BUS:B1:SPI:DATA:MOSI:VALue → (Query)

Description       Sets or queries the search triggering data value for the SPI bus when the bus is set to trigger on MOSI or MISO/MOSI.

Syntax             :SEARCH:TRIGger:BUS:B1:SPI:DATA:MOSI:VALue  
                       {<string> | ? }


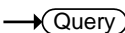
Related commands       :SEARCH:TRIGger:BUS:B1:SPI:DATA:SIZE

Parameter	<sting>	<p>The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string".</p> <p>x = don't care</p> <p>1 = binary 1</p> <p>0 = binary 0</p>
-----------	---------	---

Return Parameter   Returns the data value.

Example1           :SEARCH:TRIGger:BUS:B1:SPI:CONDition MOSI  
                       :SEARCH:TRIGger:BUS:B1:SPI:DATA:SIZE 2  
                       :SEARCH:TRIGger:BUS:B1:SPI:DATA:MOSI:VALue  
                       "1x1x0101"  
                       Sets the value to 1x1x0101

Example2           :SEARCH:TRIGger:BUS:B1:SPI:DATA:MOSI:VALue?  
                       1X1X0101

**:SEARCH:FFTPeak:SOURce** 



Description	Sets or returns the fft peak source.	
Related Commands	:SEARCH:FFTPeak:METHOD :SEARCH:FFTPeak:METHOD:MPEak :SEARCH:FFTPeak:SINFo :SEARCH:FFTPeak:LIST When in SA Mode, the SA source affects the operation object of the above command.	
Syntax	:SEARCH:FFTPeak:SOURce {SA1 SA2} :SEARCH:FFTPeak:SOURce?	
Parameter	SA1	Sets the fft peak source as SA1.
	SA2	Sets the fft peak source as SA2.
Example	:SEARCH:FFTPeak: SOURce SA2 :SEARCH:FFTPeak: SOURce? SA2	

**:SEARCH:TRIGger:BUS:TYPE** 

Description	Returns the current bus trigger search type as I <sup>2</sup> C, SPI, UART or PARALLEL .	
Syntax	:SEARCH:TRIGger:BUS:TYPE?	
Parameter	I2C	I <sup>2</sup> C mode
	SPI	SPI mode
	UART	UART mode
	PARALLEL	Parallel mode
Example	:SEARCH:TRIGger:BUS:TYPE? UART	

---

:SEARCH:FFTPeak:LIST

→ Query

---

Description Returns the data of the search event table.

---

Syntax :SEARCH:FFTPeak:LIST?

---

Example :SEARCH:FFTPeak:LIST?  
No.,Frequency,Value;  
1,1.000E+04,-6.400E+00;  
2,2.750E+06,-7.360E+01;  
3,2.830E+06,-7.280E+01;  
4,2.910E+06,-7.200E+01;  
5,3.020E+06,-7.120E+01;  
6,3.170E+06,-7.040E+01;  
7,5.550E+06,-8.240E+01;  
8,5.640E+06,-8.160E+01;  
9,5.740E+06,-8.080E+01;  
10,5.900E+06,-8.000E+01;

---



## Label Commands



:CHANnel<X>:LABel.....	239
:CHANnel<X>:LABel:DISPlay .....	240
:REF<X>:LABel .....	240
:REF<X>:LABel:DISPlay .....	241
:BUS1:LABel.....	242
:BUS1:LABel:DISPlay .....	242
:SET<X>:LABel .....	243

:CHANnel<X>:LABel 


Description	Sets or returns the file label for the selected channel.	
Syntax	:CHANnel<X>:LABel {<string>   ?}	
Related commands	:CHANnel<X>:LABel:DISPlay	
Parameter	<X>	Channel 1, 2, 3, 4
	<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>	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 channel 1 label as "CH1_lab".	
Example2	:CHANnel1:LABel? CH1_lab	


		 
<b>:CHANnel&lt;X&gt;:LABel:DISPlay</b>		
Description	Turns the label on/off for the selected channel or returns its status.	
Syntax	:CHANnel<X>:LABel:DISPlay { OFF   ON   ? }	
Related commands	:CHANnel<X>:LABel	
Parameter	<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	<pre>:CHANnel1:LABel "CH1" :CHANnel1:LABel:DISPlay ON :CHANnel1:LABel:DISPlay? ON</pre> <p>Sets the channel 1 label to "CH1" and then turns the label display on. The query return shows that the label is on.</p>	

		 
<b>:REF&lt;X&gt;:LABel</b>		
Description	Sets or returns the file label for the selected reference waveform.	
Syntax	:REF<X>:LABel {<string>   ?}	
Related commands	:REF<X>:LABel:DISPlay	
Parameter	<X>	REF 1, 2, 3, 4

	<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>	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 REF1 label as "REF1\_lab".

Example2 :REF1:LABel?  
REF1\_lab

:REF<X>:LABel:DISPlay  

Description Turns the label on/off for the selected reference waveform or returns its status.

Syntax :REF<X>:LABel:DISPlay { OFF | ON | ? }

Related commands :REF<X>:LABel

Parameter	<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.	
Syntax	:BUS1:LABel {<string>   ?}	
Related commands	:BUS1:LABel:DISPlay	
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".
Return parameter	<string>	Returns the label for the bus. No return indicates that there has not been a file label assigned for bus.

Example1 :BUS1:LABel "Bus"  
Sets the bus label as "Bus".

Example2 :BUS1:LABel?  
Bus

Set →

→ Query

**:BUS1:LABel:DISPlay**

Description	Turns the label on/off for the bus or returns its status.	
Syntax	:BUS1:LABel:DISPlay { OFF   ON   ? }	
Related commands	:BUS1:LABel :D<x>:LABel :D<x>:LABel:DISPlay	

	:DIGital:ANALog:A<x>:LABel				
	:DIGital:ANALog:A<x>:LABel:DISPlay				
Parameter	<table border="1"> <tr> <td>OFF</td> <td>Turns the file label off for the bus.</td> </tr> <tr> <td>ON</td> <td>Turns the file label on for the bus.</td> </tr> </table>	OFF	Turns the file label off for the bus.	ON	Turns the file label on for the bus.
OFF	Turns the file label off for the bus.				
ON	Turns the file label on for the bus.				
Return parameter	Returns the status of the file label for the bus (ON, OFF).				
Example	<pre>:BUS1:LABel "Bus" :BUS1:LABel:DISPlay ON :BUS1:LABel:DISPlay? ON</pre> <p>Sets the label for the bus to "Bus" and then turns the label display on. The query return shows that the label is on.</p>				

:SET<X>:LABel 

Description	Sets or returns the file label for the selected setup.				
Syntax	:SET<X>:LABel {<string>   ?}				
Related commands	:SET<X>:LABel:DISPlay				
Parameter	<table border="1"> <tr> <td>&lt;X&gt;</td> <td>Setup number 1 to 20</td> </tr> <tr> <td>&lt;string&gt;</td> <td>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".</td> </tr> </table>	<X>	Setup number 1 to 20	<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".
<X>	Setup number 1 to 20				
<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>	Returns the label for the selected setup. No return indicates that there has not been a file label assigned for the selected setup.			
Example1	<pre>:SET1:LABel "SET1_lab"</pre> <p>Sets the label for setup 1 as "SET1_lab".</p>				

Example2       :SET1:LABel?  
                  SET1\_lab

## Segment Commands

		Set →
		→ Query
<b>:SEGMents:STATE</b>		
Description	Turns the segmented memory function on/off or queries its state.	
Syntax	:SEGMents:STATE { OFF   ON   ? }	
Related commands	:RUN :STOP	
Parameter/ Return parameter	OFF	Turns the segmented memory off.
	ON	Turns the segmented memory on.
Example1	:SEGMents:STATE ON Turns segmented memory on.	

		Set →
		→ Query
<b>:SEGMents:CURRent</b>		
Description	Sets or queries the current segment. The total number of segments depends on the record length.	
Syntax	:SEGMents:CURRent {SETTOMIN SETTOMAX <NR1> ?}	
Related commands	:SEGMents:STATE :SEGMents:TOTAlnum	
Parameter/ Return parameter	SETTOMIN	Current segment = min segment
	SETTOMAX	Current segment = max segment
	<NR1>	1~29000
Example1	:SEGMents:CURRent 10 Sets the current segment to segment number 10.	

Set →

→ Query

**:SEGMENTS:TOTALnum**

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> ?}	
Related commands	:SEGMENTS:STATE :SEGMENTS:CURREnt	
Parameter/ Return parameter	SETTOMIN	Sets to the minimum number
	SETTOMAX	Sets to the maximum number
	<NR1>	1~29000
Example1	:SEGMENTS:TOTALnum SETTOMAX Sets the number of segments to max number (29000).	

**:SEGMENTS:TIME**

→ Query

Description	Returns the time of the current segment in relation to the first segment.	
Syntax	:SEGMENTS:TIME?	
Related commands	:SEGMENTS:STATE :SEGMENTS:CURREnt	
Return parameter	The segment time as <NR3>.	
Example	:SEGMENTS:TIME? >8.040E-03 Returns the segment time.	

**:SEGMents:DISPALL** 



Description	Sets or queries whether all the segments are displayed on the screen.	
Syntax	:SEGMents:DISPALL {OFF ON ?}	
Related commands	:SEGMents:STATE :SEGMents:CURRent	
Parameter/ Return parameter	OFF	Turns the display all function off.
	ON	Turns the display all function on.
Example1	:SEGMents:DISPALL ON Turns the display all function on.	

**:SEGMents:MEASure:MODE** 



Description	Sets or queries the measurement mode.	
Syntax	:SEGMents:MEASure:MODE {OFF PLOT TABLE ?}	
Related commands	:MEASUrement:MEAS<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:MEASure:MODE? >PLOT Returns the measurement mode as Statistics.	

**:SEGMents:MEASure:PLOT:SOURce** 



Description	Sets or queries the statistics source.	
Syntax	:SEGMents:MEASure:PLOT:SOURce {<NR1>   ? }	
Related commands	:SEGMents:MEASure:MODE :SEGMents:MEASure:PLOT:DIVide :SEGMents:MEASure:PLOT:SElect :SEGMents:MEASure:PLOT:RESults	
Parameter/ Return parameter	<NR1>	1~8 (Automatic measurement item 1~8)

Example1 :SEGMents:MEASure:PLOT:SOURce 1  
Sets the source as auto measurement item 1.

**:SEGMents:MEASure:PLOT:DIVide** 



Description	Sets or queries the number of bins for the statistics function.	
Syntax	:SEGMents:MEASure:PLOT:DIVide {<NR1>   ? }	
Related commands	:SEGMents:MEASure:PLOT:SOURce :SEGMents:MEASure:PLOT:SElect	
Parameter/ Return parameter	<NR1>	1~20

Example1 :SEGMents:MEASure:PLOT:DIVide 5  
Sets the number of bins to 5 for the statistics function.

**:SEGMents:MEASure:PLOT:SElect** 



Description	Sets or queries which bin to view the statics of.	
Syntax	:SEGMents:MEASure:PLOT:SElect {<NR1>   ? }	
Related commands	:SEGMents:MEASure:PLOT:SOURce :SEGMents:MEASure:PLOT:DIVide	

Parameter <NR1> 1~20 (cannot exceed the number of bins)

Return parameter Return the bin number as <NR3>.

Example1 :SEGMents:MEASure:PLOT:SElect 5  
Set to bin number 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.

