

# Mixed-Domain Oscilloscope

MDO-2000E 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
<b>COMMAND OVERVIEW .....</b>	<b>15</b>
Command Syntax .....	15
List of Commands in Functional Order ....	17
<b>COMMAND DETAILS .....</b>	<b>34</b>
Common Commands .....	36
Acquisition Commands .....	42
Autoscale Commands .....	50
Vertical Commands .....	51
Math Commands .....	57
Cursor Commands .....	66
Display Commands .....	77
Hardcopy Commands .....	81
Measure Commands .....	84
Measurement Commands .....	110
Reference Commands .....	117
Run Command .....	120
Stop Command .....	120
Single Command .....	120
Force Command .....	121
Timebase Commands .....	122
Trigger Commands .....	125
System Commands .....	167
Save/Recall Commands .....	168
Ethernet Command .....	173
Time Command .....	173
Bus Decode Commands .....	174
Mark Commands .....	190

Search Commands .....	192
Label Commands .....	225
Segment Commands .....	233
DVM Commands.....	242
Go_NoGo Commands .....	244
Mask Commands .....	250
AWG Commands.....	256
Data Logging Commands.....	277
Remote Disk Commands.....	280
DMM Commands.....	283
Spectrum Analyzer Commands .....	288
Power Supply Commands.....	302
USB Delay Command .....	304
Digital Commands .....	305
<b>APPENDIX.....</b>	<b>313</b>
Error messages .....	313
<b>INDEX.....</b>	<b>317</b>

# INTERFACE OVERVIEW

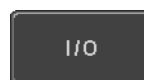
This manual describes how to use the MDO-2000E's remote command functionality and lists the command details. The Overview chapter describes how to configure the USB and Ethernet remote control interfaces.

## Interface Configuration

### Configure USB Interface

USB Configuration	PC side connector	Type A, host
	MDO-2000E side connector	Type B, device
	Speed	1.1/2.0
	USB Class	CDC (communications device class)

- Panel Operation
1. Press the Utility key.
  2. Press *I/O* from the bottom menu.
  3. Press *USB Device Port* from the side menu and select *Computer*.
  4. Connect the USB cable to the rear panel device port.



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

## USB Functionality Check

---

Terminal Application

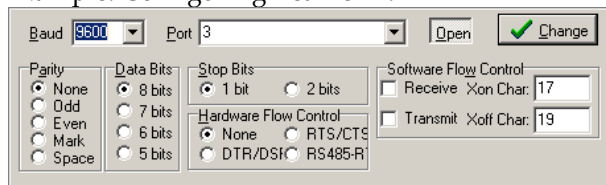
Invoke a terminal application such as RealTerm.

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

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

*Control panel* → *Hardware and Sound* → *Device Manager*

Example: Configuring RealTerm:



Functionality Check

Key in this query command via the terminal application.

`*idn?`

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

`GW,MDO-2202E,PXXXXXX,V1.00`

## 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 9.

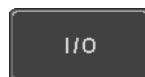
**Panel Operation** 1. Connect the Ethernet cable to the LAN port on the rear panel.



2. Press the *Utility* key.



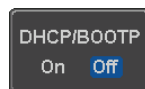
3. Press *I/O* from the bottom menu.



4. Press *Ethernet* from the side menu.

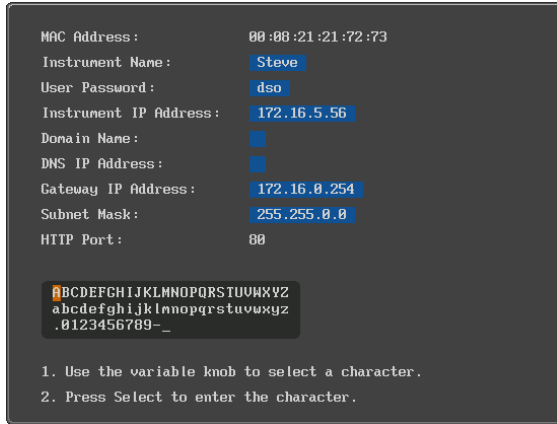


5. Set *DHCP/BOOTP* to *On* or *Off* from the side menu.

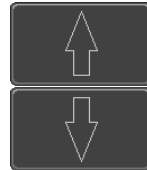


Note

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



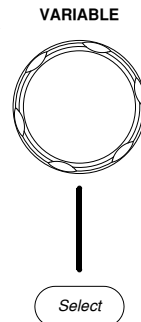
6. Use the *Up* and *Down* arrows on the side menu to navigate to each Ethernet configuration item.



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

Note: HTTP Port is fixed at 80.

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

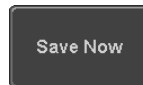




Press *Backspace* to delete a character.



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



## Configure Socket Server

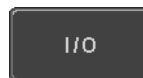
The MDO-2000E supports socket server functionality for direct two-way communication with a client PC or device over LAN. By default, the Socket Server is off.

Configure Socket Server 1. Configure the IP address for the MDO-2000E. Page 7

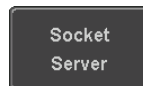
2. Press the *Utility* key.



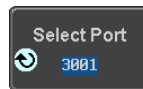
3. Press *I/O* from the bottom menu.



4. Press *Socket Server* from the side menu.

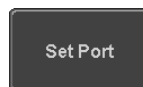


5. Press *Select Port* and choose the port number with the Variable knob.



Range 1024~65535

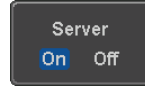
6. Press *Set Port* to confirm the port number.



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




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



## Socket Server Functionality Check

---

NI Measurement and Automation Explorer To test the socket server functionality, National Instruments Measurement and Automation Explorer can be used. This program is available on the NI website, [www.ni.com](http://www.ni.com).

- |           |  |   |
|-----------|--|---|
| Operation | 1. Configure the IP address for the MDO-2000E.   | Page 7  |
|           | 2. Configure the socket port.  | Page 9  |
|           | 3. Start the NI Measurement and Automation Explorer (MAX) program. Using Windows, press: |  |

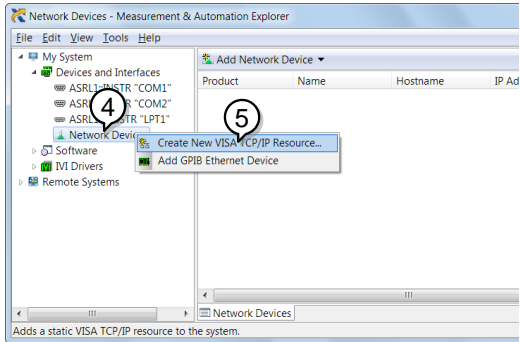
*Start>All Programs>National Instruments>Measurement & Automation*



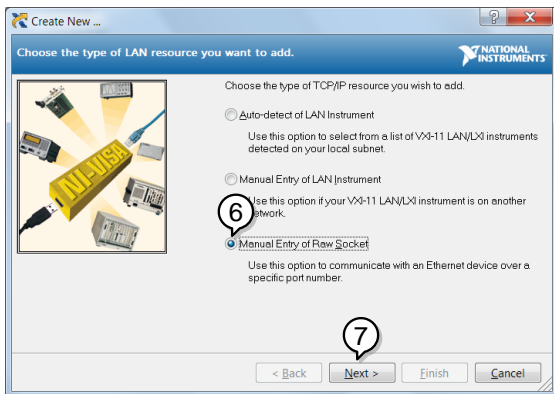
- From the Configuration panel access;
 

*My System>Devices and Interfaces>Network Devices*

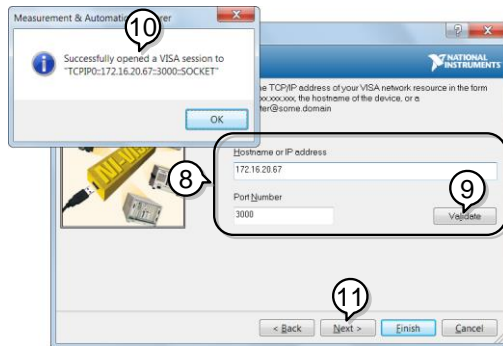
5. Right click *Network Devices* and select *Create New Visa TCP/IP Resource...*



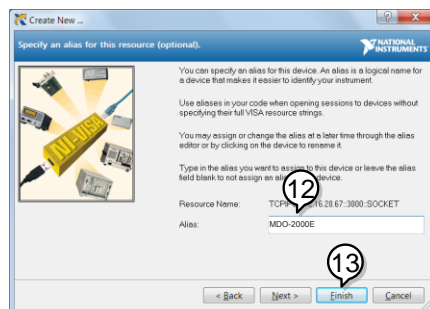
6. Select *Manual Entry of Raw Socket* from the popup window.
7. Click *Next*.



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



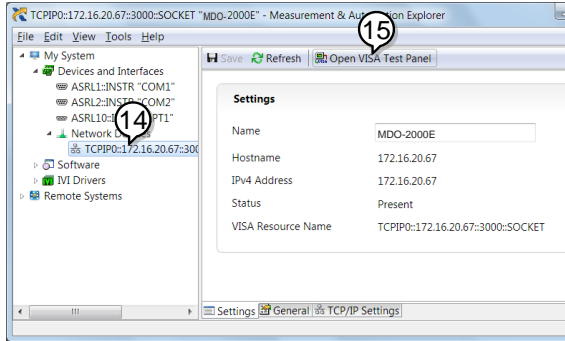
12. Choose an alias for the socket connection if you like.
13. Click *Finish* to finish the configuration.



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

Functionality  
Check

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

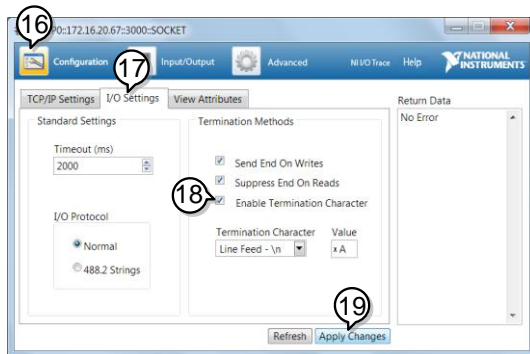


- Click on the *Configuration* icon.

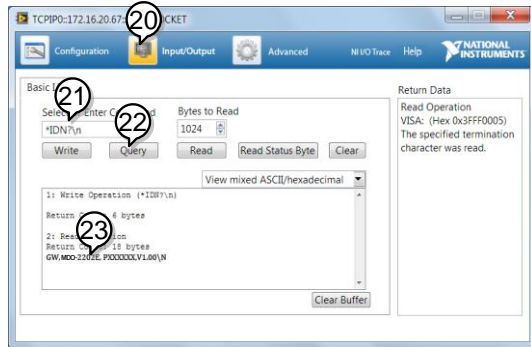
- Select the *I/O Settings* tab.

- Mark the *Enable Termination Character* checkbox. Make sure the termination character is a line feed (/n, value: xA).

- Click *Apply Changes*.



20. Click the *Input/Output* icon.
21. Make sure *\*IDN?* query is selected in the *Select or Enter Command* drop box.
22. Click on *Query*.
23. The manufacturer, model number, serial number and firmware version will be displayed in the buffer. For example:  
GW,MDO-2202E,PXXXXXX,V1.00



# C COMMAND OVERVIEW

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

## Command Syntax

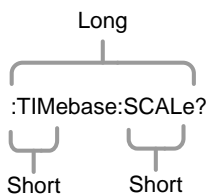
---

Compatible standard

- USB CDC\_ACM compatible
- SCPI, 1994 (partially compatible)

Command forms

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



The commands can be written in capitals or 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  
 2: single space  
 3: parameter  
 4: message terminator

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

Note Commands are non-case sensitive.



## List of Commands in Functional Order

Common commands	*IDN? .....	36
	*LRN? .....	36
	*SAV .....	37
	*RCL .....	37
	*RST .....	38
	*CLS .....	38
	*ESE .....	38
	*ESR .....	39
	*OPC .....	39
	*SRE .....	40
	*STB .....	41
Acquisition commands	:ACQuire:AVERAge .....	42
	:ACQuire:MODE .....	43
	:ACQuire<X>:MEMory? .....	43
	:ACQuire:FILTer:SOURce .....	45
	:ACQuire:FILTer .....	46
	:ACQuire:FILTer:FREQuency .....	46
	:ACQuire:FILTer:FREQuency:UPPER .....	46
	:ACQuire:FILTer:FREQuency:LOWER .....	47
	:ACQuire:FILTer:TYPe .....	47
	:ACQuire:FILTer:TRACking .....	48
	:ACQuire<X>:STATE? .....	48
	:ACQuire:INTERpolation .....	48
	:ACQuire:RECOrdlength .....	49
:HEADer .....	49	
Autoscale commands	:AUTOSet .....	50
	:AUTORSET:MODE .....	50
Vertical Scale commands	:CHANnel<X>:BWLimit .....	51
	:CHANnel<X>:COUPling .....	52
	:CHANnel<X>:DESKew .....	52

	:CHANnel<X>:DISPlay .....	52
	:CHANnel<X>:EXPand .....	53
	:CHANnel<X>:IMPedance? .....	53
	:CHANnel<X>:INVert .....	54
	:CHANnel<X>:POSition .....	54
	:CHANnel<X>:PROBe:RATio .....	55
	:CHANnel<X>:PROBe:TYPe .....	55
	:CHANnel<X>:SCALe .....	55
<hr/>		
Math commands	:MATH:DISP .....	57
	:MATH:TYPe .....	58
	:MATH:DUAL:SOURce<X> .....	58
	:MATH:DUAL:OPERator .....	58
	:MATH:DUAL:POSition .....	59
	:MATH:DUAL:SCALe .....	59
	:MATH:FFT:SOURce .....	60
	:MATH:FFT:MAG .....	60
	:MATH:FFT:WINDow .....	60
	:MATH:FFT:POSition .....	61
	:MATH:FFT:SCALe .....	61
	:MATH:FFT:HORizontal:SCALe .....	62
	:MATH:FFT:HORizontal:POSition .....	62
	:MATH:DEFine .....	62
	:MATHVAR? .....	63
	:MATHVAR:VAR<X> .....	64
	:MATH:ADVanced:POSition .....	64
	:MATH:ADVanced:SCALe .....	64
<hr/>		
Cursor commands	:CURSor:MODE .....	67
	:CURSor:SOURce.....	67
	:CURSor:HUNI .....	68
	:CURSor:HUSE .....	68
	:CURSor:VUNI .....	69
	:CURSor:VUSE .....	69
	:CURSor:DDT.....	69
	:CURSor:H1Position .....	70

	:CURSor:H2Position .....	70
	:CURSor:HDELta .....	71
	:CURSor:V1Position.....	71
	:CURSor:V2Position.....	71
	:CURSor:VDELta .....	72
	:CURSor:XY:RECTangular:X:POSition<X> .....	72
	:CURSor:XY:RECTangular:X:DELta .....	72
	:CURSor:XY:RECTangular:Y:POSition<X> .....	73
	:CURSor:XY:RECTangular:Y:DELta .....	73
	:CURSor:XY:POLar:RADIUS:POSition<X>.....	73
	:CURSor:XY:POLar:RADIUS:DELta .....	74
	:CURSor:XY:POLar:THETA:POSition<X> .....	74
	:CURSor:XY:POLar:THETA:DELta.....	74
	:CURSor:XY:PRODUct:POSition<X> .....	75
	:CURSor:XY:PRODUct:DELta.....	75
	:CURSor:XY:RATio:POSition<X> .....	75
	:CURSor:XY:RATio:DELta .....	76
<hr/>		
Display commands	:DISPlay:INTensity:WAVEform.....	77
	:DISPlay:INTensity:GRATICule.....	77
	:DISPlay:INTensity:BACKLight .....	78
	:DISPlay:INTensity:BACKLight:AUTODim:ENABle ..	78
	:DISPlay:INTENSITY:BACKLight:AUTODim:TIME ...	78
	:DISPlay:PERSistence .....	79
	:DISPlay:GRATICule .....	79
	:DISPlay:WAVEform.....	80
	:DISPlay:OUTPut .....	80
<hr/>		
Hardcopy commands	:HARDcopy:START .....	81
	:HARDcopy:MODE .....	81
	:HARDcopy:PRINTINKSaver .....	82
	:HARDcopy:SAVEINKSaver .....	82
	:HARDcopy:SAVEFORMat .....	82
	:HARDcopy:ASSIGN .....	83

Measure	:MEASure:GATing.....	85
commands	:MEASure:SOURce .....	85
	:MEASure:METHod .....	86
	:MEASUrement:REFLevel:PERCent:HIGH .....	86
	:MEASUrement:REFLevel:PERCent:LOW .....	87
	:MEASUrement:REFLevel:PERCent:MID .....	87
	:MEASUrement:REFLevel:PERCent:MID2 .....	87
	:MEASure:FALL.....	88
	:MEASure:FOVShoot.....	88
	:MEASure:FPreshoot .....	89
	:MEASure:FREQuency .....	89
	:MEASure:NWIDth .....	90
	:MEASure:PDUTy .....	90
	:MEASure:PERiod .....	91
	:MEASure:PWIDth .....	91
	:MEASure:RISe .....	92
	:MEASure:ROVShoot.....	93
	:MEASure:RPreshoot.....	93
	:MEASure:PPULSE.....	94
	:MEASure:NPULSE .....	94
	:MEASure:PEDGE .....	95
	:MEASure:NEDGE .....	95
	:MEASure:AMPlitude.....	96
	:MEASure:MEAN .....	97
	:MEASure:CMEan .....	97
	:MEASure:HIGH .....	98
	:MEASure:LOW.....	98
	:MEASure:MAX.....	99
	:MEASure:MIN .....	99
	:MEASure:PK2PK .....	100
	:MEASure:RMS .....	100
	:MEASure:CRMS.....	101
	:MEASure:AREa .....	102
	:MEASure:CARea .....	102
:MEASure:FRRDelay .....	103	

---

	:MEASure:FRFDelay.....	103
	:MEASure:FFRDelay.....	104
	:MEASure:FFFDelay.....	105
	:MEASure:LRRDelay.....	105
	:MEASure:LRFDelay.....	106
	:MEASure:LFRDelay.....	106
	:MEASure:LFFDelay.....	107
	:MEASure:PHAsE.....	108
	:MEASure:PFLI.....	108
	:MEASure:FLI.....	109
Measurement commands	:MEASUrement:MEAS<X>:SOURCE<X>.....	110
	:MEASUrement:MEAS<X>:TYPe.....	111
	:MEASUrement:MEAS<X>:STATE.....	111
	:MEASUrement:MEAS<X>:VALue.....	112
	:MEASUrement:MEAS<X>:MAXimum.....	113
	:MEASUrement:MEAS<X>:MEAN.....	113
	:MEASUrement:MEAS<X>:MINimum.....	114
	:MEASUrement:MEAS<X>:STDdev.....	115
	:MEASUrement:STATIstics:MODe.....	115
	:MEASUrement:STATIstics:WEIghting.....	116
	:MEASUrement:STATIstics.....	116
Reference commands	:REF<X>:DISPlay.....	117
	:REF<X>:TIMebase:POSition.....	117
	:REF<X>:TIMebase:SCALE.....	118
	:REF<X>:OFFSet.....	118
	:REF<x>:SCALE.....	119
Run command	:RUN.....	120
Stop command	:STOP.....	120
Single command	:SINGLE.....	120
Force command	:FORCe.....	121

---

Timebase	:TIMebase:EXPand .....	122
commands	:TIMebase:POSition .....	122
	:TIMebase:SCALe .....	122
	:TIMebase:MODE .....	123
	:TIMebase:WINDow:POSition .....	123
	:TIMebase:WINDow:SCALe .....	124
<hr/>		
Trigger	:TRIGger:FREQUency.....	127
commands	:TRIGger:TYPe .....	127
	:TRIGger:SOURce .....	128
	:TRIGger:COUPlE.....	128
	:TRIGger:NREJ .....	129
	:TRIGger:MODE.....	129
	:TRIGger:HOLDoff.....	129
	:TRIGger:LEVel.....	130
	:TRIGger:HLEVel .....	130
	:TRIGger:LLEVel.....	131
	:TRIGger:EDGE:SLOP.....	131
	:TRIGger:DELaY:SLOP .....	132
	:TRIGger:DELaY:TYPe .....	132
	:TRIGger:DELaY:TIME.....	132
	:TRIGger:DELaY:EVENT.....	133
	:TRIGger:DELaY:LEVel .....	133
	:TRIGger:PULSEwidth:POLarity .....	133
	:TRIGger:RUNT:POLarity .....	134
	:TRIGger:RUNT:WHEn.....	134
	:TRIGger:RUNT:TIME.....	135
	:TRIGger:RISEFall:SLOP.....	135
	:TRIGger:RISEFall:WHEn .....	136
	:TRIGger:RISEFall:TIME .....	136
	:TRIGger:VIDeo:TYPe .....	137
	:TRIGger:VIDeo:FIELD .....	137
	:TRIGger:VIDeo:LINE .....	138
	:TRIGger:VIDeo:POLarity .....	138

---

---

:TRIGger:PULSe:WHEn .....	138
:TRIGger:PULSe:TIME .....	139
:TRIGger:TIMEOut:WHEn .....	139
:TRIGger:TIMEOut:TIMER .....	140
:TRIGger:ALTErnate .....	140
:TRIGger:STATe .....	141
:TRIGger:EXTERnal:PROBe:TYPe .....	141
:TRIGger:EXTERnal:PROBe:RATio .....	142
:TRIGger:BUS:TYPe .....	142
:TRIGger:BUS:THReshold:CH<x> .....	143
:TRIGger:BUS:B1:I2C:CONDition .....	143
:TRIGger:BUS:B1:I2C:ADDReSS:MODE .....	144
:TRIGger:BUS:B1:I2C:ADDReSS:TYPe .....	144
:TRIGger:BUS:B1:I2C:ADDReSS:VALue .....	145
:TRIGger:BUS:B1:I2C:ADDReSS:DIRectioN .....	146
:TRIGger:BUS:B1:I2C:DATA:SIZE .....	146
:TRIGger:BUS:B1:I2C:DATA:VALue .....	147
:TRIGger:BUS:B1:UART:CONDition .....	147
:TRIGger:BUS:B1:UART:RX:DATA:SIZE .....	148
:TRIGger:BUS:B1:UART:RX:DATA:VALue .....	149
:TRIGger:BUS:B1:UART:TX:DATA:SIZE .....	149
:TRIGger:BUS:B1:UART:TX:DATA:VALue .....	150
:TRIGger:BUS:B1:SPI:CONDition .....	151
:TRIGger:BUS:B1:SPI:DATA:SIZE .....	151
:TRIGger:BUS:B1:SPI:DATA:MISO:VALue .....	152
:TRIGger:BUS:B1:SPI:DATA:MOSI:VALue .....	152
:TRIGger:BUS:B1:CAN:CONDition .....	153
:TRIGger:BUS:B1:CAN:FRAMeType .....	154
:TRIGger:BUS:B1:CAN:IDentifier:MODE .....	154
:TRIGger:BUS:B1:CAN:IDentifier:VALue .....	155
:TRIGger:BUS:B1:CAN:IDentifier:DIRectioN .....	155
:TRIGger:BUS:B1:CAN:DATA:QUALifier .....	156
:TRIGger:BUS:B1:CAN:DATA:SIZE .....	157
:TRIGger:BUS:B1:CAN:DATA:VALue .....	157
:TRIGger:BUS:B1:LIN:CONDition .....	158

---

	:TRIGger:BUS:B1:LIN:DATA:QUALifier .....	159
	:TRIGger:BUS:B1:LIN:DATA:SIZE .....	159
	:TRIGger:BUS:B1:LIN:DATA:VALue .....	160
	:TRIGger:BUS:B1:LIN:ERRTYpe.....	161
	:TRIGger:BUS:B1:LIN:IDentifier:VALue .....	161
	:TRIGger:BUS:B1:PARAllel:VALue .....	162
	:TRIGger:LOGic:INPut:CLOCK:SOURce.....	162
	:TRIGger:LOGic:INPut:CLOCK:EDGE.....	163
	:TRIGger:LOGic:FUNCTion .....	163
	:TRIGger:LOGic:PATtern .....	164
	:TRIGger:LOGic:PATtern:INPut:D<x> .....	164
	:TRIGger:LOGic:PATtern:DELTatime.....	165
	:TRIGger:LOGic:PATtern:WHEn .....	165
<hr/>		
System	:SYSTem:LOCK .....	167
commands	:SYSTem:ERRor .....	167
<hr/>		
Save/Recall	:RECALL:SETUp .....	168
commands	:RECALL:WAVEform .....	168
	:SAVE:IMAGe .....	169
	:SAVE:IMAGe:FILEFormat.....	169
	:SAVE:IMAGe:INKSaver.....	170
	:SAVE:SETUp .....	170
	:SAVE:WAVEform.....	171
	:SAVE:WAVEform:FILEFormat .....	172
<hr/>		
Ethernet	:ETHERnet:DHCP .....	173
Command		
<hr/>		
Time Command	:DATE .....	173
<hr/>		
Bus Decode	:BUS1 .....	175
Commands	:BUS1:STATE .....	175
	:BUS1:TYPe.....	175
	:BUS1:INPut .....	176
	:BUS1:I2C:ADDReSS:RWINClude .....	176
	:BUS1:I2C:SCLK:SOURce .....	177



---

:BUS1:I2C:SDA:SOURce.....	177
:BUS1:PARAllel:BIT<x>:SOURce .....	177
:BUS1:PARAllel:CLOCK:EDGE .....	178
:BUS1:PARAllel:CLOCK:SOURce .....	178
:BUS1:PARAllel:WIDth .....	178
:BUS1:UART:BITRate.....	179
:BUS1:UART:DATABits .....	179
:BUS1:UART:PARItY.....	179
:BUS1:UART:PACKEt .....	180
:BUS1:UART:POLARity .....	180
:BUS1:UART:EOFPAcket .....	180
:BUS1:UART:TX:SOURce .....	181
:BUS1:UART:RX:SOURce .....	181
:BUS1:SPI:SCLK:POLARity .....	182
:BUS1:SPI:SS:POLARity.....	182
:BUS1:SPI:WORDSize .....	182
:BUS1:SPI:BITORder .....	183
:BUS1:SPI:SCLK:SOURce .....	183
:BUS1:SPI:SS:SOURce.....	183
:BUS1:SPI:MOSI:SOURce .....	184
:BUS1:SPI:MISO:SOURce .....	184
:BUS1:DISPlay:FORMAt .....	185
:LISTer:DATA .....	185
:BUS1:CAN:SOURce.....	185
:BUS1:CAN:PROBe.....	186
:BUS1:CAN:SAMPLEpoint.....	186
:BUS1:CAN:BITRate.....	186
:BUS1:LIN:BITRate .....	187
:BUS1:LIN:IDFORmat .....	187
:BUS1:LIN:POLARity.....	188
:BUS1:LIN:SAMPLEpoint .....	188
:BUS1:LIN:SOURce .....	188
:BUS1:LIN:STANDard .....	189