		(Set) →
		→ (Query)
<b>:SEGMents:MEASure:TABLE:SOURce</b>		
Description	Sets or queries the source of the measurement list.	
Syntax	:SEGMents:MEASure:TABLE:SOURce {CH1   CH2   CH3   CH4   ? }	
Related commands	:SEGMents:MEASure:MODE :SEGMents:MEASure:TABLE:SElect :SEGMents:MEASure:TABLE:LIST	
Parameter/ Return parameter	CH1~CH4	Channel 1 to 4
Example1	:SEGMents:MEASure:TABLE:SOURce CH1 Sets the source to CH1.	

		(Set) →
		→ (Query)
<b>:SEGMents:MEASure:TABLE:SElect</b>		
Description	Sets or queries a segment to view in the measurement table.	
Syntax	:SEGMents:MEASure:TABLE:SElect {<NR1>   ? }	
Related commands	:SEGMents:TOTALnum	
Parameter	<NR1>	1~29000
Return parameter	Returns the number of segments as <NR3>.	
Example1	:SEGMents:MEASure:TABLE:SElect 10 Select segment number 10.	

		→ (Query)
<b>:SEGMents:MEASure:TABLE:LIST</b>		
Description	Returns the measurement results of each segment in the block data.	
Syntax	:SEGMents:MEASure:TABLE:LIST?	
Return parameter	Returns the measurements results as a block data for each segment.	

Example :SEGMents:MEASure:TABLE:LIST?  
 >"GW MPO-2102P, serial number PXXXXXX,  
 version V1.37",  
 Segment Summary : CH1,  
 Seg., Pk-Pk (V), Pk-Pk (V),  
 1, 8.00m, 8.00m,  
 2, 8.00m, 8.00m,  
 3, 8.00m, 8.00m,  
 4, 8.00m, 8.00m,  
 5, 8.00m, 8.00m,  
 6, 8.00m, 8.00m,  
 7, 8.00m, 8.00m,  
 8, 8.00m, 8.00m,  
 9, 12.0m, 12.0m,  
 10, 8.00m, 8.00m,

**:SEGMents:MEASure:TABLE:SAVE** (Set) →

Description Saves the list of segment automatic measurement results.

Syntax :SEGMents:MEASure:TABLE:SAVE

**:SEGMents:SAVE** (Set) →

Description Saves the segments.

Syntax :SEGMents:SAVE

Related Commands :SEGMents:SAVE:SOURce  
 :SEGMents:SAVE:SElect:START  
 :SEGMents:SAVE:SElect:END

Example :SEGMents:SAVE:SOURce CH1  
 :SEGMents:SAVE:SElect:START 1  
 :SEGMents:SAVE:SElect:END 10  
 :SEGMents:SAVE

Set →  
 → Query

**:SEGMents:SAVe:SOURce**

---

Description	Sets or queries the source segment waveform to save.	
Syntax	:SEGMents:SAVe:SOURce {CH1   CH2  CH3   CH4  ? }	
Parameter/ Return parameter	CH1~CH4	Channel 1 to 4.
Example	:SEGMents:SAVe:SOURce CH1 >Sets the source to CH1.	

Set →  
 → Query

**:SEGMents:SAVe:SElect:START**

---

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>   ? }	
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>	Sets the segment to 1~29000
Example	:SEGMents:SAVe:SElect:START 2 Sets the starting segment to segment number 2.	

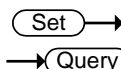
Set →  
 → Query

**:SEGMents:MASK:STATe**

---

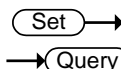
Description	Sets or queries the state of segment mask.	
Note	It could not set on when the mask hasn't create.	

Syntax	:SEGMents:MASK:STATe {ON   OFF} :SEGMents:MASK:STATe?	
Parameter	ON	Turn on the segment mask.
	OFF	Turn off the segment mask.
Example	:SEGMents:MASK:STATe ON :SEGMents:MASK:STATe? ON	



**:SEGMents:MASK:SOURce**

Description	Sets or queries the referance source of segment mask.	
Syntax	:SEGMents:MASK:SOURce {CH1 CH2 CH3 CH4} :SEGMents:MASK:SOURce?	
Parameter	CH1~CH4	Channel 1 to Channel 4
Example	SEGMents:MASK:SOURce CH1 :SEGMents:MASK:SOURce? CH1	



**:SEGMents:MASK:CREATE**

Description	Creates a segment mask fastly in according to the referance source or returns the setting of the mask.	
Syntax	:SEGMents:MASK:CREATE {<Xmask>,<Ymask>} :SEGMents:MASK:CREATE?	
Parameter	<Xmask>	Sets the horizontal range for the mask.
	<Ymask>	Sets the vertical range for the mask.
Example	:SEGMents:MASK:CREATE 0.1,0.2 Sets the range of mask as {0.1,0.2} and creates a mask.	

**:SEGMENTS:MASK:UNITs** (Set) →  
→ (Query)


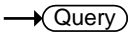
Description	Sets or queries the units of segment mask.	
Syntax	:SEGMENTS:MASK:UNITs {DIVISIONS CURRENT} :SEGMENTS:MASK:UNITs?	
Parameter	DIVISIONS	Divisions units
	CURRENT	Current units
Example	:SEGMENTS:MASK:UNITs CURRENT :SEGMENTS:MASK:UNITs? CURRENT	


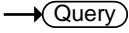
**:SEGMENTS:SAVE:SELECT:END** (Set) →  
→ (Query)

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>   ? }	
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>	Sets the segment to 1~29000.
Return parameter	<NR3>	Returns the ending segment as NR3.
Example	:SEGMENTS:SAVE: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.

		
		
<b>:DVM:STATE</b>		
Description	Sets or queries the DVM state to on or off.	
Syntax	:DVM:STATE {OFF   ON   ? }	
Related commands	:DVM:SOURce :DVM:MODE	
Parameter/ Return parameter	OFF	Turns the DVM off.
	ON	Turns the DVM on.
Example	:DVM:STATE ON Turns the DVM state on.	

		
		
<b>:DVM:SOURce</b>		
Description	Sets or queries the source of the DVM.	
Syntax	:DVM:SOURce {CH1 CH2 CH3 CH4 ?}	
Related commands	:DVM:STATE :DVM:MODE	
Parameter/ Return parameter	CH1~CH4	Channel 1 to 4.
Example	:DVM:SOURce CH1 Sets the DVM source to channel 1.	

Set →  
 → Query

**:DVM:MODE**

Description	Sets or queries the DVM mode.	
Syntax	:DVM:MODE {ACRMS DC DCRMS DUTY FREQUENCY ?}	
Related commands	:DVM:SOURce :DVM:STATE	
Parameter/ Return parameter	ACRMS	Sets the mode to AC RMS
	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 DVM mode to DUTY.	

**:DVM:VALue**

→ Query

Description	Returns the measurement value of the selected mode.	
Syntax	:DVM:VALue?	
Related commands	:DVM:SOURce :DVM:STATE :DVM:MODE	
Return parameter	Returns the measurement value as <NR3>.	
Example	:DVM:VALue? >8.410E-04 Returns the 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.

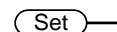
### :GONogo:CLEar



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/NoGo off.

### :GONogo:FUNcTion



Description Initializes the Go/NoGo APP. This must be run after the Go/NoGo APP has been started.

Syntax :GONogo:FUNcTion

### :GONogo:NGCcount



Description Returns the Go/NoGo counter.

Syntax :GONogo:NGCcount{?}

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 →

→ Query

**:GONogo:NGDefine**

Description Sets the Go/NoGo “When” conditions.

Syntax :GONogo:NGDefine {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/NoGo condition to EXITs.

Set →

→ Query

**:GONogo:SOURce**

Description Sets the source for the Go/NoGo signal.

Syntax :GONogo:SOURce {CH1|CH2|CH3|CH4|?}

Parameter/ Return Parameter	CH1~CH4
--------------------------------	---------

Example :GONogo:SOURce CH1  
Sets the source to CH1.

Set →

→ Query

**:GONogo:VIOLation**

Description Sets or returns actions for the Go/NoGo violations.

Syntax :GONogo:VIOLation {STOP | CONTINUE | ?}

Parameter/	STOP	The waveform will be frozen.
------------	------	------------------------------

Return Parameter **CONTInue** Ignore the violation.

Example :GONogo:VIOLation STOP  
Sets violation action to STOP.

**:GONogo:SCRipt** (Set) →

Description Activates/Deactivates the Go/NoGo APP or queries its state.

Syntax :GONogo:SCRipt {OFF | ON | ?}

Parameter/ Return Parameter	<b>ON</b>	Turns Go/NoGo APP on.
	<b>OFF</b>	Turns the Go/NoGo APP off.

Example :GONogo:SCRipt?  
>ON  
The Go/NoGo script is on.

**:TEMPlate:MODE** (Set) →  
→ (Query)

Description Sets or returns the Go/NoGo template mode.

Syntax :TEMPlate:MODE {MAXimum|MINimum|AUTO|?}

Parameter/ Return Parameter	<b>MAXimum</b>	Maximum template
	<b>MINimum</b>	Minimum template
	<b>AUTO</b>	Auto template

Example :TEMPlate:MODE AUTO  
Sets the template mode to AUTO.

**:TEMPlate:MAXimum** (Set) →  
→ (Query)

Description Defines or queries which waveform memory (REF1 or W1~W20) is set to the maximum template.

Syntax :TEMPlate:MAXimum {REF1|W1~W20|?}

Parameter/	REF1	Reference one
Return Parameter	W1~W20	Waveform memory 1 to 20

Example :TEMPLate:MAXimum REF1  
Saves the maximum template to REF1.

Set →

→ Query

:TEMPLate:MINimum

Description Defines or queries which waveform memory (REF1 or W1~W20) is set to the minimum template.

Syntax :TEMPLate:MINimum {REF2|W1~W20|?}

Parameter/	REF2	Reference one
Return Parameter	W1~W20	Waveform memory 1 to 20

Example :TEMPLate:MINimum REF2  
Saves the minimum template to REF2.

Set →

→ Query

:TEMPLate:POSition:MAXimum

Description Sets or queries the position of the maximum template.

Syntax :TEMPLate:POSition:MAXimum {<NR2>|?}

Parameter	<NR2>	Desired template position (-12.0 ~ +12.0 divisions)
-----------	-------	---

Return parameter Returns the position in the following format: "<NR2>Div"

Example :TEMPLate:POSition:MAXimum 3.00  
Sets the maximum template position to 3.00 divisions.

		Set →
		→ Query
<b>:TEMPlate:POSition:MINimum</b>		
Description	Sets or queries the position of the minimum template.	
Syntax	:TEMPlate:POSition:MINimum {<NR2> ?}	
Parameter	<NR2>	Desired template position (-12.0 ~ +12.0 divisions)
Return parameter	Returns the position in the following format: “<NR2>Div”	
Example	:TEMPlate:POSition:MINimum 3.00 Sets the minimum template position to 3.00 divisions.	

		Set →
<b>:TEMPlate:SAVe:MAXimum</b>		
Description	Saves the maximum template.	
Syntax	:TEMPlate:SAVe:MAXimum	

		Set →
<b>:TEMPlate:SAVe:MINimum</b>		
Description	Saves the maximum template.	
Syntax	:TEMPlate:SAVe:MINimum	

		Set →
		→ Query
<b>:TEMPlate:TOLerance</b>		
Description	Sets or queries the tolerance as a percentage.	
Syntax	:TEMPlate:TOLerance {<NR2> ?}	
Parameter/ Return Parameter	<NR2>	The auto tolerance range (0.4% ~ 40%)
Example	:TEMPlate:TOLerance 10 Sets the tolerance to 10%.	

**:TEMPLate:SAVe:AUTo****Set** →

---

Description	Saves the AUTO template (maximum and minimum templates).
Syntax	:TEMPLate:SAVe:AUTo

## AWG Commands

**:AWG:UTIL**

**Set** →

**Description**      Reset all of the AWG settings to the default.

**Syntax**            :AWG:UTIL{PRESet}

**Parameter**        PRESet      Set the AWG settings to default.

**Example**            :AWG:UTIL PRESet

**Set** →

**:AWG:UTIL:AMPCpl**

→ **Query**

**Description**        Set or return the state of amplitude couple.

**Syntax**            :AWG:UTIL:AMPCpl{ON|OFF}  
:AWG:UTIL:AMPCpl?

**Parameter**        ON           Turn on the amplitude couple.  
OFF           Turn off the amplitude couple.

**Example**            :AWG:UTIL:AMPCpl ON  
:AWG:UTIL:AMPCpl?  
>ON

**Set** →

**:AWG:UTIL:FREQCpl**

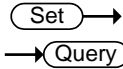
→ **Query**

**Description**        Set or return the type of frequency couple.

**Syntax**            :AWG:UTIL:FREQCpl {OFF| OFFSet| RATio}  
:AWG:UTIL:FREQCpl?

**Parameter**        OFF           Turn off the Frequency Couple.  
OFFSet            The frequency of Gen. 1 and Gen. 2  
                         are coupled by offset.  
RATio              The frequency of Gen. 1 and Gen. 2  
                         are coupled by ratio.

Example :AWG:UTIL:FREQCpl RATio  
 :AWG:UTIL:FREQCpl?  
 >RATIO



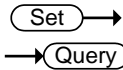
**:AWG:UTIL:FREQCpl:OFFSet**

Description Set or return the frequency offset between Gen. 1 and Gen. 2 for frequency couple

Syntax :AWG:UTIL:FREQCpl:OFFSet {<NRf>}  
 :AWG:UTIL:FREQCpl:OFFSet?

Parameter <NRf> Value of offset.

Example :AWG:UTIL:FREQCpl:OFFSet 50  
 :AWG:UTIL:FREQCpl:OFFSet?  
 >5.00000e+01



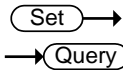
**:AWG:UTIL:FREQCpl:RATio**

Description Set or return the frequency ratio between Gen. 1 and Gen. 2 for frequency couple.

Syntax :AWG:UTIL:FREQCpl:RATio {<NRf>}  
 :AWG:UTIL:FREQCpl:RATio?

Parameter <NRf> Value of ratio.

Example :AWG:UTIL:FREQCpl:RATio 2.5  
 :AWG:UTIL:FREQCpl:RATio?  
 >2.50000e+00



**:AWG:UTIL:TRACKing**

Description Set or return the state of AWG tracking.

Syntax :AWG:UTIL:TRACKing {ON|OFF}  
 :AWG:UTIL:TRACKing?

Parameter ON Turn on the AWG tracking.  
 OFF Turn off the AWG tracking.

Example :AWG:UTIL:TRACkING ON  
 :AWG:UTIL:TRACkING?  
 >ON

Set →

→ Query

**:AWG<x>:AMPlitude**

Description Sets or returns the waveform amplitude.

Syntax :AWG<x>:AMPlitude {<NRf> | ?}

Related command :AWG<x>:OUTPut:LOAd:IMPEDance

Parameter/	<x>	Channel number 1~2.
Return parameter	<NRf>	Amplitude in Volts. (50Ω impedance 0.1~2.5V) (High Z impedance 0.2~5V)

Example :AWG1:AMP 1

Set →

→ Query

**:AWG<x>:FREQuency**

Description Sets or returns the waveform frequency.

Syntax :AWG<x>:FREQuency {<NRf> | ?}

Parameter/	<x>	Channel number 1~2.
Return parameter	<NRf>	Frequency in Hertz.

Example :AWG1:FREQ 2000

Set →

→ Query

**:AWG<x>:FUNCTion**

Description Sets or returns the type of waveform.

Syntax :AWG<x>:FUNCTion {ARBITrary | SINE | SQUAre | PULSe | RAMP | DC | NOISe | SINC | GAUSSian | LORENTz | EXPRise | EXPFall | HAVERSINe | CARDIac | ?}

Parameter/	<x>	Channel number 1~2.
------------	-----	---------------------

Return parameter	ARBitrary	Arbitrary waveform
	SINE	Sine waveform
	SQUAre	Square waveform
	PULSe	Pulse waveform
	RAMP	Ramp waveform
	DC	DC waveform
	NOISe	Noise waveform
	SINC	Sinc waveform
	GAUSSian	Gaussian waveform
	LORENTz	Lorentz waveform
	EXPRise	Exponential rise waveform
	EXPFall	Exponential fall waveform
	HAVERSINe	Haversine waveform
	CARDIac	Cardiac waveform

Example :AWG1:FUNC?  
>SINE

:AWG<x>:OFFSet (Set) →  
→ (Query)

Description Sets or returns the waveform offset.

Syntax :AWG<x>:OFFSet {<NRf> | ?}

Parameter/ Return parameter	<x>	Channel number 1~2.
	<NRf>	Offset in Volts.

Example :AWG1:OFFS

:AWG<x>:OUTPut:LOAd:IMPEDance (Set) →  
→ (Query)

Description Sets or returns the output termination

Syntax :AWG<x>:OUTPut:LOAd:IMPEDance {FIFty | HIGHZ | ?}

Parameter/	<x>	Channel number 1~2
Return parameter	FIFTy	50 Ohm output termination
	HIGHZ	High Z output termination

**Example** :AWG1:OUTP:LOA:IMPED HIGHZ  
 Sets the output termination of channel 1 to high impedance.

Set →  
 → Query

**Description** Sets or returns the channel output state.

**Syntax** :AWG<x>:OUTPut:STATE {OFF | ON | ?}

Parameter/	<x>	Channel number 1~2
Return parameter	OFF	Turns the channel output off
	ON	Turns the channel output on

**Example** :AWG1:OUTP:STATE OFF  
 Turns the channel 1 output off.

Set →  
 → Query

**Description** Sets or returns the channel phase.

**Syntax** :AWG<x>:PHAsE {<NRf> | ?}

Parameter/	<x>	Channel number 1~2.
Return parameter	<NRf>	Phase in degree -180~180°

**Example** :AWG1:PHA 45  
 Sets the channel 1 phase to 45°.

Set →  
 → Query

**Description** Sets or returns the pulse duty cycle.

**Syntax** :AWG<x>:PULSe:DUTYcycle {<NRf> | ?}

Parameter/	<x>	Channel number 1~2.
------------	-----	---------------------

Return parameter <NRf> Duty cycle in percentage 0.2~99.8%

Example :AWG1:PULS:DUTY 50  
Sets the channel 1 pulse duty cycle to 50%.

Set →

:AWG<x>:RAMP:SYMmetry → Query

Description Sets or returns the ramp symmetry.

Syntax :AWG<x>:RAMP:SYMmetry {<NRf> | ?}

Parameter/ Return parameter <x> Channel number 1~2.  
<NRf> Symmetry of the ramp waveform 0~100%

Example :AWG1:RAMP:SYM 15  
Sets the channel 1 ramp symmetry to 15%.

Set →

:AWG<x>:MODulation:STATE → Query

Description Sets or returns the modulation state.

Syntax :AWG<x>:MODulation:STATE {OFF | ON | ?}

Parameter/ Return parameter <x> Channel number 1~2.  
OFF Sets the modulation to off.  
ON Sets the modulation to on.

Example :AWG1:MOD:STATE ON  
Turns the modulation on for channel 1.

Set →

:AWG<x>:MODulation:TYPE → Query

Description Sets or returns the type of modulation.

Syntax :AWG<x>MODulation:TYPE {AM | FM | FSK | ?}

Parameter/ Return parameter <x> Channel number 1~2.  
AM Sets a AM modulation.  
FM Sets a FM modulation.

FSK Sets a FSK modulation.

Example :AWG1:MOD:TYPE AM  
Sets a AM modulation for channel 1.

:AWG<x>:MODulation:AM:DEPth (Set) →  
→ (Query)

Description Sets or returns the AM modulation depth.

Syntax :AWG<x>:MODulation:AM:DEPth {<NRf> | ?}

Parameter/ Return parameter <x> Channel number 1~2.  
<NRf> AM depth in percentage 0~120%.

Example :AWG1:MOD:AM:DEP?  
>1.20000e+02

:AWG<x>:MODulation:AM:FREQ (Set) →  
→ (Query)

Description Sets or returns the AM modulation frequency.

Syntax :AWG<x>:MODulation:AM:FREQ {<NRf> | ?}

Parameter/ Return parameter <x> Channel number 1~2.  
<NRf> AM frequency in Hertz.

Example :AWG1:MOD:AM:FREQ 1000  
Sets the AM frequency to 1kHz.

:AWG<x>:MODulation:AM:SHApe (Set) →  
→ (Query)

Description Sets or returns the shape of the AM modulation.

Syntax :AWG<x>:MODulation:AM:SHApe {SINE | SQUare | PULSe | RAMP | NOISe | ?}

Parameter/ Return parameter <x> Channel number 1~2.  
SINE Sine wave shape.  
SQUare Square wave shape.  
PULSe Pulse wave shape.

RAMP	Ramp wave shape.
NOISe	Noise wave shape.

**Example** :AWG1:MOD:AM:SHA RAMP  
 Sets a ramp shape to the AM modulating waveform.

Set →

:AWG<x>:MODulation:AM:PHase → Query

**Description** Sets or returns the phase of the AM modulation (sine wave shape only).

**Syntax** :AWG<x>:MODulation:AM:PHase {<NRf> | ?}

Parameter/ Return parameter	<x>	Channel number 1~2.
	<NRf>	Phase in degree -180~180°.

**Example** :AWG1:MOD:AM:PHA?  
 >-1.80000e+02

Set →

:AWG<x>:MODulation:AM:DUTYcycle → Query

**Description** Sets or returns the duty cycle of the AM modulation (pulse wave shape only).

**Syntax** :AWG<x>:MODulation:AM:DUTYcycle {<NRf> | ?}

Parameter/ Return parameter	<x>	Channel number 1~2.
	<NRf>	Duty cycle in percentage 2~98%.

**Example** :AWG1:MOD:AM:DUTY 50  
 Sets the duty cycle of the AM modulating waveform to 50%.

Set →

:AWG<x>:MODulation:AM:SYMmetry → Query

**Description** Sets or returns the symmetry of the AM modulation (ramp wave shape only).

**Syntax** :AWG<x>:MODulation:AM:SYMmetry {<NRf> | ?}

Parameter/	<x>	Channel number 1~2.
Return parameter	<NRf>	Symmetry in percentage 0~100%.