---

Mark Commands	:MARK .....	190
	:MARK:CREATE.....	190
	:MARK:DELEte .....	191
<hr/>		
Search	:SEARCH:COpy .....	193
Commands	:SEARCH:STATe .....	194
	:SEARCH:TOTAL .....	194
	:SEARCH:TRIGger:TYPe .....	194
	:SEARCH:TRIGger:SOURce.....	195
	:SEARCH:TRIGger:EDGE:SLOP .....	195
	:SEARCH:TRIGger:LEVel .....	196
	:SEARCH:TRIGger:HLEVel .....	196
	:SEARCH:TRIGger:LLEVel .....	197
	:SEARCH:TRIGger:PULSEWidth:POLarity .....	197
	:SEARCH:TRIGger:RUNT:POLarity .....	197
	:SEARCH:TRIGger:RISEFall:SLOP .....	198
	:SEARCH:TRIGger:PULSe:WHEn.....	198
	:SEARCH:TRIGger:PULSe:TIME.....	199
	:SEARCH:TRIGger:RUNT:WHEn .....	199
	:SEARCH:TRIGger:RUNT:TIME .....	200
	:SEARCH:TRIGger:RISEFall:WHEn .....	200
	:SEARCH:TRIGger:RISEFall:TIME .....	201
	:SEARCH:TRIGger:BUS:TYPe.....	201
	:SEARCH:TRIGger:BUS:B1:I2C:CONDition .....	202
	:SEARCH:TRIGger:BUS:B1:I2C:ADDRes:MODE ...	203
	:SEARCH:TRIGger:BUS:B1:I2C:ADDRes:TYPe .....	203
	:SEARCH:TRIGger:BUS:B1:I2C:ADDRes:VALue....	204
	:SEARCH:TRIGger:BUS:B1:I2C:ADDRes	
	:DIRection .....	205
	:SEARCH:TRIGger:BUS:B1:I2C:DATA:SIZE .....	205
	:SEARCH:TRIGger:BUS:B1:I2C:DATA:VALue .....	206
	:SEARCH:TRIGger:BUS:B1:UART:CONDition.....	206
	:SEARCH:TRIGger:BUS:B1:UART:RX:DATA:SIZE....	207
	:SEARCH:TRIGger:BUS:B1:UART:RX:DATA:VALue .....	208
	:SEARCH:TRIGger:BUS:B1:UART:TX:DATA:SIZE....	209

	:SEARCH:TRIGger:BUS:B1:UART:TX:DATa:VALue	209
	:SEARCH:TRIGger:BUS:B1:SPI:CONDition	210
	:SEARCH:TRIGger:BUS:B1:SPI:DATa:SIze	210
	:SEARCH:TRIGger:BUS:B1:SPI:DATa:MISO	
	:VALue	211
	:SEARCH:TRIGger:BUS:B1:SPI:DATa:MOSI	
	:VALue	212
	:SEARCH:TRIGger:BUS:B1:CAN:CONDition	212
	:SEARCH:TRIGger:BUS:B1:CAN:FRAMeType	213
	:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODE	214
	:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:VALue	214
	:SEARCH:TRIGger:BUS:B1:CAN:IDentifier	
	:DIRection	215
	:SEARCH:TRIGger:BUS:B1:CAN:DATa:QUALifier	215
	:SEARCH:TRIGger:BUS:B1:CAN:DATa:SIze	216
	:SEARCH:TRIGger:BUS:B1:CAN:DATa:VALue	217
	:SEARCH:TRIGger:BUS:B1:LIN:CONDition	218
	:SEARCH:TRIGger:BUS:B1:LIN:DATa:QUALifier	218
	:SEARCH:TRIGger:BUS:B1:LIN:DATa:SIze	219
	:SEARCH:TRIGger:BUS:B1:LIN:DATa:VALue	220
	:SEARCH:TRIGger:BUS:B1:LIN:ERRTYpe	220
	:SEARCH:TRIGger:BUS:B1:LIN:IDentifier:VALue	221
	:SEARCH:FFTPeak:METhod	222
	:SEARCH:FFTPeak:METhod:MPeak	222
	:SEARCH:FFTPeak:SI NFo	223
	:SEARCH:FFTPeak:LISt	223
Label Commands	:CHANnel<X>:LABel	225
	:CHANnel<X>:LABel:DISPlay	226
	:REF<X>:LABel	226
	:REF<X>:LABel:DISPlay	227
	:BUS1:LABel	228
	:BUS1:LABel:DISPlay	228
	:D<x>:LABel	229
	:D<x>:LABel:DISPlay	230

	:DIGital:ANALog:A<x>:LABel .....	230
	:DIGital:ANALog:A<x>:LABel:DISPlay .....	231
	:SET<X>:LABel .....	232
<hr/>		
Segment	:SEGMENTS:STATE .....	233
Commands	:SEGMENTS:CURRent.....	234
	:SEGMENTS:TOTALnum .....	234
	:SEGMENTS:TIME .....	235
	:SEGMENTS:DISPALL .....	235
	:SEGMENTS:MEASure:MODE .....	235
	:SEGMENTS:MEASure:PLOT:SOURce .....	236
	:SEGMENTS:MEASure:PLOT:DIVide.....	236
	:SEGMENTS:MEASure:PLOT:SElect .....	237
	:SEGMENTS:MEASure:PLOT:RESults .....	237
	:SEGMENTS:MEASure:TABLE:SOURce.....	238
	:SEGMENTS:MEASure:TABLE:SElect.....	238
	:SEGMENTS:MEASure:TABLE:LIST .....	239
	:SEGMENTS:MEASure:TABLE:SAVE .....	239
	:SEGMENTS:SAVE .....	240
	:SEGMENTS:SAVE:SOURce .....	240
	:SEGMENTS:SAVE:SElect:START .....	240
	:SEGMENTS:SAVE:SElect:END .....	241
<hr/>		
DVM Commands	:DVM:STATE .....	242
	:DVM:SOURce .....	242
	:DVM:MODE.....	243
	:DVM:VALue .....	243
<hr/>		
Go-NoGo	:GONogo:CLEar.....	244
Commands	:GONogo:EXECute .....	244
	:GONogo:FUNCTION.....	245
	:GONogo:NGCount .....	245
	:GONogo:NGDefine.....	245
	:GONogo:SOURce .....	246
	:GONogo:VIOLation .....	246

---

	:GONogo:SCRipt .....	246
	:TEMPlate:MODE .....	247
	:TEMPlate:MAXimum .....	247
	:TEMPlate:MINimum .....	247
	:TEMPlate:POSition:MAXimum .....	248
	:TEMPlate:POSition:MINimum .....	248
	:TEMPlate:SAVe:MAXimum .....	248
	:TEMPlate:SAVe:MINimum .....	249
	:TEMPlate:TOLerance.....	249
	:TEMPlate:SAVe:AUTO.....	249

---

Mask Commands	:MASK:STATe.....	250
	:MASK:RATio .....	250
	:MASK:SOURce .....	251
	:MASK:VIOLation .....	251
	:MASK:VIOLation:SAVe.....	251
	:MASK:AUTO .....	252
	:MASK:AUTO:SOURce .....	252
	:MASK:AUTO:UNITs .....	252
	:MASK:USEr:UNITs .....	253
	:MASK:USEr:AREa<x1>:POINt<x2>.....	253
	:MASK:USEr:AREa<x1>:POINt<x2>:STATe .....	254
	:MASK:USEr:CREATE .....	254
	:MASK:USEr:SAVe .....	255
	:MASK:USEr:LOAD .....	255

---

AWG Commands	:AWG:UTIL.....	257
	:AWG:UTIL:AMPCpl .....	257
	:AWG:UTIL:FREQCpl.....	258
	:AWG:UTIL:FREQCpl:OFFSet .....	258
	:AWG:UTIL:FREQCpl:RATio .....	259
	:AWG:UTIL:TRACking.....	259
	:AWG<x>:AMPlitude.....	259
	:AWG<x>:FREQuency .....	260
	:AWG<x>:FUNCTion .....	260

---

:AWG<x>:OFFSet .....	261
:AWG<x>:OUTPut:LOAd:IMPEDance .....	261
:AWG<x>:OUTPut:STATE .....	261
:AWG<x>:PHAsE.....	262
:AWG<x>:PULSe:DUTYcycle .....	262
:AWG<x>:RAMP:SYMmetry .....	262
:AWG<x>:MODulation:STATE .....	263
:AWG<x>:MODulation:TYPe.....	263
:AWG<x>:MODulation:AM:DEPth.....	263
:AWG<x>:MODulation:AM:FREQ.....	264
:AWG<x>:MODulation:AM:SHApe.....	264
:AWG<x>:MODulation:AM:PHAsE .....	264
:AWG<x>:MODulation:AM:DUTYcycle .....	265
:AWG<x>:MODulation:AM:SYMmetry .....	265
:AWG<x>:MODulation:AM:RATE .....	265
:AWG<x>:MODulation:FM:DEV .....	266
:AWG<x>:MODulation:FM:FREQ .....	266
:AWG<x>:MODulation:FM:SHApe .....	267
:AWG<x>:MODulation:FM:PHAsE .....	267
:AWG<x>:MODulation:FM:DUTYcycle.....	268
:AWG<x>:MODulation:FM:SYMmetry .....	268
:AWG<x>:MODulation:FM:RATE.....	268
:AWG<x>:MODulation:FSK:FREQ .....	269
:AWG<x>:MODulation:FSK:RATE.....	269
:AWG<x>:SWEep:STATE .....	270
:AWG<x>:SWEep:TYPe.....	270
:AWG<x>:SWEep:START .....	270
:AWG<x>:SWEep:STOP.....	271
:AWG<x>:SWEep:TIME.....	271
:AWG<x>:SWEep:SPAN .....	271
:AWG<x>:SWEep:CENTer .....	272
:AWG<x>:ARBitrary:EDIT:NUMPoint .....	272
:AWG<x>:ARBitrary:EDIT:FUNcTION .....	272
:AWG<x>:ARBitrary:SAVe:WAVEform.....	273
:AWG<x>:ARBitrary:LOAd:WAVEform .....	273

---

	:AWG<x>:ARBitrary:EDIT:COpy.....	274
	:AWG<x>:ARBitrary:EDIT:CLear.....	274
	:AWG<x>:ARBitrary:EDIT:LINE .....	275
	:AWG<x>:ARBitrary:EDIT:SCALE.....	275
	:AWG<x>:ARBitrary:EDIT:POINT.....	275
	:AWG<x>:ARBitrary:EDIT:POINT:ADD.....	276
	:AWG<x>:ARBitrary:EDIT:POINT:DELEte.....	276
<hr/>		
Data Logging	:DATALOG:STATE .....	277
Commands	:DATALOG:SOURce .....	277
	:DATALOG:SAVe .....	278
	:DATALOG:INTerval .....	278
	:DATALOG:DURation .....	279
<hr/>		
Remote Disk	:REMOTEDisk:IPADdress .....	280
Commands	:REMOTEDisk:PATHName .....	280
	:REMOTEDisk:USERName .....	280
	:REMOTEDisk:PASSWord .....	281
	:REMOTEDisk:MOUNT.....	281
	:REMOTEDisk:AUTOMount.....	282
<hr/>		
DMM	:DMM .....	283
Commands	:DMM:STATE .....	283
	:DMM:VALue .....	284
	:DMM:HOLD .....	284
	:DMM:MMIN.....	284
	:DMM:MODE.....	285
	:DMM:MODE:RANGe .....	285
	:DMM:TEMPerature:UNITs .....	286
	:DMM:TEMPerature:TYPE.....	286
	:DMM:TEMPerature:SIM .....	287
<hr/>		
Spectrum	:SA:STATE .....	288
Analyzer	:SA:LIST.....	289
Commands	:SA:MEMory.....	289

:SA:MEMory:SOURce .....	291
:SA:SOURce .....	292
:SA:SPECTRUMTrace .....	292
:SElect:NORMal .....	292
:SElect:MAXHold .....	293
:SElect:MINHold .....	293
:SElect:AVErage .....	294
:SA:AVErage:NUMAVg .....	294
:SA:DETECTionmethod:MODE .....	294
:SA:DETECTionmethod:MAXHold .....	295
:SA:DETECTionmethod:MINHold .....	295
:SA:DETECTionmethod:NORMAL .....	296
:SA:DETECTionmethod:AVErage .....	296
:SA:FREQuency .....	297
:SA:SPAN .....	297
:SA:START .....	298
:SA:STOP .....	298
:SA:RBW:MODE .....	298
:SA:RBW .....	299
:SA:SPANRbwratio .....	299
:SA:WINDow .....	300
:SA:UNIts .....	300
:SA:SCALE .....	301
:SA:POSition .....	301

Power Supply	:POWERSupply:OUTPut<X> .....	302
Commands	:POWERSupply:CONFigure .....	302
	:POWERSupply:OUTPut<X>:VOLTage .....	303
	:POWERSupply:OUTPut<X>:RECONFigure .....	303
	:POWERSupply:OUTPut<X>:OCP .....	303

USB Delay	:USBDelay .....	304
Command		

Digital	:D<x>:DISPlay .....	305
Commands	:D<x>:POSition .....	305
	:DISPlay:DIGital:HEIght .....	306



:DIGital:GROUp<x>:THReshold .....	306
:DIGital:ANALog:A<x>:DISPlay .....	307
:DIGital:ANALog:A<x>:RATio .....	307
:DIGital:ANALog:A<x>:POSition .....	308
:D<x>:MEMory .....	308
:DIGital:MEMory .....	310

---

# 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 page17.

---

Common Commands .....	36
Acquisition Commands .....	42
Autoscale Commands .....	50
Vertical Commands .....	51
Math Commands .....	57
Cursor Commands .....	66
Display Commands .....	77
Hardcopy Commands.....	81
Measure Commands .....	84
Measurement Commands .....	110
Reference Commands .....	117
Run Command .....	120
Stop Command .....	120
Single Command.....	120
Force Command.....	121
Timebase Commands.....	122
Trigger Commands.....	125
System Commands .....	167
Save/Recall Commands.....	168
Ethernet Command .....	173
Time Command .....	173
Bus Decode Commands .....	174
Mark Commands.....	190
Search Commands .....	192

Label Commands .....	225
Segment Commands .....	233
DVM Commands .....	242
Go_NoGo Commands .....	244
Mask Commands .....	250
AWG Commands .....	256
Data Logging Commands .....	277
Remote Disk Commands .....	280
DMM Commands .....	283
Spectrum Analyzer Commands .....	288
Power Supply Commands .....	302
USB Delay Command .....	304
Digital Commands .....	305

## Common Commands

*IDN?	36
*LRN?	36
*SAV	37
*RCL	37
*RST	38
*CLS	38
*ESE	38
*ESR	39
*OPC	39
*SRE	40
*STB	41

### \*IDN?

→ Query

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

### \*LRN?

→ Query

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

**\*SAV**



**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**



**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 MDO-2000E (recalls the default panel settings).

Syntax \*RST

**\*CLS** (Set) →

Description Clears the error queue.