**Example** :AWG1:MOD:AM:SYM 50  
 Sets the symmetry of the AM modulating waveform to 50%.

Set →  
 → Query

**:AWG<x>:MODulation:AM:RATE**

**Description** Sets or returns the rate of the AM modulation (noise wave shape only).

**Syntax** :AWG<x>:MODulation:AM:RATE {RATE10M | RATE5M | RATE1M | RATE500K | RATE100K | RATE50K | RATE10K | RATE5K | RATE1K | ?}

Parameter/	<x>	Channel number 1~2.
Return parameter	RATE10M	10MHz noise rate.
	RATE5M	5MHz noise rate.
	RATE1M	1MHz noise rate.
	RATE500K	500kHz noise rate.
	RATE100K	100kHz noise rate.
	RATE50K	50kHz noise rate.
	RATE10K	10kHz noise rate.
	RATE5K	5kHz noise rate.
	RATE1K	1kHz noise rate.

**Example** :AWG1:MOD:AM:RATE RATE5K  
 Sets the noise rate of the AM modulating waveform to 5kHz.

Set →  
 → Query

**:AWG<x>:MODulation:FM:DEV**

**Description** Sets or returns the deviation of the FM modulation.

**Syntax** :AWG<x>:MODulation:FM:DEV {<NRf> | ?}

Parameter/	<x>	Channel number 1~2.
Return parameter	<NRf>	Frequency deviation in Hertz.

Example :AWG1:MOD:FM:DEV?  
>2.000000000e+02

:AWG<x>:MODulation:FM:FREQ Set →  
→ Query

Description Sets or returns the frequency of the FM modulation.

Syntax :AWG<x>:MODulation:FM:FREQ {<NRf> | ?}

Parameter/	<x>	Channel number 1~2.
Return parameter	<NRf>	Frequency in Hertz.

Example :AWG1:MOD:FM:FREQ 1000  
Sets the frequency of the FM modulating waveform to 1kHz.

:AWG<x>:MODulation:FM:SHApe Set →  
→ Query

Description Sets or returns the shape of the FM modulation.

Syntax :AWG<x>:MODulation:FM:SHApe {SINE | SQUare | PULSe | RAMP | NOISe | ?}

Parameter/	<x>	Channel number 1~2.
Return parameter	SINE	Sine wave shape.
	SQUare	Square wave shape.
	PULSe	Pulse wave shape.
	RAMP	Ramp wave shape.
	NOISe	Noise wave shape.

Example :AWG1:MOD:FM:SHA SINE  
Sets a sine shape to the FM modulation.

Set →  
 → Query

**:AWG<x>:MODulation:FM:PHase**

**Description**      Sets or returns the phase of the FM modulation (sine wave shape only).

**Syntax**            :AWG<x>:MODulation:FM:PHase {<NRf> | ?}

<b>Parameter/ Return parameter</b>	<x>	Channel number 1~2.
	<NRf>	Phase in degree -180~180°.

**Example**            :AWG1:MOD:FM:PHA 90  
 Sets a 90° phase to the FM modulating waveform.

Set →  
 → Query

**:AWG<x>:MODulation:FM:DUTYcycle**

**Description**      Sets or returns the duty cycle of the FM modulation (pulse shape wave only).

**Syntax**            :AWG<x>:MODulation:FM:DUTYcycle {<NRf> | ?}

<b>Parameter/ Return parameter</b>	<x>	Channel number 1~2.
	<NRf>	Duty cycle in percentage 1~99%.

**Example**            :AWG1:MOD:FM:DUTY 50  
 Sets the duty cycle of the FM modulating waveform to 50%.

Set →  
 → Query

**:AWG<x>:MODulation:FM:SYMMetry**

**Description**      Sets or returns the symmetry of the FM modulation (ramp shape wave only).

**Syntax**            :AWG<x>:MODulation:FM:SYMMetry {<NRf> | ?}

<b>Parameter/ Return parameter</b>	<x>	Channel number 1~2.
	<NRf>	Symmetry in percentage 0~100%.

**Example**            :AWG1:MOD:FM:SYM 50  
 Sets the symmetry of the FM modulating waveform to 50%.

Set →

:AWG<x>:MODulation:FM:RATE

→ Query

**Description** Sets or returns the noise rate of the FM modulation (noise shape wave only).

**Syntax** :AWG<x>:MODulation:FM:RATE {RATE10M | RATE5M | RATE1M | RATE500K | RATE100K | RATE50K | RATE10K | RATE5K | RATE1K | ?}

<b>Parameter/ Return parameter</b>	<x>	Channel number 1~2.
	RATE10M	10MHz noise rate.
	RATE5M	5MHz noise rate.
	RATE1M	1MHz noise rate.
	RATE500K	500kHz noise rate.
	RATE100K	100kHz noise rate.
	RATE50K	50kHz noise rate.
	RATE10K	10kHz noise rate.
	RATE5K	5kHz noise rate.
RATE1K	1kHz noise rate.	

**Example** :AWG1:MOD:FM:RATE RATE5K  
Sets the noise rate of the FM modulating waveform to 5kHz.

Set →

:AWG<x>:MODulation:FSK:FREQ

→ Query

**Description** Sets or returns the hop frequency of the FSK modulation.

**Syntax** :AWG<x>:MODulation:FSK:FREQ {<NRf> | ?}

<b>Parameter/ Return parameter</b>	<x>	Channel number 1~2.
	<NRf>	Frequency in Hertz.

**Example** :AWG1:MOD:FSK:FREQ 2000000  
Sets the FSK hop frequency to 2MHz.

**:AWG<x>:MODulation:FSK:RATE** (Set) →  
→ (Query)

Description	Sets or returns the FSK modulation rate.	
Syntax	:AWG<x>:MODulation:FSK:RATE {<NRF>   ?}	
Parameter/ Return parameter	<x>	Channel number 1~2.
	<NRF>	Frequency in Hertz.
Example	:AWG1:MOD:FSK:RATE 100000 Sets the FSK rate to 100kHz.	

**:AWG<x>:SWEep:STATE** (Set) →  
→ (Query)

Description	Sets or returns the Sweep mode state.	
Syntax	:AWG<x>:SWEep:STATE {OFF   ON   ?}	
Parameter/ Return parameter	<x>	Channel number 1~2.
	OFF	Sets the sweep mode to off.
	ON	Sets the sweep mode to on.
Example	:AWG1:SWE:STATE ON Turns the sweep mode to on for channel 1.	

**:AWG<x>:SWEep:TYPE** (Set) →  
→ (Query)

Description	Sets or returns the sweep mode type.	
Syntax	:AWG<x>:SWEep:TYPE {LINEAR   LOG   ?}	
Parameter/ Return parameter	<x>	Channel number 1~2.
	LINEAR	Sets the sweep mode to linear.
	LOG	Sets the sweep mode to logarithmic.
Example	:AWG1:SWE:TYP LIN Sets the sweep mode to linear for channel 1.	

:AWG<x>:SWEep:START (Set) →  
→ (Query)

Description	Sets or returns the start frequency of the sweep mode.	
Syntax	:AWG<x>:SWEep:START {<NRf>   ?}	
Parameter/ Return parameter	<x>	Channel number 1~2.
	<NRf>	Start frequency in Hertz.
Example	:AWG1:SWE:START 1000 Sets the sweep mode start frequency to 1kHz.	

:AWG<x>:SWEep:STOP (Set) →  
→ (Query)

Description	Sets or returns the stop frequency of the sweep mode.	
Syntax	:AWG<x>:SWEep:STOP {<NRf>   ?}	
Parameter/ Return parameter	<x>	Channel number 1~2.
	<NRf>	Stop frequency in Hertz.
Example	:AWG1:SWE:STOP 500000 Sets the sweep mode stop frequency to 500kHz.	

:AWG<x>:SWEep:TIME (Set) →  
→ (Query)

Description	Sets or returns the sweep time.	
Syntax	:AWG<x>:SWEep:TIME {<NRf>   ?}	
Parameter/ Return parameter	<x>	Channel number 1~2.
	<NRf>	Sweep time in seconds.
Example	:AWG1:SWE:TIM 6.500e-01 Sets the sweep time to 650ms.	

**:AWG<x>:SWEep:SPAN** 


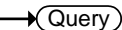
  


**Description**      Alternatively to setting the start and stop frequencies, the span and center frequency can be set.

**Syntax**            :AWG<x>:SWEep:SPAN {<NRF> | ?}

<b>Parameter/ Return parameter</b>	<b>&lt;x&gt;</b>	Channel number 1~2.
	<b>&lt;NRF&gt;</b>	Span of the sweep in Hertz.

**Example**            :AWG1:SWE:SPAN 1100  
Sets the span of the sweep to 1.1kHz.

**:AWG<x>:SWEep:CENTER** 

  


**Description**      Alternatively to setting the start and stop frequencies, the span and center frequency can be set.

**Syntax**            :AWG<x>:SWEep:CENTER {<NRF> | ?}

<b>Parameter/ Return parameter</b>	<b>&lt;x&gt;</b>	Channel number 1~2.
	<b>&lt;NRF&gt;</b>	Center frequency of the sweep in Hertz.

**Example**            :AWG1:SWE:CENT 550  
Sets the center frequency of the sweep to 550Hz.

**:AWG<x>:ARbitrary:EDIT:NUMPoint** 


**Description**      Sets or returns the number of points of an arbitrary waveform.

**Syntax**            :AWG<x>:ARbitrary:EDIT:NUMPoint { <NR1> | ?}

<b>Parameter/ Return parameter</b>	<b>&lt;x&gt;</b>	Channel number 1~2.
	<b>&lt;NR1&gt;</b>	Number of points

**Example**            :AWG1:ARB:EDIT:NUMP 1500  
Sets 1500 points for the arbitrary waveform.

**:AWG<x>:ARbitrary:EDIT:FUNction** (Set) →

Description	Sets the inbuilt waveform of the arbitrary waveform.	
Syntax	:AWG<x>:ARbitrary:EDIT:FUNction { SINE   SQUare   PULSe   RAMP   NOISe}	
Parameter	<x>	Channel number 1~2.
	SINE	Sine wave shape.
	SQUare	Square wave shape.
	PULSe	Pulse wave shape.
	RAMP	Ramp wave shape.
	NOISe	Noise wave shape.

**Example**                   :AWG1:ARB:EDIT:FUNCT RAMP  
 Sets a ramp shape to the arbitrary waveform.

**:AWG<x>:ARbitrary:SAVE:WAVEform** (Set) →

Description	Saves an arbitrary waveform.	
Syntax	:AWG<x>:ARbitrary:SAVe:WAVEform {ARB1  ARB2   ARB3   ARB4   <file path>}	
Parameter	<x>	Channel number 1~2.
	ARB1~4	Saves the arbitrary waveform to one of the internal memory slots.
	<file path>	Saves the arbitrary waveform to disk or USB to the specified file path. Exemple: "Disk:/xxx.UAW" "USB:/xxx.UAW"

**Example**                   :AWG1:ARB:SAVE:WAVE ARB2  
 Saves the arbitrary waveform to ARB2.

**:AWG<x>:ARbitrary:LOAD:WAVEform** (Set) →

Description	Loads an arbitrary waveform.	
Syntax	:AWG<x>:ARbitrary:LOAD:WAVEform { ARB1   ARB2   ARB3   ARB4   <file path> }	
Parameter	<x>	Channel number 1~2.
	ARB1~4	Loads the arbitrary waveform from one of the internal memory slots.
	<file path>	Loads the arbitrary waveform from disk or USB at the specified file path. Example: "Disk:/xxx.UAW" "USB:/xxx.UAW"
Example	:AWG1:ARB:LOA:WAVE ARB2 Loads the arbitrary waveform from ARB2.	

**:AWG<x>:ARbitrary:EDIT:COPY** (Set) →

Description	Copies a segment of an arbitrary waveform to a specific point.	
Syntax	:AWG<x>:ARbitrary:EDIT:COPY {<START> , <LENGth> , <PASTe> }	
Parameter	<x>	Channel number 1~2.
	<START>	NR1, point at which the segment to copy starts.
	<LENGth>	NR1, length of the segment to copy.
	<PASTe>	NR1, point at which the segment is to be copied.
Example	:AWG1:ARB:EDIT:COPY 5,100,106 Copies a segment of 100 points starting from point 5 of an arbitrary waveform and paste it to point 106 of this arbitrary waveform.	

**:AWG<x>:ARBitrary:EDIT:CLEar** (Set) →

Description	Deletes a segment of an arbitrary waveform	
Syntax	:AWG<x>:ARBitrary:EDIT:CLEar { ALL   <START> , <LENGTh> }	
Parameter	<x>	Channel number 1~2.
	ALL	Deletes the entire arbitrary waveform.
	<START>	NR1, point at which the segment to delete starts.
	<LENGTh>	NR1, length of the segment to delete.
Example	:AWG1:ARB:EDIT:CLE ALL	

**:AWG<x>:ARBitrary:EDIT:LINE** (Set) →

Description	Creates a line on an arbitrary waveform.	
Syntax	:AWG<x>:ARBitrary:EDIT:LINE { <address1> , <data1> , address2> , <data2> }	
Parameter	<x>	Channel number 1~2.
	<address1>	NR1, the point at which the line starts.
	<data1>	NRf, the value at the starting point.
	<address2>	NR1, the point at which the line ends.
	<data2>	NRf, the value at the ending point.
Example	:AWG1:ARB:EDIT:LIN 40,0.05,100,0.1 Creates a line between point 40 at value 0.05 and point 100 at value 0.01.	

**:AWG<x>:ARBitrary:EDIT:SCALE** (Set) →

Description	Sets the vertical scale of the arbitrary waveform.	
Syntax	:AWG<x>:ARBitrary:EDIT:SCALE {<NRf>}	

Parameter	<x>	Channel number 1~2.
	<NRf>	Scale 0.1~ 10
Example	:AWG1:ARB:EDIT:SCAL 5.5	

### :AWG<x>:ARBITrary:EDIT:POINt

Description	Edits a single point on an arbitrary waveform.	
Syntax	:AWG<x>:ARBITrary:EDIT:POINt {<address1> , <data1>}	
Parameter	<x>	Channel number 1~2.
	<address1>	NR1, the point to be edited.
	<data1>	NRf, the value of that point.
Example	:AWG1:ARB:EDIT:POIN 20,0.2	


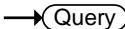
### :AWG<x>:ARBITrary:EDIT:POINt:ADD


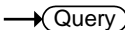
Description	Adds the edited point to the arbitrary waveform.	
Syntax	:AWG<x>:ARBITrary:EDIT:POINt:ADD {<NR1>}	
Parameter	<x>	Channel number 1~2.
	<NR1>	The point to be added.
Example	:AWG1:ARB:EDIT:POIN:ADD 20	


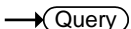
### :AWG<x>:ARBITrary:EDIT:POINt:DELEte


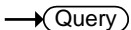
Description	Adds the edited point to the arbitrary waveform.	
Syntax	:AWG<x>:ARBITrary:EDIT:POINt:DELEte {<NR1>}	
Parameter	<x>	Channel number 1~2.
	<NR1>	The point to be deleted.
Example	:AWG1:ARB:EDIT:POIN:DELE 20	

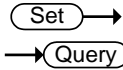
## Data Logging Commands

		 
<b>:DATALOG:STATE</b>		
Description	Sets or queries the state of the data logging app.	
Syntax	:DATALOG:STATE {OFF ON ?}	
Related commands	:DATALOG:SOURce :DATALOG:SAVe :DATALOG:INTerval :DATALOG:DURation	
Parameter/Return parameter	OFF	Turns the data logging off.
	ON	Turns the data logging on.
Example	:DATALOG:STATE ON Turns the data logging app on.	

		 
<b>:DATALOG:SOURce</b>		
Description	Sets or queries the data logging source channel.	
Syntax	:DATALOG:SOURce { CH1~CH4  all   ? }	
Related commands	:DATALOG:STATE :DATALOG:SAVe :DATALOG:INTerval :DATALOG:DURation	
Parameter/Return parameter	CH1 ~CH4	Channel 1, 2, 3, 4
	all	All displayed channels.
Example	:DATALOG:SOURce CH1 Sets the source to CH1.	

		 
<b>:DATALOG:SAVe</b>		
Description	Sets or queries the save format as image or waveform.	
Syntax	:DATALOG:SAVe {IMAGe WAVEform ?}	
Related commands	:DATALOG:STATE :DATALOG:SOURce :DATALOG:INTerval :DATALOG:DURation	
Parameter/Return parameter	IMAGe	Save as images.
	WAVEform	Save as waveforms.
Example	:DATALOG:SAVe WAVEform Sets the save format to waveform.	

		 
<b>:DATALOG:INTerval</b>		
Description	Sets or queries the interval time between each recording.	
Syntax	:DATALOG:INTerval <NRF> :DATALOG:INTerval?	
Related commands	:DATALOG:STATE :DATALOG:SOURce :DATALOG:SAVe :DATALOG:DURation	
Parameter	<NRF>	Discrete time intervals in seconds:
Example	:DATALOG:INT 2 Sets the interval time to 2 seconds.	



**:DATALOG:DURation**

Description	Sets or queries the duration time of each recording.	
Syntax	:DATALOG:DURation <NRf> :DATALOG:DURation?	
Related commands	:DATALOG:STATE :DATALOG:SOURce :DATALOG:SAVe :DATALOG:INTerval	
Parameter	<NRf>	Discrete recording time in seconds.
Example	:DATALOG:DUR 5 Sets the recording time to 5 seconds.	

## Remote Disk Commands

**:REMOTEDisk:IPADdress** 



Description	Sets or returns the IP address of remote disk.	
Syntax	:REMOTEDisk:IPADdress {<string> ?}	
Parameter/ Return parameter	<string>	IP address enclosed in double quotes. Eg., 172.16.20.255
Example	:REMOTEDisk:IPADdress "172.16.20.255" Sets the remote disk IP address as 172.16.20.255.	

**:REMOTEDisk:PATHName** 



Description	Sets or returns the file path of the remote disk.	
Syntax	:REMOTEDisk:PATHName {<string> ?}	
Parameter/ Return parameter	<string>	File path in enclosed in double quotes eg., "remote_disk"
Example	:REMOTEDisk:PATHName "remote_disk" Sets the file path to c:/remote_disk.	

**:REMOTEDisk:USERName** 



Description	Sets or queries the account username for the remote disk.	
Syntax	:REMOTEDisk:USERName {<string>   ? }	
Parameter/Return parameter	<string>	User name enclosed in double quotes eg., "User_Name".
Example	:REMOTEDisk:USERName "User_Name" Sets the account name as User_Name.	

Set →

→ Query

**:REMOTEDisk:PASSWord**

Description	Sets or queries the account password for the remote disk.	
Syntax	:REMOTEDisk:PASSWord {<string>   ? }	
Parameter/Return parameter	<string>	Username password enclosed in double quotes eg., "Password".
Example	:REMOTEDisk:PASSWord "Password" Sets the account password as Password.	

Set →

→ Query

**:REMOTEDisk:MOUNT**

Description	Turns remote disk on/off or queries its state.	
Syntax	:REMOTEDisk:MOUNT { OFF   ON   ? }	
Parameter/Return parameter	OFF	Unmount remote disk
	ON	Mount remote disk
Example	:REMOTEDisk: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.	

Set →

→ Query



**:REMOTEDisk:AUTOMount**


Description	Turns automount on/off or queries its state. The remote disk must be configured beforehand.	
Syntax	:REMOTEDisk:AUTOMount { OFF   ON   ? }	
Parameter/Return	OFF	Don't mount the remote disk at start up.

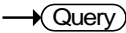
parameter	ON	Automatically mount the remote disk on start up.
Example	:REMOTEDisk:AUTOMount ON	Turns the automount function on.

---

## Spectrum Analyzer Commands

		
:SA:STATE		
Description	Sets or returns the state of the spectrum analyzer.	
Syntax	:SA:STATE {OFF ON} :SA:STATE?	
Parameter	OFF	Disable this function.
	ON	Enable this function.
Example	SA:STATE ON SA:STATE? ON	

:SA:LIST		
Description	Returns the data of the spectrum analyzer peak table.	
Syntax	:SA:LIST?	
Example	SA:LIST? NO., Frequency, Value; 1, 1.482E+07, -7.680E+01; 2, 2.790E+07, -7.600E+01; 3, 3.670E+07, -7.600E+01;	

:SA:MEMory		
Description	Returns the data in acquisition memory for the spectrum analyzer function as a header + raw data.	
Syntax	:SA:MEMory?	
Related Commands	:SA:MEMory:SOURce	
Return parameter	<string>	Returns acquisition settings followed

<waveform block data>

by raw waveform block data.

<string>

Returns the spectrum analyzer settings .

Format:

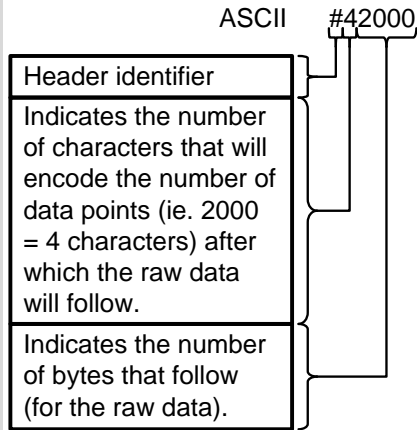
parameter(1),setting(1);parameter(2),setting(2)...parameter(n),setting(n);Waveform Data;

<waveform block data>

Header followed by the raw waveform 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 ~ 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

:SA:MEMory?

Format,2.0E;Firmware,V1.28;Time,24-Apr-17  
15:54:49;Memory  
Length,1.000E+03;Source,CH1;Probe  
Ratio,1.000E+00;Vertical Unit,dB;Vertical  
Position,3.000E+00;Vertical  
Scale,2.000E+01;Horizontal Unit,Hz;Horizontal  
Scale,1.000E+04;Sampling Period,1.000E+02;Center  
Frequency,2.300E+03;Span,1.000E+05;FREQUENCY,N  
ORM,Waveform Data;

#42000 ..... follows waveform block data in hex

The above indicates that the Memory Length is 1000.

The only exception is Waveform Data. A semicolon following Waveform Data indicates that the subsequent data consists of a raw data header followed by raw data, formatted as:

#<x><yyy><ddddddddddddd....>.

Here, "x" indicates the number of "y" elements, "y"

indicates the number of bytes in each “d” element, and “d” represents raw data where each point is represented by two bytes.

For example, #420001234567.....

indicates 2000 raw data points, where 01234567.... is the raw data.

Set →

→ Query

**:SA:MEMory:SOURce**

Description	Sets or returns the source of the waveform data	
Syntax	:SA:MEMory:SOURce {NORMal   AVErage  MAXHold   MINHold} :SA:MEMory:SOURce?	
Parameter/Return parameter	NORMal	Normal data
	AVErage	Average data
	MAXHold	Maxhold data
	MINHold	Minhold data
Example	:SA:MEMory:SOURce AVE Sets the memory source to average data.	