Syntax \*CLS

**\*ESE** (Set) →  
→ (Query)

Description Sets or queries the Standard Event Status Enable register.

Syntax \*ESE <NR1>

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**

→ **Query**

**Description**      Queries the Standard Event Status (Event) register. The Event Status register is cleared after it is read.

**Query Syntax**      \*ESR?

**Return parameter** <NR1> 0~255

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**

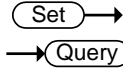
→ **Query**

**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.



**\*SRE**

**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

→ Query

Description	Queries the bit sum of the Status Byte register with MSS (Master summary Status) replacing the RQS bit (bit 6).
-------------	---

Query Syntax	*STB?
--------------	-------

Return parameter	<NR1> 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?
---------	-------

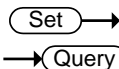
>16

Indicates that the MAV bit is set.

## Acquisition Commands

:ACQuire:AVERage .....	42
:ACQuire:MODE .....	43
:ACQuire<X>:MEMory? .....	43
:ACQuire:FILTer:SOURce .....	45
:ACQuire:FILTer .....	46
:ACQuire:FILTer:FREQuency.....	46
:ACQuire:FILTer:FREQuency:UPPER .....	46
:ACQuire:FILTer:FREQuency:LOWER .....	47
:ACQuire:FILTer:TYPe .....	47
:ACQuire:FILTer:TRACking.....	48
:ACQuire<X>:STATe? .....	48
:ACQuire:INTERpolation .....	48
:ACQuire:RECOrdlength.....	49
:HEADER .....	49

### :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	:ACQuire:MODE AVERage :ACQuire:AVERage 2  Selects the average acquisition mode, and sets the average number to 2.

Set →  
→ Query

---

**: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	:ACquire:MODE PDETECT Sets the sampling mode to peak detection.						

→ Query

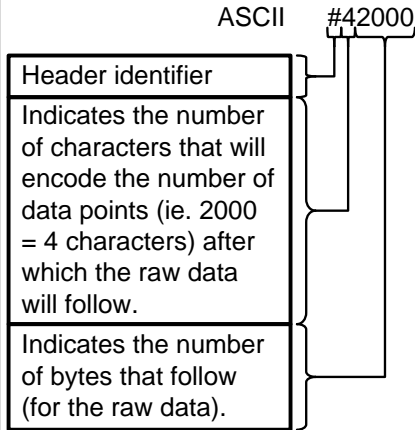
---

**:ACquire<X>:MEMory?**

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	<table border="0"> <tr> <td>&lt;string&gt;</td> <td>Returns acquisition settings followed by raw waveform block data.</td> </tr> <tr> <td>&lt;waveform block data&gt;</td> <td>               &lt;string&gt;                Returns the acquisition settings for the selected channel.                 Format:                parameter(1),setting(1);parameter(2),setting(2)...parameter(n),setting(n);Waveform Data;                 &lt;waveform block data&gt;                Header followed by the raw waveform data.             </td> </tr> </table>	<string>	Returns acquisition settings followed by raw waveform block data.	<waveform block data>	<string> Returns the acquisition settings for the selected channel.  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.
<string>	Returns acquisition settings followed by raw waveform block data.				
<waveform block data>	<string> Returns the acquisition settings for the selected channel.  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.....

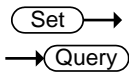
The actual value of a data point can be calculated with the following formula:

(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 :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**

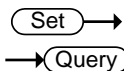
Description	Sets or queries the filter frequency.	
Syntax	:ACQUIRE:FILTER:FREQUENCY {DEFAULT <NRf> ?}	
Parameter/ Return parameter	DEFAULT	Sets the filter frequency to the default.
	<NRf>	Manually sets the filter frequency. (1Hz ~ 500MHz)
Example	:ACQUIRE:FILTER:FREQUENCY 1 Sets the filter frequency to 1Hz.	


**:ACQUIRE:FILTER:FREQUENCY:UPPER**

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



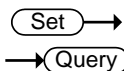
**:ACQuire:FILTer:FREQuency:LOWER**

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



**: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

**:ACQUIRE:FILTER:TRACKING**

Set →

→ Query

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.	

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

→ Query

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.

Set →

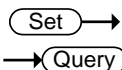
→ Query

**:ACQUIRE:INTERPOLATION**

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



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



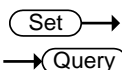
**:ACquire:RECOrdlength**

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.



**:HEADer**

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.....	50
:AUTORSET:MODE.....	50

### :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 .....	51
:CHANnel<X>:COUPling.....	52
:CHANnel<X>:DESKew .....	52
:CHANnel<X>:DISPlay .....	52
:CHANnel<X>:EXPanD .....	53
:CHANnel<X>:IMPedance? .....	53
:CHANnel<X>:INVert .....	54
:CHANnel<X>:POSition .....	54
:CHANnel<X>:PROBe:RATio .....	55
:CHANnel<X>:PROBe:TYPe .....	55
:CHANnel<X>:SCALE .....	55

:CHANnel<X>:BWLimit 
 →  
 →

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** 



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.	

**: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.	

**:CHANnel<X>:DISPlay** 

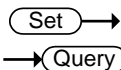


Description	Turns a channel on/off or returns its status.	
Syntax	:CHANnel<X>:DISPlay {OFF   ON   ?}	
Parameter	<X>	Channel 1,2,3,4
	OFF	Channel off
	ON	Channel on

Return Parameter	ON	Channel is on
	OFF	Channel is off

Example :CHANnel1:DISPlay ON  
Turns on Channel 1

**: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?**



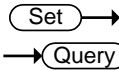
Description Returns the impedance of the oscilloscope. (The impedance of the MDO-2000E is fixed at 1MΩ)

Syntax :CHANnel<X>:IMPedance?

Parameter	<x>	Channel
	1/2/3/4	CH1/2/3/4

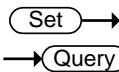
Return parameter <NR3> Returns the impedance value.

Example :CHANnel1:IMPedance?  
1.000000E+06  
The impedance is 1M ohms.




**:CHANnel<X>:INVert**

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	The vertical position will only be set to closest allowed value. The position level range depends on the vertical scale.	
	The scale must first be set before the position can be set.	
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	:CHANnel1:POSition 2.4E-3 Sets the Channel 1 position to 2.4mV/mA	
Example 2	:CHANnel1:POSition? 2.4E-3 Returns 2.4mV as the vertical position.	

Set →  
 → Query

**: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	:CHANnel1:PROBE:RATio 1.00E+0 Sets the Channel 1 probe attenuation factor to 1x	

Set →  
 → Query


**:CHANnel<X>:PROBE:TYPE**

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.	

Set →  
 → Query

**:CHANnel<X>:SCALE**

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.

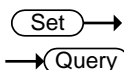
Syntax	:CHANnel<X>:SCALE { <NRf>   ?}	
Parameter	<X>	Channel 1, 2, 3, 4
	<NRf>	Vertical scale: 2e-3 to 1e+1 2mV to 10V (Probe x1)
Return parameter	<NR3>	Returns the vertical scale in volts or amps.
Example	:CHANnel1:SCALE 2.00E-2	
	Sets the Channel 1 vertical scale to 20mV/div	



## Math Commands

:MATH:DISP .....	57
:MATH:TYPE .....	58
:MATH:DUAL:SOURce<X> .....	58
:MATH:DUAL:OPERator .....	58
:MATH:DUAL:POSition .....	59
:MATH:DUAL:SCALe .....	59
:MATH:FFT:SOURce .....	60
:MATH:FFT:MAG .....	60
:MATH:FFT:WINDow .....	60
:MATH:FFT:POSition .....	61
:MATH:FFT:SCALe .....	61
:MATH:FFT:HORizontal:SCALe .....	62
:MATH:FFT:HORizontal:POSition .....	62
:MATH:DEFine .....	62
:MATHVAR? .....	63
:MATHVAR:VAR<X> .....	64
:MATH:ADVanced:POSition .....	64
:MATH:ADVanced:SCALe .....	64

### :MATH:DISP



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

Set →

→ Query

**: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	
Parameter	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.	

Set →

→ Query

**: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 →

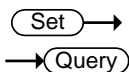
→ Query

**:MATH:DUAL:OPERator**

Description	Sets the math operator for dual math operations.	
-------------	--	--

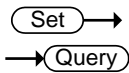
Syntax	:MATH:DUAL:OPERator {PLUS   MINUS   MUL   DIV ?}	
Parameter	PLUS	+ operator
	MINUS	- operator
	MUL	× operator
	DIV	÷ operator
Return parameter	Returns operator type.	
Example	:MATH:DUAL:OPERator PLUS Sets the math operator as plus (+).	

**: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	Channel 1 to 4
	REF1~4	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	Linear units (Vrms)
	DB	Logarithmic units (dB)
Return parameter	Returns the FFT vertical units.	
Example	:MATH:FFT:MAG DB Sets FFT vertical units to dB.	

**:MATH:FFT:WINDow** 

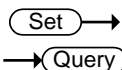
  


Description	Sets the windowing filter used for the FFT function.	
Syntax	:MATH:FFT:WINDow {RECTangular HAMming HANning BLAckman ?}	

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.



**:MATH:FFT:POSition**

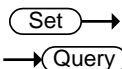
Description Sets the vertical position of the displayed FFT result.

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.



**:MATH:FFT:SCALe**

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.

Set →  
 → Query

**:MATH:FFT:HORizontal:SCALE**

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.	

Set →  
 → Query

**:MATH:FFT:HORizontal:POSition**

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.	

Set →  
 → Query

**:MATH:DEFine**

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, 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 →  
 → Query

**:MATHVAR:VAR<X>**

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 →  
 → Query

**:MATH:ADVanced:POSition**

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.

Set →  
 → Query

**: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


---

:CURSor:MODE	67
:CURSor:SOURce	67
:CURSor:HUNI	68
:CURSor:HUSE	68
:CURSor:VUNI	69
:CURSor:VUSE	69
:CURSor:DDT	69
:CURSor:H1Position	70
:CURSor:H2Position	70
:CURSor:HDELta	71
:CURSor:V1Position	71
:CURSor:V2Position	71
:CURSor:VDELta	72
:CURSor:XY:RECTangular:X:POSition<X>	72
:CURSor:XY:RECTangular:X:DELta	72
:CURSor:XY:RECTangular:Y:POSition<X>	73
:CURSor:XY:RECTangular:Y:DELta	73
:CURSor:XY:POLar:RADIUS:POSition<X>	73
:CURSor:XY:POLar:RADIUS:DELta	74
:CURSor:XY:POLar:THETA:POSition<X>	74
:CURSor:XY:POLar:THETA:DELta	74
:CURSor:XY:PRODuct:POSition<X>	75
:CURSor:XY:PRODuct:DELta	75
:CURSor:XY:RATio:POSition<X>	75
:CURSor:XY:RATio:DELta	76

---

**:CURSor:MODE** 



**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 | ? }

<b>Parameter</b>	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.

**:CURSor:SOURce** 



**Description** Sets or queries the cursor source.

**Syntax** :CURSor:SOURce {CH1 | CH2 | CH3 | CH4 | REF1 | REF2 | REF3 | REF4 | MATH | LOGic | BUS1 | ? }

<b>Parameter</b>	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 →

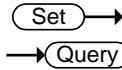
Description	Sets the current cursor position as the phase or ratio reference for the Percent or Degrees (horizontal) cursors.	
 Note	This command can only be used when :CURSor:HUNI is set to DEGrees or PERcent.	
Syntax	:CURSor:HUSE {CURRent}	
Related Commands	:CURSor:MODE :CURSor:HUNI	
Parameter	CURRent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

		Set →
		→ Query
<b>:CURSor:VUNI</b>		
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.	

		Set →
<b>:CURSor:VUSE</b>		
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.	

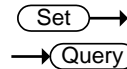
		→ Query
<b>:CURSor:DDT</b>		
Description	Returns the deltaY/deltaT (dy/dT) readout. This function is only supported if the source channels are CH1~4, Ref1~4 or Math.	

Syntax	:CURSor:DDT{?}
Related Commands	:CURSor:MODE
Return Parameter	<NR3> Returns the readout in <NR3> format.
Example	:CURSor:DDT? 4.00E-05



**: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.



**: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> (Set) →  
→ (Query)

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 → (Query)

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> → (Query)

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	<NR3> 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	<X> 1, 2 (Cursor 1, Cursor 2)
Return parameter	<NR3> 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	<NR3> 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.	

**:CURSor:XY:RATio:DELta** → Query

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.....	77
:DISPlay:INTensity:GRATICule.....	77
:DISPlay:INTensity:BACKLight .....	78
:DISPlay:INTensity:BACKLight:AUTODim:ENABle ..	78
:DISPlay:INTENSITY:BACKLight:AUTODim:TIME ...	78
:DISPlay:PERsistence .....	79
:DISPlay:GRATICule .....	79
:DISPlay:WAVEform.....	80
:DISPlay:OUTPut .....	80

: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%.

Set →  
 → Query

**:DISPlay:INTensity:BACKLight**

---

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%.	

Set →  
 → Query

**:DISPlay:INTensity:BACKLight:AUTODim:ENABLE**

---

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.	

Set →  
 → Query

**:DISPlay:INTENSITY:BACKLight:AUTODim:TIME**

---

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** 







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.
	INFInite	Infinite persistence
	OFF	No persistence


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

**:DISPlay:GRATicule** 



Description	Sets or queries graticule display type.	
Syntax	:DISPlay:GRATicule {FULL   GRID CROsS   FRAMe   ?}	
Parameter	FULL	
	FRAMe	
Parameter	CROsS	
	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 | ?}

<b>Parameter</b>	VECTor	Vectors
	DOT	Dots

**Return parameter** Returns VECTOR or DOT.

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

**: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

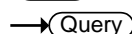
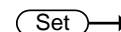
:HARDcopy:START .....	81
:HARDcopy:MODE .....	81
:HARDcopy:PRINTINKSaver .....	82
:HARDcopy:SAVEINKSaver .....	82
:HARDcopy:SAVEFORMat .....	82
:HARDcopy:ASSIGN .....	83

### :HARDcopy:START



Description	Executing the HARDcopy:START command is the equivalent of pressing the Hardcopy key on the front panel.
Syntax	:HARDcopy:START
Related Commands	:HARDcopy:MODE :HARDcopy:PRINTINKSaver :HARDcopy:SAVEINKSaver :HARDcopy:SAVEFORMat :HARDcopy:ASSIGN

### :HARDcopy:MODE



Description	Sets or queries whether hardcopy is set to print or save.				
Syntax	:HARDcopy:MODE { PRINT   SAVE   ? }				
Related Commands	:HARDcopy:START				
Parameter	<table> <tr> <td>PRINT</td> <td>Print mode</td> </tr> <tr> <td>SAVE</td> <td>Save mode</td> </tr> </table>	PRINT	Print mode	SAVE	Save mode
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 (Set) →  
→ (Query)

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).	

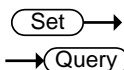
Example :HARDcopy:SAVEINKSaver ON  
Sets Inksaver to ON for saving screen images.

:HARDcopy:SAVEFORMat (Set) →  
→ (Query)

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.

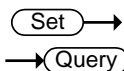
## Measure Commands

---

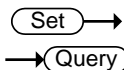
:MEASure:GATing.....	85
:MEASure:SOURce .....	85
:MEASure:METHod .....	86
:MEASUrement:REFLevel:PERCent:HIGH .....	86
:MEASUrement:REFLevel:PERCent:LOW .....	87
:MEASUrement:REFLevel:PERCent:MID .....	87
:MEASUrement:REFLevel:PERCent:MID2 .....	87
:MEASure:FALL.....	88
:MEASure:FOVShoot.....	88
:MEASure:FPreshoot .....	89
:MEASure:FREQuency.....	89
:MEASure:NWIDth .....	90
:MEASure:PDUTy .....	90
:MEASure:PERiod .....	91
:MEASure:PWIDth .....	91
:MEASure:RISe .....	92
:MEASure:ROVShoot.....	93
:MEASure:RPreshoot.....	93
:MEASure:PPULSE.....	94
:MEASure:NPULSE.....	94
:MEASure:PEDGE .....	95
:MEASure:NEDGE .....	95
:MEASure:AMPlitude.....	96
:MEASure:MEAN .....	97
:MEASure:CMEan .....	97
:MEASure:HIGH .....	98
:MEASure:LOW.....	98
:MEASure:MAX.....	99
:MEASure:MIN .....	99
:MEASure:PK2PK.....	100
:MEASure:RMS .....	100
:MEASure:CRMS.....	101

:MEASure:AREa .....	102
:MEASure:CARea .....	102
:MEASure:FRRDelay .....	103
:MEASure:FRFDelay.....	103
:MEASure:FFRDelay.....	104
:MEASure:FFFDelay.....	105
:MEASure:LRRDelay .....	105
:MEASure:LRFDelay.....	106
:MEASure:LFRDelay.....	106
:MEASure:LFFDelay.....	107
:MEASure:PHAsE.....	108
:MEASure:PFLI.....	108
:MEASure:FLI.....	109

**: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**

Set →

→ Query

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.

**:MEASUrement:REFLevel:PERCent:HIGH**

Set →

→ 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 →  
 → Query

**:MEASUrement:REFLevel:PERCent:LOW**

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%.

Set →  
 → Query

**: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%.

Set →  
 → Query

**: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**

→ Query

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**

→ Query

Description	Returns the fall overshoot amplitude.	
Syntax	:MEASure:FOVShoot{?}	
Related Commands	:MEASure:SOURce<X>	
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

→ Query

Description	Returns fall preshoot amplitude.	
Syntax	:MEASure:FPReshoot{?}	
Related Commands	:MEASure:SOURce<X>	
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** → Query

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



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

```
:MEASure:SOURce1 CH1
```

```
:MEASure:PERiod?
```

```
1.0E-3
```

Selects Channel 1, and then measures the period.

### :MEASure:PWIDth



Description

Returns the first positive pulse width.


Syntax

```
:MEASure:PWIDth{?}
```

Related  
Commands

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


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** :MEASure:SOURce1 CH1  
 :MEASure:PWIDth?  
 5.0E-6  
 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** → Query

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 Commands :MEASure:SOURce<X>

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** → **Query**

**Description** Returns the amplitude difference between the Vhigh-Vlow.

**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**


→ **Query**

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

**Syntax** :MEASure:MEAN{?}

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

<b>Return parameter</b>	<b>&lt;NR3&gt;</b>	Returns the 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: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>

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

 **Note** Before using this command, select the measurement channel. See the example below.

Example :MEASure:SOURce1 CH1  
 :MEASure:HIGH?  
 3.68E-3  
 Selects Channel 1, and then measures the high voltage/current.

**:MEASure:LOW** → Query

Description Returns the global low voltage/current.

Syntax :MEASure:LOW{?}

Related Commands :MEASure:SOURce<X>

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

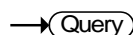
```
:MEASure:SOURce1 CH1
```

```
:MEASure:LOW?
```

```
1.00E-0
```

Selects Channel 1, and then measures the low current/voltage.

**:MEASure:MAX**



Description

Returns the maximum amplitude.

Syntax

```
:MEASure:MAX{?}
```

Related

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

Commands

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 Commands	:MEASure:SOURce<X>
------------------	--------------------

Return parameter	<NR3>	Returns the voltage or current peak to peak measurement.
	Chan Off	Indicates the source channel is not activated.



Note

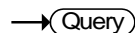
Before using this command, select the measurement channel. See the example below.

Example

```
:MEASure:SOURce1 CH1
:MEASure:PK2Pk?
2.04E-1
```


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

**:MEASure:RMS**



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

Syntax	:MEASure:RMS{?}	
Related Commands	:MEASure:SOURce<X>	
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 → **

Description	Returns the root-mean-square voltage/current of one full periods.	
Syntax	:MEASure:CRMS{?}	
Related Commands	:MEASure:SOURce<X>	
Return parameter	<NR3>	Returns the CRMS value.
	Chan Off	Indicates the source channel is not activated.

 **Note** Before using this command, select the measurement channel. See the example below.

**Example**

```
:MEASure: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>	
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: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** → Query

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

Syntax :MEASure:FRFDelay{?}

Related Commands :MEASure:SOURce<X>

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:FRFDelay?
3.43E-6
```

Select channel 1 and 2 as source1/2, and then measures FRF.

**:MEASure:FFRDelay**



Description

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

Syntax

```
:MEASure:FRRDelay{?}
```

Related

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

Commands

Return parameter

<NR3>	Returns the delay.
-------	--------------------

Chan Off	Indicates the source channel is not activated.
----------	--



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** → **Query**

**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>

<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:FFFDelay?
-8.89E-6
```

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


**:MEASure:LRRDelay** → **Query**

**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>

<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:LRRDelay?  
 -8.89E-6  
 Select channel 1 and 2 as delay source1/2, and then measure LRR.

**:MEASure:LRFDelay** → Query

Description	Returns the delay between the first rising edge of source1 and the last rising edge of source2.	
Syntax	:MEASure:LRFDelay{?}	
Related Commands	:MEASure:SOURce<X>	
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:LRFDelay?  
 -4.99E-6  
 Select channel 1 and 2 as delay source1/2, and then measure LRF.

**:MEASure:LFRDelay** → Query

Description	Returns the delay between the first falling edge of source1 and the last rising edge of source2.	
Syntax	:MEASure:LFRDelay{?}	

Related Commands :MEASure:SOURce<X>

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:LFRDelay?  
 -9.99E-6  
 Select channel 1 and 2 as delay source1/2, and then measure LFR.


**:MEASure:LFFDelay** → 

**Description** Returns the delay between the first falling edge of source1 and the last falling edge of source2.

**Syntax** :MEASure:LFFDelay{?}

Related Commands :MEASure:SOURce<X>

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**

→ Query

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> ..... 110  
 :MEASUrement:MEAS<X>:TYPE ..... 111  
 :MEASUrement:MEAS<X>:STATE ..... 111  
 :MEASUrement:MEAS<X>:VALue ..... 112  
 :MEASUrement:MEAS<X>:MAXimum ..... 113  
 :MEASUrement:MEAS<X>:MEAN ..... 113  
 :MEASUrement:MEAS<X>:MINImum ..... 114  
 :MEASUrement:MEAS<X>:STDdev ..... 115  
 :MEASUrement:STATIstics:MODE ..... 115  
 :MEASUrement:STATIstics:WEIghting ..... 116  
 :MEASUrement:STATIstics ..... 116

:MEASUrement:MEAS<X>:SOURCE<X> 


**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 | MATH | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | ? }

**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

	D0~D15	Digital source D0~D15
Return parameter	CH1 to CH4	Channel 1, 2, 3, 4
	MATH	Math source

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

Set →  
 → Query

**:MEASUREMENT:MEAS<X>:TYPE**

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

**Syntax** :MEASUREMENT:MEAS<X>:TYPE  
{PK2pk | MAXimum | MINimum | AMplitude | HIGH | LOW | MEAN | CMean | RMS | CRMs | AREa | CAREa | ROVShoot | FOVShoot | RPReshoot | FPReshoot | FREQuency | PERIod | RISE | FALL | PWIdth | NWIdth | PDUTy | PPULSE | NPULSE | PEDGE | NEDGE | PFLicker | FLicker | FRRDelay | FRFDelay | FFRDelay | FFFDelay | LRRDelay | LRFDelay | LFRDelay | LFFDelay | PHAse | ?}

**Related commands** :MEASUREMENT:MEAS<X>:SOURCE<X>

<b>Parameter</b>	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 →  
 → Query

**:MEASUREMENT:MEAS<X>:STATE**

**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&lt;X&gt;: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&lt;X&gt;:MEAN

→ Query

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	<pre>:MEASUrement:MEAS3:SOUrce1 CH1 :MEASUrement:MEAS3:TYPe PK2PK :MEASUrement:MEAS3:STATE ON :MEASUrement:STATIstics:MODE ON :MEASUrement:MEAS3:MEAN? 2.090E-02</pre> <p>Returns the mean measurement result for measurement number 3.</p>	

**:MEASUrement:MEAS<X>:MINImum** → 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>:MINImum?	
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:MINImum?</pre>	

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

<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:STDdev?
1.530E-03
```

Returns the standard deviation for measurement number 3.

→ **Set** →

**:MEASUrement:STATIstics:MODE** → **Query**

**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 (Set) →  
 → (Query)

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.

: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.....	117
:REF<X>:TIMebase:POSition .....	117
:REF<X>:TIMebase:SCALe .....	118
:REF<X>:OFFSet .....	118
:REF<x>:SCALe .....	119

:REF<X>:DISPlay 
 Set →  
 →  Query

**Description** Sets or queries whether a reference waveform will be shown on the display. A reference waveform must first be saved before this command can be used.