Set →

→ Query

**:SA:SOURce**

Description	Sets or returns the source of the spectrum analyzer	
Syntax	:SA:SOURce {CH1   CH2  CH3   CH4} :SA:SOURce?	
Parameter/Return parameter	CH1	Channel one
	CH2e	Chnanel two
	CH3	Channel three
	CH4	Channel four

Example :SA:SOURce CH2  
 Sets the source of spectrum analyzer to channel two.

**:SA<x>:SPECTRUMTrace Set →**

Description	Resets all spectrum traces.	
Syntax	SA<x>:SPECTRUMTrace {RESET}	
Parameter	RESET	Reset the trace
	<x>	1~2

Example :SA1:SPECTRUMTrace RESET  
 Reset the trace one of spectrum analyzer.

**:SA<x>:NORMAl Set →  
→ Query**

Description	Sets or returns the frequency domain Normal trace display on or off in the frequency domain graticule.	
Syntax	:SA<x>:NORMAl {ON OFF} :SA<x>t:NORMAl?	
Parameter/Return parameter	ON	Turns the normal trace display on.
	OFF	Turns the normal trace display off.
	<x>	1~2

Example :SA<1>:NORMAl ON  
 Sets the normal trace one display on.

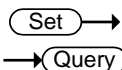
**:SA<x>:MAXHold Set →  
→ Query**

Description	Sets or returns the frequency domain Max Hold trace display on or off in the frequency domain graticule.	
-------------	--	--

Syntax :SA<x>:MAXHold {ON|OFF}  
: SA<x>:MAXHold?

Parameter/Return parameter	ON	Turns the Max Hold trace display on.
	OFF	Turns the Max Hold trace display off.
	<x>	1~2

Example : SA<1>:MAXHold OFF  
Sets the Max Hold trace one display off.



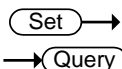
**:SA<x>:MINHold**

Description Sets or returns the frequency domain Min Hold trace display on or off in the frequency domain graticule.

Syntax : SA<x>:MINHold {ON|OFF}  
: SA<x>:MINHold?

Parameter/Return parameter	ON	Turns the Min Hold trace display on.
	OFF	Turns the Min Hold trace display off.
	<x>	1~2

Example : SA<2>:MINHold OFF  
Sets the Min Hold trace two display off.



**:SA<x>:AVERage**

Description Sets or returns the frequency domain Average trace display on or off in the frequency domain graticule.

Syntax : SA<x>:AVERage {ON|OFF}  
: SA<x>: AVERage?

Parameter/Return parameter	ON	Turns the Average trace display on.
	OFF	Turns the Average trace display off.
	<x>	1~2

Example : SA<1>: AVErage ON  
 Sets the Average trace one display on.

Set →

:SA<x>:AVErage:NUMAVg

→ Query

Description Sets or returns the number of acquisitions to be used when creating the Average frequency domain trace.

Syntax :SA<x>:AVErage:NUMAVg {<NR1>}  
 :SA<x>:AVErage:NUMAVg?

Parameter/Return parameter <NR1> The range is 2 – 256, in exponential increments.  
 <x> 1~2

Example :SA<1>:AVErage:NUMAVg 128  
 Sets the Average number of trace one to 128.

Set →

:SA<x>:DETECTIonmethod:MODE

→ Query

Description Sets or returns the detection within the oscilloscope occurs automatically or manually.

Syntax :SA<x>:DETECTIonmethod:MODE {AUTo|MANual}  
 :SA<x>:DETECTIonmethod:MODE?

Related commands :SA<x>:DETECTIonmethod:MAXHold,:SA:DETECTIonmethod:MINHold  
 :SA<x>:DETECTIonmethod:NORMAL,:SA:DETECTIonmethod:AVErage

Parameter/Return parameter AUTo Automatically mode  
 MANual Manually mode  
 <x> 1~2

Example :SA<1>:DETECTIonmethod:MODE AUTo  
 Sets the detection mode of trace one to automatic.

**:SA<x>:DETECTionmethod:MAXHold** 



Description	Sets or returns the detection method of max Hold frequency domain trace.	
Syntax	:SA<x>:DETECTionmethod:MAXHold {PLUSpeak MINUSpeak SAMPlE AVERage} :SA<x>:DETECTionmethod:MAXHold?	
Parameter/Return parameter	PLUSpeak	Sets the detection method to plus peak.
	MINUpeak	Sets the detection method to minus peak.
	SAMPlE	Sets the detection method to sample.
	AVERage	Sets the detection method to average.
	<x>	1~2
Example	:SA<1>:DETECTionmethod:MAXHold AVERage Sets the detection method of trace one to average.	

**:SA<x>:DETECTionmethod:MINHold** 



Description	Sets or returns the detection method of min Hold frequency domain trace.	
Syntax	:SA<x>:DETECTionmethod:MINHold {PLUSpeak MINUSpeak SAMPlE AVERage} :SA<x>:DETECTionmethod:MINHold?	
Parameter/Return parameter	PLUSpeak	Sets the detection method to plus peak.
	MINUpeak	Sets the detection method to minus peak.
	SAMPlE	Sets the detection method to sample.
	AVERage	Sets the detection method to average.
	<x>	1~2

Example :SA<2>:DETECTIonmethod:MINHold AVErAge  
 Sets the detection method of trace two to average.

Set →

:SA<x>:DETECTIonmethod:NORMAl

→ Query

Description Sets or returns the detection method of normal frequency domain trace.

Syntax :SA<x>:DETECTIonmethod:NORMAl  
 {PLUSpeak|MINUSpeak|SAMPlE|AVErAge}  
 :SA<x>:DETECTIonmethod:NORMAl?

Parameter/Return parameter	PLUSpeak	Sets the detection method to plus peak.
	MINUpeak	Sets the detection method to minus peak.
	SAMPlE	Sets the detection method to sample.
	AVErAge	Sets the detection method to average.
	<x>	1~2

Example :SA<1>:DETECTIonmethod:NORMAl AVErAge  
 Sets the detection method of trace one to average.

Set →

:SA<x>:DETECTIonmethod:AVErAge

→ Query

Description Sets or returns the detection method of average frequency domain trace.

Syntax :SA<x>:DETECTIonmethod:AVErAge  
 {PLUSpeak|MINUSpeak|SAMPlE|AVErAge}  
 :SA<x>:DETECTIonmethod:AVErAge?

Parameter/Return parameter	PLUSpeak	Sets the detection method to plus peak.
	MINUpeak	Sets the detection method to minus peak.
	SAMPlE	Sets the detection method to sample.

AVERage	Sets the detection method to average.
<x>	1~2

Example :SA<1>:DETECTIONmethod:AVERage AVERage  
Sets the detection method of trace one to average.

Set →  
 → Query

**:SA<x>:FREQUENCY**

Description Sets or returns the frequency (or center frequency) of the acquisition system.

Syntax :SA<x>:FREQUENCY {<NRf>|CENTER}  
:SA<x>:FREQUENCY?

Parameter/Return parameter	<NRf>	Sets the frequency by user.
	CENTER	Sets the frequency to center.
	<x>	1~2

Example SA<1>:FREQUENCY 3.0E+06  
Sets the center frequency of trace one to 3 MHz.

Set →  
 → Query

**:SA<x>:SPAN**

Description Sets or returns the span frequency setting.

Syntax :SA<x>:SPAN <NRf>  
:SA<x>:SPAN?

Parameter/Return parameter	<NRf>	Sets the span frequency by user.
----------------------------	-------	----------------------------------

Example SA<1>:SPAN 25E+06  
Sets the span frequency of trace one to 25 MH.

Set →  
 → Query

**:SA<x>:START**

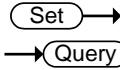
Description Sets or returns the start frequency setting.

Syntax :SA<x>:START <NRf>  
:SA<x>:START?

Parameter/Return parameter	<NRf> <x>	Sets the start frequency by user. 1~2
----------------------------	--------------	--

Example SA<1>:START -9.5E+06  
Sets the start frequency of trace one to -9.5 MHz.

:SA<x>:STOP

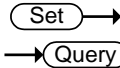


Description	Sets or returns the stop frequency setting.	
Syntax	:SA<x>:STOP <NRf> :SA<x>:STOP?	

Parameter/Return parameter	<NRf> <x>	Sets the stop frequency by user. 1~2
----------------------------	--------------	---

Example SA<1>:START 100E+06  
Sets the stop frequency of trace one to 100MHz.

:SA<x>:RBW:MODE



Description	Sets or returns the resolution bandwidth (RBW) mode, either automatic or manual.	
Syntax	:SA<x>:RBW:MODE {AUTO MANual} :SA<x>:RBW:MODE?	

Parameter/Return parameter	AUTO MANual <x>	Automatically mode. Manually mode. 1~2
----------------------------	-----------------------	--

Example SA<1>:RBW:MODE AUTO  
Sets the mode of trace one to automatic.

Set →  
 → Query

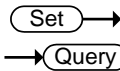
<b>:SA&lt;x&gt;:RBW</b>					
Description	Sets or returns the resolution bandwidth (RBW) when the RBW mode has been set to MANUAL (using the command SA:RBW:MODE).				
Syntax	:SA<x>:RBW <NRf> :SA<x>:RBW?				
Related commands	SA<x>:RBW:MODE				
Parameter/Return parameter	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; padding: 2px;"><b>&lt;NRf&gt;</b></td> <td style="padding: 2px;">Sets the RBW by user.</td> </tr> <tr> <td style="padding: 2px;"><b>&lt;x&gt;</b></td> <td style="padding: 2px;">1~2</td> </tr> </table>	<b>&lt;NRf&gt;</b>	Sets the RBW by user.	<b>&lt;x&gt;</b>	1~2
<b>&lt;NRf&gt;</b>	Sets the RBW by user.				
<b>&lt;x&gt;</b>	1~2				
Example	Sets SA<1>:RBW 2.0E+04 Query SA<1>:RBW? Return 1.825017e+04 If the RBW set to 20kHz, the query will return the nearestvalue (1.825017e+04).				