**Syntax** :REF<x>:DISPlay { OFF| ON| ? }

<b>Parameter</b>	<X>	Reference waveform 1, 2, 3, 4.
	OFF	Turns the selected reference waveform off
	ON	Turns the selected reference waveform on

**Return parameter** Returns the status of the selected reference waveform. (OFF, ON).

**Example** :REF1:DISPlay ON  
Turns on reference1 (REF 1) on the display.

:REF<X>:TIMebase:POSition 
 Set →  
 →  Query

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

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

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

Parameter	<X>	Reference waveform 1, 2, 3, 4.
	<NRf>	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.

Set →

:REF<X>:TIMebase:SCALE

→ Query

Description Sets or returns the selected reference waveform time base scale.

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

Related commands :REF<X>:DISPlay

Parameter	<X>	Reference waveform 1, 2, 3, 4.
	<NRf>	Horizontal scale

Return parameter <NR3> Returns the reference waveform scale.

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

Set →

:REF<X>:OFFSet

→ Query

Description Sets or returns the selected reference waveform vertical position (offset).

Syntax :REF<X>:OFFSet { <NRf> | ?}

Related commands :REF<X>:DISPlay

Parameter	<X>	Reference waveform 1, 2, 3, 4.
	<NRf>	Vertical offset

Return parameter <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.

:REF<x>:SCALE 


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).
-------------	---

---

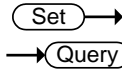
Syntax	:FORCe
--------	--------

---

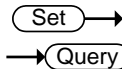
## Timebase Commands

:TIMebase:EXPand .....	122
:TIMebase:POSition .....	122
:TIMebase:SCALe .....	122
:TIMebase:MODe .....	123
:TIMebase:WINDow:POSition .....	123
:TIMebase:WINDow:SCALe .....	124

### :TIMebase:EXPand

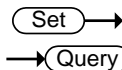


Description	Sets or queries the horizontal expansion mode.	
Syntax	:TIMebase:EXPand {CENTer TRIGger ?}	
Parameter/Return parameter	CENTer	Expand from the center of the display.
	TRIGger	Expand from the trigger point.
Example	:TIMebase:EXPand TRIGger Sets the expansion point to the trigger point.	



### :TIMebase:POSition

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

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 →

### :TIMebase:MODE

→ Query

Description	Sets or queries the time base mode. The time base mode determines the display view window on the scope.	
Syntax	:TIMebase:MODE {MAIN   WINDow   XY   ?}	
Parameter	MAIN	Sets the time base mode to the main screen.
	WINDow	Sets the time base mode to the zoom window.
	XY	Sets the time base mode to the XY display.
Return parameter	Returns the time base mode (MAIN, WINDOW, XY)	
Example	:TIMebase:MODE MAIN Sets the time base mode to the main mode.	

Set →

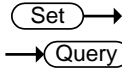
### :TIMebase:WINDow:POSition

→ Query

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.



**: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.....	127
:TRIGger:TYPe.....	127
:TRIGger:SOURce .....	128
:TRIGger:COUPle.....	128
:TRIGger:NREJ .....	129
:TRIGger:MODE.....	129
:TRIGger:HOLDoff.....	129
:TRIGger:LEVel.....	130
:TRIGger:HLEVel.....	130
:TRIGger:LLEVel.....	131
:TRIGger:EDGE:SLOP .....	131
:TRIGger:DELay:SLOP .....	132
:TRIGger:DELay:TYPe .....	132
:TRIGger:DELay:TIME.....	132
:TRIGger:DELay:EVENT.....	133
:TRIGger:DELay:LEVel .....	133
:TRIGger:PULSEWidth:POLarity.....	133
:TRIGger:RUNT:POLarity .....	134
:TRIGger:RUNT:WHEn .....	134
:TRIGger:RUNT:TIME .....	135
:TRIGger:RISEFall:SLOP .....	135
:TRIGger:RISEFall:WHEn.....	136
:TRIGger:RISEFall:TIME.....	136
:TRIGger:VIDeo:TYPe .....	137
:TRIGger:VIDeo:FIELD .....	137
:TRIGger:VIDeo:LINE.....	138
:TRIGger:VIDeo:POLarity .....	138
:TRIGger:PULSe:WHEn .....	138
:TRIGger:PULSe:TIME .....	139
:TRIGger:TIMEOut:WHEn .....	139
:TRIGger:TIMEOut:TIMER .....	140
:TRIGger:ALTErnate .....	140

:TRIGger:STATe ..... 141

:TRIGger:EXTERnal:PROBe:TYPe..... 141

:TRIGger:EXTERnal:PROBe:RATio ..... 142

:TRIGger:BUS:TYPe ..... 142

:TRIGger:BUS:THReshold:CH<x>..... 143

:TRIGger:BUS:B1:I2C:CONDition ..... 143

:TRIGger:BUS:B1:I2C:ADDRes:MODE..... 144

:TRIGger:BUS:B1:I2C:ADDRes:TYPe..... 144

:TRIGger:BUS:B1:I2C:ADDRes:VALue ..... 145

:TRIGger:BUS:B1:I2C:ADDRes:DIRection ..... 146

:TRIGger:BUS:B1:I2C:DATa:SIZE ..... 146

:TRIGger:BUS:B1:I2C:DATa:VALue..... 147

:TRIGger:BUS:B1:UART:CONDition ..... 147

:TRIGger:BUS:B1:UART:RX:DATa:SIZE ..... 148

:TRIGger:BUS:B1:UART:RX:DATa:VALue ..... 149

:TRIGger:BUS:B1:UART:TX:DATa:SIZE ..... 149

:TRIGger:BUS:B1:UART:TX:DATa:VALue ..... 150

:TRIGger:BUS:B1:SPI:CONDition ..... 151

:TRIGger:BUS:B1:SPI:DATa:SIZE ..... 151

:TRIGger:BUS:B1:SPI:DATa:MISO:VALue ..... 152

:TRIGger:BUS:B1:SPI:DATa:MOSI:VALue ..... 152

:TRIGger:BUS:B1:CAN:CONDition..... 153

:TRIGger:BUS:B1:CAN:FRAMeType ..... 154

:TRIGger:BUS:B1:CAN:IDentifier:MODE..... 154

:TRIGger:BUS:B1:CAN:IDentifier:VALue ..... 155

:TRIGger:BUS:B1:CAN:IDentifier:DIRection ..... 155

:TRIGger:BUS:B1:CAN:DATa:QUALifier ..... 156

:TRIGger:BUS:B1:CAN:DATa:SIZE..... 157

:TRIGger:BUS:B1:CAN:DATa:VALue..... 157

:TRIGger:BUS:B1:LIN:CONDition ..... 158

:TRIGger:BUS:B1:LIN:DATa:QUALifier ..... 159

:TRIGger:BUS:B1:LIN:DATa:SIZE ..... 159

:TRIGger:BUS:B1:LIN:DATa:VALue ..... 160

:TRIGger:BUS:B1:LIN:ERRTYPe..... 161

:TRIGger:BUS:B1:LIN:IDentifier:VALue ..... 161

:TRIGger:BUS:B1:PARAllel:VALue ..... 162  
 :TRIGger:LOGic:INPut:CLOCK:SOURce..... 162  
 :TRIGger:LOGic:INPut:CLOCK:EDGE ..... 163  
 :TRIGger:LOGic:FUNCTion..... 163  
 :TRIGger:LOGic:PATtern ..... 164  
 :TRIGger:LOGic:PATtern:INPut:D<x>..... 164  
 :TRIGger:LOGic:PATtern:DELtAtime..... 165  
 :TRIGger:LOGic:PATtern:WHEn ..... 165

**: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   LOGic   BUS   TIMEOut   ? }
Parameter	EDGE Edge trigger DELay Delay trigger PULSEWidth Pulse width trigger VIDeo Video trigger RUNT Runt trigger RISEFall Rise and fall trigger LOGic Logic 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 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | ? }

Parameter	CH1 to CH4	Channel 1 to channel 4
	EXT	External source
	LINe	AC Line
	D0~D15	Digital channels D0~D15

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 | ? }

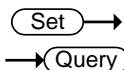
Parameter	AC	AC mode
	DC	DC mode
	HF	High frequency rejection
	LF	Low frequency rejection



Return parameter Returns the trigger coupling.

Example :TRIGger:COUPle AC  
Sets the trigger coupling to AC.

**:TRIGger:NREJ**



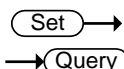
Description Sets or queries noise rejection status.

Syntax :TRIGger:NREJ {OFF| ON| ?}

Parameter	OFF	Turns noise rejection off
	ON	Turns noise rejection on

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

Example :TRIGger:NREJ ON  
Turns noise rejection on.



**:TRIGger:MODE**

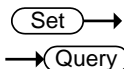
Description Sets or queries the trigger mode.

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.



**:TRIGger:HOLDoff**

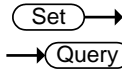
Description Sets or queries the holdoff time.

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.



**:TRIGger:LEVel**

Description Sets or queries the level.



Note

Not applicable to Pulse Runt and Rise & Fall triggers.

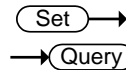
Syntax :TRIGger:LEVel {TTL | ECL | SETTO50 | <NRf> | ?}

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.



**: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.	

: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	

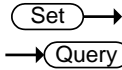
Parameter	<NRf>	Low level value.
Return parameter	<NR3>	Returns the trigger low level.
Example	:TRIGger:LLEVel -3.30E-3 Sets the trigger low level to -330mV/mA.	

:TRIGger:EDGE:SLOP 
 →  
 →

Description	Sets or queries the trigger slope.	
Syntax	:TRIGger:EDGE:SLOP {RISe   FALL   EITHer   ? }	

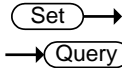
Related commands	:TRIGger:TYPe	
Parameter	RISe	Rising slope
	FALL	Falling slope
	EITHer	Either rising or falling slope

Return parameter	Returns the trigger slope.	
Example	:TRIGger:EDGE:SLOP FALL Sets the trigger slope to falling.	



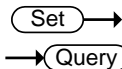
**:TRIGger:DElay:SLOP**

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.

Set →

→ Query

### :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

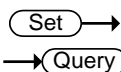
**:TRIGger:RUNT:WHEn**

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 (≠).



### :TRIGger:RUNT:TIME

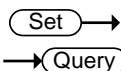
Description Sets or queries the Pulse Runt trigger time.

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.



### :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	<table border="1"> <tr> <td>MOREthan</td> <td>&gt;</td> </tr> <tr> <td>LESSthan</td> <td>&lt;</td> </tr> <tr> <td>Equal</td> <td>=</td> </tr> <tr> <td>UNEQual</td> <td>≠</td> </tr> </table>	MOREthan	>	LESSthan	<	Equal	=	UNEQual	≠
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 (≠).								

Set →  
→ Query

**:TRIGger:RISEFall:TIME**

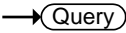
Description	Sets or queries the Rise and Fall time.
Syntax	:TRIGger:RISEFall:TIME {<NRf>   ? }
Related commands	:TRIGger:TYPe :TRIGger:RISEFall:WHEn
Parameter	<NRf> Rise and Fall time (4nS to 10S)
Return Parameter	<NR3> Returns the rise and fall time in seconds.
Example	:TRIGger:RISEFall:TIME 4.00E-5 Sets the trigger rise & fall to 40.0us.



:TRIGger:VIDeo:TYPe 

  


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.

Set →  
→ Query

**: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.

Set →  
→ Query

**:TRIGger:VIDeo:POLarity**

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.

Set →  
→ Query

**:TRIGger:PULSe:WHEN**

Description Sets or queries the pulse width trigger conditions.

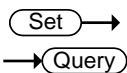
Syntax :TRIGger:PULSe:WHEn { MOREthan | LESSthan | EQual | UNEQual | ? }

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.

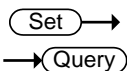
Syntax :TRIGger:PULSe:TIME {<NRf> | ?}

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.

Set →

→ Query

### :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

Set →

→ Query

### :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.

**:TRIGger:STATe** → Query

Description Returns the current state of the triggering system.

Syntax :TRIGger:STATe?

Return parameter	*ARMED	Indicates that the oscilloscope is acquiring pretrigger information.
	*AUTO	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.

Set →

**:TRIGger:EXTERnAl:PROBE:TYPE** → Query

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

Set →  
← Query

: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


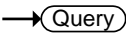
:TRIGger:BUS:TYPE → Query

Description Returns the current bus type.

Syntax :TRIGger:BUS:TYPE?

Return parameter	12C	I <sup>2</sup> C mode
	SPI	SPI mode
	UART	UART mode
	CAN	CAN mode
	LIN	LIN mode

Example :TRIGger:BUS:TYPE?  
UART

**:TRIGger:BUS:THReshold:CH<x>**

**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**

**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 | ? }

<b>Parameter</b>	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 (Set) →  
→ (Query)

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 currenty set to 7 bits.

:TRIGger:BUS:B1:I2C:ADDRess:TYPE (Set) →  
→ (Query)

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 



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** 



**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

<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: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** 



**Description** Sets or queries the UART triggering condition.

**Syntax** :TRIGger:BUS:B1:UART:CONDition { RXSTArt | RXDATA | RXENDPacket | TXSTArt | TXDATA | TXENDPacket | TXPARItYerr | RXPARTYerr | ? }

<b>Parameter</b>	RXSTArt	Set trigger on the RX Start Bit.
	RXDATA	Set trigger on RX Data.

RXENDPacket	Set trigger on the RX End of Packet condition.
RXPARltyerr	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.
TXPARltyerr	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 (Set) →  
→ (Query)

**Description** Sets or queries the number of bytes for UART data.

**Note** This setting only applies when the UART trigger is set to trigger on 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>	<b>&lt;sting&gt;</b>	The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string".  x = don't care  1 = binary 1  0 = binary 0
------------------	----------------------	--

**Return Parameter** Returns the data value.


**Example1** :TRIGger:BUS:B1:UART:CONDition RXDATA  
 :TRIGger:BUS:B1:UART:RX:DATA: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 <NR1> Number of bytes (1 to 10).

Return parameter <NR1> Returns the number of bytes.

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

Set →

→ Query

**:TRIGger:BUS:B1:UART:TX:DATA:VALue**

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 <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.

Example1 :TRIGger:BUS:B1:UART:CONDition TXDATA  
:TRIGger:BUS:B1:UART:TX:DATA:SIZE 1  
:TRIGger:BUS:B1:UART:TX:DATA:VALue "1x1x0101"  
Sets the value to 1x1x0101

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

**:TRIGger:BUS: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.

**: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.

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

**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

<b>Parameter</b>	<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** :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> | ? }



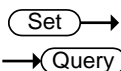
Related commands :TRIGger:BUS:B1:SPI:DATA:SIZE

Parameter	<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:CAN:CONDition**

Description Sets or returns the CAN trigger condition.

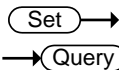
Syntax :TRIGger:BUS:B1:CAN:CONDition  
 {SO|FRAMeType|IDentifier|DATA|IDANDDATA|EOF|ACKMISS|STUFFERR|?}

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



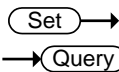
**:TRIGger:BUS:B1:CAN:FRAMeType**

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.



**:TRIGger:BUS:B1:CAN:IDentifier:MODe**

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

**:TRIGger:BUS:B1:CAN:IDentifier:VALue**

**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

<b>Parameter/ Return parameter</b>	<code>&lt;string&gt;</code>	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 →  
 → Query

**:TRIGger:BUS:B1:CAN:IDentifier:DIRection**

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

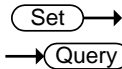
**Syntax**            :TRIGger:BUS:B1:CAN:IDentifier:DIRection {READ|WRITE|NOCARE|?}

<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



:TRIGger:BUS:B1:CAN:DATA:QUALifier

**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|M  
OREEQUAL|?}

<b>Parameter/ Return parameter</b>	<b>LESSthan</b>	Triggers when the data is less than the qualifier value.
	<b>MOREthan</b>	Triggers when the data is greater than the qualifier value.
	<b>EQUAL</b>	Triggers when the data is equal to the qualifier value.
	<b>UNEQUAL</b>	Triggers when the data is not equal to the qualifier value.
	<b>LESSEQUAL</b>	Triggers when the data is less than or equal to the qualifier value.
	<b>MOREEQUAL</b>	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


Set →  
 → Query

**:TRIGger:BUS:B1:CAN:DATA:SIZE**

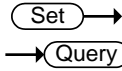
Description	Sets or returns the length of the data string in bytes for a CAN trigger.	
 Note	Only applicable when the condition is set to DATA or IDANDDATA.	
Syntax	:TRIGger:BUS:B1:CAN:DATA:SIZE {<NR1> ?}	
Parameter/ Return parameter	<NR1>	1~8 (bytes)
Example	:TRIGger:BUS:B1:CAN:DATA:SIZE? >1 :TRIGger:BUS:B1:CAN:DATA:SIZE 2 :TRIGger:BUS:B1:CAN:DATA:SIZE? >2	

Set →  
 → Query

**:TRIGger:BUS:B1:CAN:DATA:VALue**

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      :TRIGger:BUS:B1:CAN:DATA:SIZE 1
              :TRIGger:BUS:B1:CAN:DATA:VALue "01010X1X"
              :TRIGger:BUS:B1:CAN:DATA:VALue?
              >01010X1X
```



```
:TRIGger:BUS:B1:LIN:CONDition
```

**Description**      Sets or returns the LIN trigger condition.


**Syntax**            :TRIGger:BUS:B1:LIN:CONDition  
 {SYNCFieLd|IDentifier|DATA|IDANDDATA|WAKEup|SLEEP|ERRor|?}

<b>Parameter/ Return parameter</b>	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 MOREEQual ?}	
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** 



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	
		 → → 
<b>:TRIGger:BUS:B1:LIN:DATA:VALue</b>		

**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	<pre>:TRIGger:BUS:B1:LIN:ERRTYPE? &gt;SYNC :TRIGger:BUS:B1:LIN:ERRTYPE CHECKSUM :TRIGger:BUS:B1:LIN:ERRTYPE? &gt;CHECKSUM</pre>	

**:TRIGger:BUS:B1:LIN:IDentifier:VALue** 



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

Example :TRIGger:BUS:B1:LIN:CONDition ID  
 :TRIGger:BUS:B1:LIN:IDentifier:VALue "00X1X01X"  
 :TRIGger:BUS:B1:LIN:IDentifier:VALue?  
 >01100X1X01X

Set →  
 → Query

:TRIGger:BUS:B1:PARAllel:VALue

Description Sets or returns the binary data string to be used for a Parallel trigger.

Syntax :TRIGger:BUS:B1:PARAllel:VALue {string}  
 :TRIGger:BUS:B1:PARAllel:VALue?

Related Commands :BUS1:PARAllel:WIDth

Parameter	<code>&lt;string&gt;</code>	String contents: x = don't care 1 = binary 1 0 = binary 0
-----------	-----------------------------	--

Example :BUS1:PARAllel:WIDth 8  
 :TRIGger:BUS:B1:PARAllel:VALue "1X1X0101"  
 :TRIGger:BUS:B1:PARAllel:VALue?  
 >1X1X0101

Set →  
 → Query

:TRIGger:LOGic:INPut:CLOCK:SOURce

Description Sets or returns which digital channel is used as the clock source for the logic trigger. If none are selected, a pattern trigger will have to be used.

Syntax :TRIGger:LOGic:INPut:CLOCK:SOURce {NONE | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | ?}

Parameter/ Return parameter	None	No clock source selected. A pattern trigger will have to be set.
--------------------------------	------	--

	D0~D15	Sets one of the digital channels as the clock source.
Example	:TRIG:LOG:INP:CLOCK:SOUR D0 :TRIG:LOG:INP:CLOCK:SOUR? >D0	

**:TRIGger:LOGic:INPut:CLOCK:EDGE** (Set) →

Description	Sets the polarity of the clock source.	
Syntax	:TRIGger:LOGic:INPut:CLOCK:EDGE {RISe   FALL   EITher}	
Related Commands	:TRIGger:LOGic:INPut:CLOCK:SOURce	
Parameter	RISe	Sets the clock source on the rising edge.
	FALL	Sets the clock source on the falling edge.
	EITher	Sets the clock source to either rising or falling edge.
Example	:TRIG:LOG:INP:CLOCK:EDG RIS	

**:TRIGger:LOGic:FUNCTION** (Set) →  
→ (Query)

Description	Sets or queries the logical combination of the digital channels for the logic trigger.	
Syntax	:TRIGger:LOGic:FUNCTION{AND   NAND   NOR   OR   ?}	
Related Commands	:TRIGger:LOGic:PATtern:INPut:D<x>	
Parameter/ Return parameter	AND	Sets the logic combination to AND.
	NAND	Sets the logic combination to NAND.
	NOR	Sets the logic combination to NOR.
	OR	Sets the logic combination to OR.

Example :TRIGger:LOGic:FUNcTION?  
 >AND  
 :TRIGger:LOGic:FUNcTION NAND  
 Sets the logic combination of the digital channels to NAND.

**:TRIGger:LOGic:PATtern** → Query

Description Queries the conditions used for generating a logic pattern trigger in terms of input pattern, pattern trigger time and conditions.

Syntax :TRIGger:LOGic:PATtern?

Example :TIGger:LOGic:PATtern?  
 >:TRIGGER:LOGIC:PATTERN:INPUT:D0 HIGH; D1 X;  
 D2 X; D3 X; D4 X; D5 X; D6 X; D7 X; D8 X; D9 X; D10  
 X; D11 X; D12 X; D13 X; D14 X; D15 X;  
 :TRIGGER:LOGIC:PATTERN:WHEN TRUE;  
 :TRIGGER:LOGIC:PATTERN:DELTA TIME 1.000e-08;

Set →

**:TRIGger:LOGic:PATtern:INPut:D<x>** → Query

Description Sets or returns the logic level for the selected digital channel.

Syntax :TRIGger:LOGic:PATtern:INPut:D<x> {HIGH | LOW | X | ?}

Related Commands :TRIGger:LOGic:FUNcTION

Parameter/ Return parameter	<x>	Digital channel number 0~15.
	HIGH	Sets to logical high state.
	LOW	Sets to logical low state.
	X	Sets to "don't care" state.

Example :TRIGger:LOGic:PATtern:INPut:D0?  
 >HIGH

**:TRIGger:LOGic:PATtern:DELTatime** 



**Description** Sets or returns the pattern trigger delta time value.

**Syntax** :TRIGger:LOGic:PATtern:DELTatime {<NR3> | ?}

**Related Commands** :TRIGger:LOGic:PATtern:WHEn

<b>Parameter/Return parameter</b>	<NR3>	Pattern trigger time value in seconds 1e-9 (1 ns) to 10.0e0 (10 s).
-----------------------------------	-------	--

**Example**

```
:TRIG:LOG:PAT:DELT 8.960e-05
:TRIG:LOG:PAT:DELT?
>8.960e-05
```

**:TRIGger:LOGic:PATtern:WHEn** 



**Description** Sets or returns the pattern logic condition on which to trigger the oscilloscope.

**Syntax** :TRIGger:LOGic:PATtern:WHEn {TRUE | FALSE | LESSthan | MOREthan | EQUAL | UNEQual | ?}

**Related Commands** :TRIGger:LOGic:PATtern:DELTatime

<b>Parameter/Return parameter</b>	TRUE	Triggers when the defined input pattern is met.
	FALSE	Triggers when the defined input pattern is not met.
	LESSthan	Triggers when the defined input pattern is met during a time lower than the defined delta time.
	MOREthan	Triggers when the defined input pattern is met during a time greater than the defined delta time.

---

EQUAL	Triggers when the defined input pattern is met during a time equal to the defined delta time.
UNEQUAL	Triggers when the defined input pattern is met during a time other than the defined delta time.

---

Example

```
:TRIG:LOG:PAT:DELT FALSE
:TRIG:LOG:PAT:DELT?
>FALSE
```

## System Commands

:SYSTem:LOCK .....	167
:SYSTem:ERRor.....	167

### :SYSTem:LOCK

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

### :SYSTem:ERRor

Description	Queries the error queue. See the appendix on page 313 for details.	
Syntax	:SYSTem:ERRor?	
Return parameter	Returns the last message in the error queue.	
Example	:SYSTem:ERRor? +0, "No error."	

## Save/Recall Commands

:RECALL:SETUp .....	168
:RECALL:WAVEform .....	168
:SAVe:IMAGe .....	169
:SAVe:IMAGe:FILEFormat.....	169
:SAVe:IMAGe:INKSaver.....	170
:SAVe:SETUp .....	170
:SAVe:WAVEform.....	171
:SAVe:WAVEform:FILEFormat .....	172

### :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**

```
:RECALL:SETUp S1
```

Recalls setup setting S1 from memory.