Set →  
 → Query

<b>:SA&lt;x&gt;:SPANRbwratio</b>									
Description	Sets or returns the resolution bandwidth (RBW) when the RBW mode has been set to AUTO (using the command SA<x>:RBW:MODE).								
Syntax	:SA<x>:SPANRbwratio {RATIO1K RATIO2K RATIO5K RATIO10K RATIO20K RATIO50K RATIO100K RATIO200K <NRf>} :SA<x>:SPANRbwratio?								
Related commands	SA<x>:RBW:MODE								
Parameter/Return parameter	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; padding: 2px;"><b>&lt;NRf&gt;</b></td> <td style="padding: 2px;">Sets the RBW by user.</td> </tr> <tr> <td style="padding: 2px;">RATIO1K</td> <td style="padding: 2px;">1000 : 1</td> </tr> <tr> <td style="padding: 2px;">RATIO2K</td> <td style="padding: 2px;">2000 : 1</td> </tr> <tr> <td style="padding: 2px;">RATIO5K</td> <td style="padding: 2px;">5000 : 1</td> </tr> </table>	<b>&lt;NRf&gt;</b>	Sets the RBW by user.	RATIO1K	1000 : 1	RATIO2K	2000 : 1	RATIO5K	5000 : 1
<b>&lt;NRf&gt;</b>	Sets the RBW by user.								
RATIO1K	1000 : 1								
RATIO2K	2000 : 1								
RATIO5K	5000 : 1								

RATIO10K	10000 : 1
RATIO20K	20000 : 1
RATIO50K	50000 : 1
RATIO100K	100000 : 1
RATIO200K	200000 : 1
<x>	1~2

**Example** :SA<1>:SPANRbwratio RATIO2K  
 Sets the ratio of trace one to 2000:1.  
 Sets :SA<1>:SPANRbwratio 2000  
 Query :SA<1>:SPANRbwratio?  
 Return RATIO2K



**:SA<x>:WINDow**

**Description** Sets or returns the windowing function, which is only used for traces.

**Syntax** :SA<x>:WINDow {RECTangular|HAMming|HANning|BLAckman}  
 :SA<x>:WINDow?

<b>Parameter/ Return parameter</b>	RECTangular	Sets to Rectangular window
	HAMming	Sets to Hamming window
	HANning	Sets to Hanning window
	BLAckman	Sets to Blackman window
	<x>	1~2

**Example** :SA<1>:WINDow HANning  
 Sets to the hanning window for the trace one.

Set →  
→ Query

**:SA<x>:UNIts**

---

Description	Sets or returns the vertical units.	
Syntax	:SA<x>:UNIts {DBV LINEAR DBM} :SA<x>:UNIts?	
Parameter/ Return parameter	DBV	Sets to DBV unit
	LINEAR	Sets to Linear unit
	DBM	Sets to DBM unit
	<x>	1~2
Example	:SA<1>:UNIts DBM Sets the unit of trace one to DBM unit.	

Set →  
→ Query

**:SA<x>:SCALE**

---

Description	Sets or returns the overall vertical scale.	
Syntax	:SA<x>:SCALE <NRf> :SA<x>:SCALE?	
Related commands	:SA<x>:UNIts	
Parameter/ Return parameter	<NRf>	Vertical scale, the value may vary which depends on the unit selected. dBm and dBV : 1, 2, 5, 10, 20 (dB) Linear: 2m, 5m, 10m, 20m, 50m, 100m, 200m, 500m, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1k (V)
	<x>	1~2
Example	:SA<1>:SCALE 2 Sets the scale of trace one to 2.	

Set →  
 → Query

**:SA:POSition**

Description	Sets or returns the overall vertical position.	
Syntax	:SA<x>:POSition <NRf> :SA<x>:POSition?	
Parameter/ Return parameter	<NRf> <x>	Vertical position range: +/-12 1~2
Example	:SA<1>:POSition 3 Sets the vertical position of trace one to 3.	

Set →  
 → Query

**:SA<x>:INPut**

Description	Sets or returns the input state of the spectrum analyzers.	
Syntax	:SA<x>:INPut {OFF ON} :SA<x>:INPut?	
Parameter	ON OFF <x>	Turn off the SA<x> input. Turn on the SA<x> input. 1~2
Example	SA2:INPut ON SA2:INPut? ON	

→ Query

**:SA:SPECTRogram:NUMSLICES?**

Description	Query the total number of slice in spectrogram.
Syntax	:SA:SPECTRogram:NUMSLICES?
Example	SA:SPECTRogram:NUMSLICES? 57

**:SA<x>:SPECTRogram:SLICESElect** 
 Set →  
 Query

Description	Set or query the selected slice in spectrogram.	
Syntax	:SA<x>:SPECTRogram:SLICESElect <NR1> :SA<x>:SPECTRogram:SLICESElect?	
Parameter	<NR1> <x>	The number of slice. Range:0 ~ -197 1~2
Example	SA2:SPECTRogram:SLICESElect -20 SA2:SPECTRogram:SLICESElect? -20	

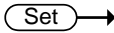
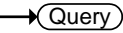
**:SA<x>:SPECTRogram:SLICETIME?**  Query

Description	Query the timestamp of selected slice.	
Syntax	:SA<x>:SPECTRogram:SLICETIME?	
Parameter	<x>	1~2
Example	SA1:SPECTRogram:SLICETIME?	

**:SA:SPECTRogram:STATe** 
 Set →  
 Query

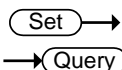
Description	Set or query the state of spectrogram display.	
Syntax	:SA:SPECTRogram:STATe {OFF ON} :SA:SPECTRogram:STATe?	
Parameter	OFF ON	Turn off the spectrogram display. Turn on the spectrogram display.
Example	SA:SPECTRogram:STATe ON SA:SPECTRogram:STATe?	

## USB Delay Command

		
:USBDelay		
Description	Sets or returns the USB delay function for the PC connection which Windows 10 installed	
Syntax	:USBDelay {OFF ON} :USBDelay?	
Parameter/ Return parameter	<ON>	Turns on the USB delay function
	<OFF>	Turns off the USB delay function
Example	:USBDelay ON Turns on the USB delay function when the scope connected with window 10 installed PC.	
Example	:REMOVEDisk:AUTOMount ON Turns the automount function on.	

## FRA Commands

**:FRA:RUN**

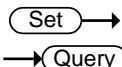


**Description** Runs the FRA function or returns the FRA state.

**Syntax** :FRA:RUN  
:FRA:RUN?

**Example** :FRA:RUN  
FRA starts.

**:FRA:STOP**

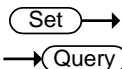


**Description** Stops the FRA function or returns the FRA state.

**Syntax** :FRA:STOP  
:FRA:STOP?

**Example** :FRA:STOP  
FRA stops.

**:FRA:SOURce:INPut**



**Description** Sets or returns the input source for FRA.

**Syntax** :FRA:SOURce:INPut {CH1~CH4}  
:FRA:SOURce:INPut?

**Parameter** CH1~CH4 Channel 1 to Channel 4

**Example** :FRA:SOURce:INPut CH1  
Set the input source as channel 1.

:FRA:SOURce:OUTPut (Set) →  
→ (Query)


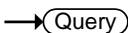
Description	Sets or returns the output source for FRA.
Syntax	:FRA:SOURce:OUTPut {CH1~CH4} :FRA:SOURce:OUTPut?
Parameter	CH1~CH4 Channel 1 to Channel 4
Example	:FRA:SOURce:OUTPut CH2 Set the input source as channel 2.



:FRA:FREQUency:STARt (Set) →  
→ (Query)

Description	Sets or returns the start frequency for FRA.
Syntax	:FRA:FREQUency:STARt {<NRF>} :FRA:FREQUency:STARt?
Parameter	<NRF> Sets the frequency to use. (Range:20Hz~25MHz)
Example	:FRA:FREQUency:STARt 100 Sets the start frequency as 100Hz.

:FRA:FREQUency:STOP (Set) →  
→ (Query)

Description	Sets or returns the stop frequency for FRA.
Syntax	:FRA:FREQUency:STOP {<NRF>} :FRA:FREQUency:STOP?
Parameter	<NRF> Sets the frequency to use. (Range:20Hz~25MHz)
Example	:FRA:FREQUency:STOP 500 Sets the start frequency as 500Hz.

		
:FRA:AWG:LOAD		
Description	Sets or returns the impedance for load.	
Syntax	:FRA:AWG:LOAD {FIFty HIGHZ} :FRA:AWG:LOAD?	
Parameter	FIFty	50 ohm
	HIGHZ	High impedance
Example	:FRA:AWG:LOAD HIGHZ Sets the load as high impedance.	

		
:FRA:AWG:AMplitude		
Description	Sets or returns the amplitude for FRA. When amplitude profile is on, sets or returns the amplitude for the selected frequency.	
Syntax	:FRA:AWG:AMplitude {<NRf>,[<range>]} :FRA:AWG:AMplitude? [<range>]	
Parameter	<NRf>	Sets the amplitude to user.(Load=FIFTY, Range:0.01Vpp~2.5Vpp; Load=HIGHZ, Range:0.02Vpp~5Vpp)
	<range>	The selected frequency. {F20hz   F100hz   F1Khz   F10Khz   F100Khz   F1Mhz   F10Mhz   F25Mhz} F20hz: Frequency range >20Hz (The default <range>). F100hz: Frequency range >100Hz. F1Khz: Frequency range >1kHz. F10Khz: Frequency range >10kHz. F100Khz: Frequency range >100kHz. F1Mhz: Frequency range >1MHz. F10Mhz: Frequency range >10MHz.

F25Mhz: Frequency range 25MHz.

Example :FRA:AWG:AMPlitude 0.2  
 Sets the amplitude as 0.2Vpp.  
 :FRA:AWG:AMPlitude 0.5,F100HZ  
 :FRA:AWG:AMPlitude? F100HZ  
 0.5

Set →

→ Query

**:FRA:POINt**

Description Sets or returns the number of processing points in a decade.

Syntax :FRA:POINt {<NR1>}  
 :FRA:POINt?

Parameter <NR1> The number of points in a decade.  
 (Range:10, 15, 30, 45, 90)

Example :FRA:POINt 15  
 Sets the number of processing points as 15 in a decade.

**:FRA:SAVe**

Set →

Description Saves the FRA result.

Syntax :FRA:SAVe

Example :FRA:SAVe  
 Saves the result to default file.

**:FRA:RECALL**

Set →

Description Recalls the FRA result from memory or USB.

Syntax :FRA:RECALL {<file path>  
 ("Disk:/xxx.FRD", "USB:/xxx.FRD")}

Parameter xxx.FRD Filename

Example :FRA:RECALL "Disk:/FRA1.FRD"  
 Recalls a FRA result named FRA1.FRD from root directory (Disk:/) of the scope.

:FRA:DATA → Query

Description Shows the detailed information of FRA settings and results.

Syntax :FRA:DATA?

Example :FRA:DATA?  
 Shows the FRA result's detail.

:FRA:SAVETOCsv Set →

Description Saves the FRA result as a CSV file.

Syntax :FRA:SAVETOCsv

Example :FRA: SAVETOCsv  
 Saves results as CSV file.

:FRA:AWG:AMPlitude:PROFile Set →  
→ Query

Description Sets or returns the state of amplitude profile.

Syntax :FRA:AWG:AMPlitude:PROFile {ON|OFF}  
 :FRA:AWG:AMPlitude:PROFile?

Parameter	ON	Turn on the profile.
	OFF	Turn off the profile.

Example :FRA:AWG:AMPlitude:PROFile ON  
 :FRA:AWG:AMPlitude:PROFile?  
 ON

:FRA:AWG:AMPlitude:INTERPolation → Set →  
→ Query

Description	Sets or returns the state of linear interpolation for the selected frequency.	
Syntax	:FRA:AWG:AMPlitude:INTERPolation {ON OFF,[<range>]} :FRA:AWG:AMPlitude:INTERPolation? [<range>]	
Parameter	ON	Turn off the interpolation.
	OFF	Turn off the interpolation.
	<range>	The selected frequency. {F20hz   F100hz   F1Khz   F10Khz   F100Khz   F1Mhz   F10Mhz   F25Mhz} F20hz: Frequency range >20Hz(The default <range>). F100hz: Frequency range >100Hz. F1Khz: Frequency range >1kHz. F10Khz: Frequency range >10kHz. F100Khz: Frequency range >100kHz. F1Mhz: Frequency range >1MHz. F10Mhz: Frequency range >10MHz. F25Mhz: Frequency range 25MHz.
Example	:FRA:AWG:AMPlitude:INTERPolation ON,F100HZ :FRA:AWG:AMPlitude:INTERPolation? F100HZ ON	

:FRA:DATA:GMARgin → Query

Description	Returns the gain margin of FRA results.
Syntax	:FRA:DATA:GMARgin?

**:FRA:DATA:GMARgin:FREQuency** → Query

Description Returns the gain margin frequency of FRA results.

Syntax :FRA:DATA:GMARgin:FREQuency?

**:FRA:DATA:PMARgin** → Query

Description Returns the phase margin of FRA results.

Syntax :FRA:DATA:PMARgin?

**:FRA:DATA:PMARgin:FREQuency** → Query

Description Returns the phase margin frequency of FRA results.

Syntax :FRA:DATA:PMARgin:FREQuency?

**:FRA:STATe** → Query

Description Query or turn on/off the FRA function.


Syntax :FRA:STATe {ON|OFF}  
:FRA:STATe?

Parameter	ON	Turn on the FRA. (No work when PWR function running)
	OFF	Turn off the FRA.

Example :FRA:STATe ON  
:FRA:STATe?  
ON

## Power Supply Commands


:POWERSupply:OUTPut<X> (Set) →  
→ (Query)

Description	Sets or returns the power supply output.	
 Note	Power supply output won't turn on when OCP occurs or power supply not yet config.	
Syntax	:POWERSupply:OUTPut<X> {ON OFF} :POWERSupply:OUTPut<X>?	
Parameter/ Return parameter	ON	Turns on the power supply output.
	OFF	Turns off the power supply output.
	<X>	range:1~2
Example	:POWERSupply:OUTPut1 ON	

:POWERSupply:OUTPut<X>:VOLTage (Set) →  
→ (Query)

Description	Sets or returns the power supply voltage.	
Syntax	:POWERSupply:OUTPut<X>:VOLTage <NR3> :POWERSupply:OUTPut<X>:VOLTage?	
Parameter/ Return parameter	<NR3>	Sets the voltage.(range:1~5)
	<X>	Range:1~2
Example	POWERSupply:OUTPut1:VOLTage 3.3	

:POWERSupply:OUTPut<X>:RECONFigure (Set) →  
→ (Query)

Description	Reconfigure the power supply.	
 Note	Clear OCP flag.	
Syntax	:POWERSupply:OUTPut<X>:RECONFigure {ON}	
Parameter	ON	Reconfigure the power supply.

<X> range:1~2

Example :POWERSupply:OUTPut1:RECONFigure ON

:POWERSupply:OUTPut<X>:OCP Set →  
→ Query

Description Returns the power supply OCP.


Syntax :POWERSupply:OUTPut<X>:OCP?

Return parameter <X> Range:1~2

Example :POWERSupply:OUTPut1:OCP?  
 Return ON ->Output1 OCP occurs.  
 Return OFF ->Output1 OCP did not occur.

:POWERSupply:OUTPut<X>:CONFigure Set →  
→ Query

Description Configur the power supply.

 Note It is must delay about 6 sec after power supply config.

Syntax :POWERSupply:CONFigure {ON|?}

Parameter ON Range:1~2

Example :POWERSupply:CONFigure ON

## Python Commands

Set →  
 → Query

**:PYDEcode:TYPE**

Description	Set or return the state and decode type of python bus decode application.	
Syntax	:PYDEcode:TYPE {CANFD FLEXRay USBPD USBPACKET I2S OFF} :PYDEcode:TYPE?	
Parameter/ Return parameter	CANFD	Turn on the bus decode operation and set the decode type as CANFD.
	FLEXRay	Turn on the bus decode operation and set the decode type as flexray.
	USBPD	Turn on the bus decode operation and set the decode type as USBPD.
	USBPACKET	Turn on the bus decode operation and set the decode type as USB packet.
	I2S	Turn on the bus decode operation and set the decode type as I2S.
	OFF	Turn off the bus decode operation.
Example	:PYDEcode:TYPE CANFD :PYDEcode:TYPE? CANFD	

Set →  
 → Query

**:PYDEcode:CANFD:SOURce**

Description	Set or return the input source of the CANFD bus.	
Syntax	:PYDEcode:CANFD:SOURce {CH1 CH2 CH3 CH4} :PYDEcode:CANFD:SOURce?	
Parameter/ Return parameter	CH1 ~ CH4	Set the analog channel to use as the CANFD input.
Example	:PYDEcode:CANFD:SOURce CH2 :PYDEcode:CANFD:SOURce? CH2	

Set →  
 → Query

**:PYDEcode:CANFD:PROBe**

Description	Set or return the signal type of the CANFD.	
Syntax	:PYDEcode:CANFD:PROBe {CANH CANL} :PYDEcode:CANFD:PROBe?	
Parameter/ Return parameter	CANH	Specifies the CANH signal.
	CANL	Specifies the CANL signal.
Example	:PYDEcode:CANFD:PROBe CANH :PYDEcode:CANFD:PROBe? CANH	

→ Query

**:PYDEcode:CANFD:SAMPLEpoint**

Description	Returns the sample point of the CANFD.	
Syntax	:PYDEcode:CANFD:SAMPLEpoint?	
Example	:PYDEcode:CANFD:SAMPLEpoint? 50	

Set →  
 → Query

**:PYDEcode:CANFD:BITRate**

Description	Set or return the bit rate for CANFD.	
Syntax	:PYDEcode:CANFD:BITRate {<NR1>} :PYDEcode:CANFD:BITRate?	
Parameter/ Return parameter	<NR1>	The bit rate.
Example	:PYDEcode:CANFD:BITRate 125000 :PYDEcode:CANFD:BITRate? 125000	

Set →  
 → Query

**:PYDEcode:CANFD:FASTBITRate**

Description	Set or return the fast bit rate for CANFD.	
Syntax	:PYDEcode:CANFD:FASTBITRate {<NR1>} :PYDEcode:CANFD:FASTBITRate?	

Parameter/ Return parameter	<NR1>	The fast bit rate.
--------------------------------	-------	--------------------

Example :PYDEcode:CANFD: FASTBITRate 5000000  
:PYDEcode:CANFD: FASTBITRate? 5000000

(Set) →

:PYDEcode:FLEXRay:SOURce ← (Query)

Description	Set or return the input source of the flexray bus.	
Syntax	:PYDEcode:FLEXRay:SOURce {CH1 CH2 CH3 CH4} :PYDEcode:FLEXRay:SOURce?	

Parameter/ Return parameter	CH1 ~ CH4	Set the analog channel to use as the flexray input.
--------------------------------	-----------	---

Example :PYDEcode:FLEXRay:SOURce CH2  
:PYDEcode:FLEXRay:SOURce? CH2

(Set) →

:PYDEcode:FLEXRay:BDIFF ← (Query)

Description	Set or return the signal type of the flexray.	
Syntax	:PYDEcode:FLEXRay:BDIFF {NORMal INVerted} :PYDEcode:FLEXRay:BDIFF?	

Parameter/ Return parameter	NORMal	Specifies the normal signal.
	INVerted	Specifies the inverted signal.

Example :PYDEcode:FLEXRay:BDIFF INVerted  
:PYDEcode:FLEXRay:BDIFF? INVerted

(Set) →

:PYDEcode:FLEXRay:CHTYPe ← (Query)

Description	Set or return the channel type of the flexray.	
Syntax	:PYDEcode:FLEXRay:CHTYPe {A B} :PYDEcode:FLEXRay:CHTYPe?	

Parameter/ Return parameter	A	Set the channel type as A.
	B	Set the channel type as B.

Example :PYDEcode:FLEXRay:CHTYPe A  
:PYDEcode:FLEXRay:CHTYPe? A

:PYDEcode:FLEXRay:BITRate (Set) →  
→ (Query)

Description Set or return the bit rate for flexray..

Syntax :PYDEcode:FLEXRay:BITRate {<NR1>}  
:PYDEcode:FLEXRay:BITRate?

Parameter/ Return parameter <NR1> The bit rate.

Example :PYDEcode:FLEXRay:BITRate 5000000  
:PYDEcode:FLEXRay:BITRate? 5000000

:PYDEcode:USBPD:SOURce (Set) →  
→ (Query)

Description Set or return the input source of the USBPD bus.

Syntax :PYDEcode:USBPD:SOURce {CH1|CH2|CH3|CH4}  
:PYDEcode:USBPD:SOURce?

Parameter/ Return parameter CH1 ~ CH4 Set the analog channel to use as the USBPD input.

Example :PYDEcode:USBPD:SOURce CH1  
:PYDEcode:USBPD:SOURce? CH1

:PYDEcode:USBPACKET:SIGnal (Set) →  
→ (Query)

Description Set or return the signal type of the USB packet.

Syntax :PYDEcode:USBPACKET:SIGnal  
{AUTOMatic|LOWSpeed|FULLSpeed}  
:PYDEcode:USBPACKET:SIGnal?

Parameter/ Return parameter AUTOMatic Set the signal type as automatic.  
LOWSpeed Set the signal type as low speed.  
FULLSpeed Set the signal type as full speed.

Example :PYDEcode:USBPACKET:SIGnal FULLSpeed  
 :PYDEcode:USBPACKET:SIGnal? FULLSpeed

:PYDEcode:USBPACKET:DPlus  

Description Set or return the D+ input source of the USB packet bus.

Syntax :PYDEcode:USBPACKET:DPlus {CH1|CH2|CH3|CH4}  
 :PYDEcode:USBPACKET:DPlus?

Parameter/ Return parameter CH1 ~ CH4 Set the analog channel to use as the USB packet D+ input.

Example :PYDEcode:USBPACKET:DPlus CH1  
 :PYDEcode:USBPACKET:DPlus? CH1

:PYDEcode:USBPACKET:DMinus  

Description Set or return the D- input source of the USB packet bus.

Syntax :PYDEcode:USBPACKET:DMinus {CH1|CH2|CH3|CH4}  
 :PYDEcode:USBPACKET:DMinus?

Parameter/ Return parameter CH1 ~ CH4 Set the analog channel to use as the USB packet D- input.

Example :PYDEcode:USBPACKET:DMinus CH1  
 :PYDEcode:USBPACKET:DMinus? CH1

:PYDEcode:I2S:SCK  

Description Set or return the SCK input source of the I2S bus.

Syntax :PYDEcode:I2S:SCK {CH1|CH2|CH3|CH4}  
 :PYDEcode:I2S:SCK?

Parameter/ Return parameter CH1 ~ CH4 Set the analog channel to use as the I2S SCK input.

Example :PYDEcode:I2S:SCK CH3  
:PYDEcode:I2S:SCK? CH3

:PYDEcode:I2S:WS

Set →

→ Query

Description Set or return the WS input source of the I2S bus.

Syntax :PYDEcode:I2S:WS {CH1|CH2|CH3|CH4}  
:PYDEcode:I2S:WS?

Parameter/  
Return parameter CH1 ~ CH4 Set the analog channel to use as the I2S  
WS input.

Example :PYDEcode:I2S:WS CH1  
:PYDEcode:I2S:WS? CH1

:PYDEcode:I2S:SD

Set →

→ Query

Description Set or return the SD input source of the I2S bus.

Syntax :PYDEcode:I2S:SD {CH1|CH2|CH3|CH4}  
:PYDEcode:I2S:SD?

Parameter/  
Return parameter CH1 ~ CH4 Set the analog channel to use as the I2S  
SD input.

Example :PYDEcode:I2S:SD CH1  
:PYDEcode:I2S:SD? CH1

# APPENDIX

## Error messages

Description      The following error messages may be returned from the :SYSTem:ERRor? query. For details see page 182.

List of error messages	Error number, "Error Description"
	+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"