```
:RECALL:SETUp "Disk:/DS0001.SET"
```

Recall the setup setting DS0001.SET from the internal memory.

### :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	:RECALL:WAVEform W1, REF1 Recalls the waveform stored in Wave1 to reference 1.	

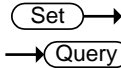
**:SAVE:IMAGe** (Set) →

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	Sets the file format to PNG
	BMP	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**

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	<table> <tr> <td>CH1~REF4,</td> <td>CH1~CH4, Math, D0~D15, REF1~4</td> </tr> <tr> <td>&lt;X&gt;</td> <td>1,2,3,4 (REF1, REF2, REF3, REF4)</td> </tr> <tr> <td>W1~W20</td> <td>Wave1~Wave20</td> </tr> <tr> <td>ALL</td> <td>All the displayed waveforms on screen.</td> </tr> <tr> <td>File path</td> <td>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.)</td> </tr> </table>	CH1~REF4,	CH1~CH4, Math, D0~D15, 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.)
CH1~REF4,	CH1~CH4, Math, D0~D15, 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:FILEFormat LSF :SAVe:WAVEform ALL, "Disk:/ALL001" Sets the file format to LSF. A folder named "ALL001" is created and saves all displayed waveforms to the "ALL001" directory in the LSF format.										

Example 3 :SAVE:WAVEform:FILEFormat FCSV  
 :SAVE:WAVEform ALL, "Disk:/ALL002"  
 Sets the file format to FCSV(fast CSV format). It then saves the all channel's waveforms to the root directory (Disk:/) of the internal flash disk in the CSV format (with the filename ALL002.CSV).

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

:SAVE:WAVEform:FILEFormat 
 →  
 →

Description	Sets the waveform savefile format.	
Syntax	:SAVE:WAVEform:FILEFormat {LSF   DCSV   FCSV   ?}	
Parameter	LSF	Sets the file format to the MDO-2000E/2000EX'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:DHCP

Set →  
→ Query

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

Set →

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 .....	175
:BUS1:STATE .....	175
:BUS1:TYPE .....	175
:BUS1:INPut .....	176
:BUS1:I2C:ADDReSS:RWINClude .....	176
:BUS1:I2C:SCLK:SOURce .....	177
:BUS1:I2C:SDA:SOURce .....	177
:BUS1:PARAllel:BIT<x>:SOURce .....	177
:BUS1:PARAllel:CLOCK:EDGE .....	178
:BUS1:PARAllel:CLOCK:SOURce .....	178
:BUS1:PARAllel:WIDth .....	178
:BUS1:UART:BITRate .....	179
:BUS1:UART:DATABits .....	179
:BUS1:UART:PARItY .....	179
:BUS1:UART:PACKet .....	180
:BUS1:UART:POLARity .....	180
:BUS1:UART:EOFPAcket .....	180
:BUS1:UART:TX:SOURce .....	181
:BUS1:UART:RX:SOURce .....	181
:BUS1:SPI:SCLK:POLARity .....	182
:BUS1:SPI:SS:POLARity .....	182
:BUS1:SPI:WORDSize .....	182
:BUS1:SPI:BITORder .....	183
:BUS1:SPI:SCLK:SOURce .....	183
:BUS1:SPI:SS:SOURce .....	183
:BUS1:SPI:MOSI:SOURce .....	184
:BUS1:SPI:MISO:SOURce .....	184
:BUS1:DISPlay:FORMAt .....	185
:LISTer:DATA .....	185
:BUS1:CAN:SOURce .....	185
:BUS1:CAN:PROBe .....	186
:BUS1:CAN:SAMPLEpoint .....	186

:BUS1:CAN:BITRate.....	186
:BUS1:LIN:BITRate .....	187
:BUS1:LIN:IDFORmat .....	187
:BUS1:LIN:POLARity.....	188
:BUS1:LIN:SAMPLEpoint .....	188
:BUS1:LIN:SOURce .....	188
:BUS1:LIN:STANDard .....	189

**:BUS1** → Query

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

**:BUS1:STATE** Set →  
→ 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.

**:BUS1:TYPE** Set →  
→ 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	Sets the bus to UART mode.
	I2C	Sets the bus to I2C mode.
	SPI	Sets the bus to SPI mode.
	PARAllel	Sets the bus to parallel mode.
	CAN	Sets the bus to CAN mode.
	LIN	Sets the bus to LIN mode.

Example :BUS1:TYPE SPI  
Sets the bus to SPI mode.

Set →

**:BUS1:INPut**

→ Query

Description Sets or returns the bus source.

Syntax :BUS1:INPut { ANAlOG | DIGItal | ? }

Parameter/Return parameter	ANAlOG	Sets the bus source as analog inputs.
	DIGItal	Sets the bus source as digital inputs.

Example1 :BUS1:INPut?  
>ANALOG

Set →

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

→ Query

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.

→  
 →

**:BUS1:I2C:SCLK:SOURce**

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 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | ? }

Parameter/Return parameter	CH1 to CH4	Analog channels 1 ~ 4.
	D0 to D15	Digital channels D0~D15

Example :BUS1:I2C:SCLK:SOURce CH1  
Sets channel 1 as the SCLK source.

→  
 →

**:BUS1:I2C:SDA:SOURce**

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| D0| D1| D2| D3| D4| D5| D6| D7| D8| D9| D10| D11| D12| D13| D14| D15|? }

Parameter/Return parameter	CH1 to CH4	Analog channels 1 ~ 4.
	D0 to D15	Digital channels D0~D15

Example :BUS1:I2C:SDA:SOURce CH1  
Sets channel 1 as the SDA source.

→  
 →

**:BUS1:PARAllel:BIT<x>:SOURce**

Description Sets or returns the parallel bit source for B1.

Syntax                    BUS1:PARAllel:BIT<x>:SOURce {D0| D1| D2| D3| D4| D5| D6| D7| D8| D9| D10| D11| D12| D13| D14| D15}  
 BUS1:PARAllel:BIT<x>:SOURce?

Parameter/Return parameter    <x>                    the bit number  
 D0 to D15            Set the bit source B<x>

Set →

→ Query

**:BUS1:PARAllel:CLOCK:EDGE**

Description            Sets or returns the parallel clock edge for bus1

Syntax                    BUS1:PARAllel:CLOCK:EDGE{RISe| FALL|EITHer| OFF}  
 BUS1:PARAllel:CLOCK:EDGE?

Parameter              EITHer                Set either edge as the clock edge.  
 RISe                    Set the rising edge as the clock edge.  
 FALL                    Set the rising edge as the clock edge.  
 OFF                     Turn off the clock edge.

Set →

→ Query

**:BUS1:PARAllel:CLOCK:SOURce**

Description            Sets or returns the Parallel bus1

Syntax                    BUS1:PARAllel:CLOCK:SOURce{D0| D1| D2| D3| D4| D5| D6| D7| D8| D9| D10| D11| D12| D13| D14| D15}  
 BUS1:PARAllel:CLOCK:SOURce?

Parameter              D0 to D15            Set the clock source

Set →

→ Query

**:BUS1:PARAllel:WIDth**

Description            Sets or returns the number of bits used for the width of the Parallel bus1.

Syntax                    :BUS1:PARAllel:WIDth <NR1>  
 :BUS1:PARAllel:WIDth?

Parameter              <NR1>                The number of bits.


**:BUS1:UART:BITRate**

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	


**:BUS1:UART:DATABits**

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


**:BUS1:UART:PARITY**

Description	Sets or queries the UART bus parity.	
Syntax	:BUS1:UART:PARITY { <NR1>   ? }	
Parameter/Return parameter	<NR1>	0: None
		1: Odd parity
		2: Even parity

Example :BUS1:UART:PARity 1  
Sets the parity to odd.

Set →

:BUS1:UART:PACKEt

→ Query

Description Sets or queries the UART packet setting.

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

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

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

Set →

:BUS1:UART:POLARity

→ 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

Set →

:BUS1:UART:EOFPacket

→ 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 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | ? }

Parameter/Return parameter	OFF	Off, no Tx source
	CH1 to CH4	Analog channels CH1 to CH4
	D0 to D15	Digital channels D0 to D15

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 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | ? }

Parameter/Return parameter	OFF	Off, no Rx source
	CH1 to CH4	Analog channels CH1 to CH4
	D0 to D15	Digital channels D0 to D15

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

:BUS1:SPI:SCLK:POLARity Set →  
→ 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.

:BUS1:SPI:SS:POLARity Set →  
→ Query

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.

:BUS1:SPI:WORDSize Set →  
→ Query

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.

→  
 →

**: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 D0 to D15	Analog channels CH1 to CH4 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 D0 to D15	Analog channels CH1 to CH4 Digital channels D0 to D15

Example :BUS1:SPI:SS:SOURce CH1  
Sets channel 1 as the SPI SS source.

:BUS1:SPI:MOSI:SOURce (Set) →  
→ (Query)

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.

:BUS1:SPI:MISO:SOURce (Set) →  
→ (Query)

Description Sets or queries which channel is used for the SPI MISO source.

Syntax :BUS1:SPI:MISO:SOURce { OFF | CH1 | CH2 | CH3 | CH4 | 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.



**:BUS1:DISPlay:FORMAt** 



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** 

Description	Returns the Event Table data as a binary block data.	
Syntax	:LISTer:DATA?	
Return Parameter	Returns the event table as binary block data. The binary block data contains comma separated data with new lines at the end of each row.	

**:BUS1:CAN:SOURce** 



Description	Sets or returns the CAN input source.	
Syntax	:BUS1:CAN:SOURce { CH1   CH2   CH3   CH4   D0   D1   D2   D3   D4   D5   D6   D7   D8   D9   D10   D11   D12   D13   D14   D15   ? }	
Parameter/Return parameter	CH1 ~ CH4	Analog channel source
	D0 to D15	Digital channels D0 to D15
Example	:BUS1:CAN:SOURce? >CH1 Returns the CAN source.	

Set →  
→ Query

**:BUS1:CAN:PROBe**

Description	Sets or returns the signal type of the CAN bus.	
Syntax	:BUS1:CAN:PROBe {CANH   CANL   TX   RX   ? }	
Parameter/Return parameter	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 →  
→ Query

**:BUS1:CAN:BITRate**

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

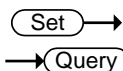
Example

```
:BUS1:CAN:BITRate?
>1000000

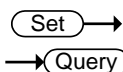
:BUS1:CAN:BITRate rate800k
:BUS1:CAN:BITRate?
>800000

:BUS1:CAN:BITRate 25000
:BUS1:CAN:BITRate?
>25000
```

**:BUS1:LIN:BITRate**



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.	



**:BUS1:LIN:IDFormat**

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 →

→ Query

**:BUS1:LIN:POLARity**

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 →

→ Query

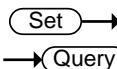
**:BUS1:LIN:SOURce**

Description	Sets or returns the LIN data source.	
-------------	--------------------------------------	--

Syntax :BUS1:LIN: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 ~ CH4	Analog channel source
	D0 to D15	Digital channels D0 to D15

Example :BUS1:LIN:SOURCE?  
>CH1  
Returns the LIN source.



**:BUS1:LIN:STANDARD**

Description Sets or returns the LIN standard.

Syntax :BUS1:LIN:STANDARD {V1X|V2X|BOTH|?}

Parameter/Return parameter	V1X	Lin standard version 1.x
	V2X	Lin standard version 2.x
	BOTH	Both standards

Example :BUS1:LIN:STANDARD?  
>BOTH  
Returns the LIN standard.

## Mark Commands

:MARK.....	190
:MARK:CREATE.....	190
:MARK:DELEte.....	191

### :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**

Set →

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:COPIY .....	193
:SEARCH:STATE .....	194
:SEARCH:TOTAL .....	194
:SEARCH:TRIGger:TYPe .....	194
:SEARCH:TRIGger:SOURce.....	195
:SEARCH:TRIGger:EDGE:SLOP .....	195
:SEARCH:TRIGger:LEVel .....	196
:SEARCH:TRIGger:HLEVel .....	196
:SEARCH:TRIGger:LLEVel .....	197
:SEARCH:TRIGger:PULSEWidth:POLarity .....	197
:SEARCH:TRIGger:RUNT:POLarity.....	197
:SEARCH:TRIGger:RISEFall:SLOP .....	198
:SEARCH:TRIGger:PULSE:WHEn.....	198
:SEARCH:TRIGger:PULSE:TIME.....	199
:SEARCH:TRIGger:RUNT:WHEn .....	199
:SEARCH:TRIGger:RUNT:TIME .....	200
:SEARCH:TRIGger:RISEFall:WHEn .....	200
:SEARCH:TRIGger:RISEFall:TIME .....	201
:SEARCH:TRIGger:BUS:TYPe.....	201
:SEARCH:TRIGger:BUS:B1:I2C:CONDition .....	202
:SEARCH:TRIGger:BUS:B1:I2C:ADDRes:MODE ...	203
:SEARCH:TRIGger:BUS:B1:I2C:ADDRes:TYPe .....	203
:SEARCH:TRIGger:BUS:B1:I2C:ADDRes:VALue....	204
:SEARCH:TRIGger:BUS:B1:I2C:ADDRes	
:DIRection .....	205
:SEARCH:TRIGger:BUS:B1:I2C:DATA:SIZE .....	205
:SEARCH:TRIGger:BUS:B1:I2C:DATA:VALue .....	206
:SEARCH:TRIGger:BUS:B1:UART:CONDition.....	206
:SEARCH:TRIGger:BUS:B1:UART:RX:DATA:SIZE....	207
:SEARCH:TRIGger:BUS:B1:UART:RX:DATA:VALue .	208
:SEARCH:TRIGger:BUS:B1:UART:TX:DATA:SIZE ....	209
:SEARCH:TRIGger:BUS:B1:UART:TX:DATA:VALue .	209
:SEARCH:TRIGger:BUS:B1:SPI:CONDition.....	210



:SEARCH:TRIGger:BUS:B1:SPI:DATa:SIze..... 210  
 :SEARCH:TRIGger:BUS:B1:SPI:DATa:MISO  
 :VALue ..... 211  
 :SEARCH:TRIGger:BUS:B1:SPI:DATa:MOSI  
 :VALue ..... 212  
 :SEARCH:TRIGger:BUS:B1:CAN:CONDition ..... 212  
 :SEARCH:TRIGger:BUS:B1:CAN:FRAMeType ..... 213  
 :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODe 214  
 :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:VALue . 214  
 :SEARCH:TRIGger:BUS:B1:CAN:IDentifier:  
 DIRection ..... 215  
 :SEARCH:TRIGger:BUS:B1:CAN:DATa:QUALifier.. 215  
 :SEARCH:TRIGger:BUS:B1:CAN:DATa:SIze ..... 216  
 :SEARCH:TRIGger:BUS:B1:CAN:DATa:VALue ..... 217  
 :SEARCH:TRIGger:BUS:B1:LIN:CONDition ..... 218  
 :SEARCH:TRIGger:BUS:B1:LIN:DATa:QUALifier ... 218  
 :SEARCH:TRIGger:BUS:B1:LIN:DATa:SIze ..... 219  
 :SEARCH:TRIGger:BUS:B1:LIN:DATa:VALue ..... 220  
 :SEARCH:TRIGger:BUS:B1:LIN:ERRTYpe ..... 220  
 :SEARCH:TRIGger:BUS:B1:LIN:IDentifier:VALue... 221  
 :SEARCH:FFTPeak:METhod ..... 222  
 :SEARCH:FFTPeak:METhod:MPEak..... 222  
 :SEARCH:FFTPeak:SI NFo..... 223  
 :SEARCH:FFTPeak:LIST ..... 223

**:SEARCH:COpy**



Description	Copies the search settings to the trigger settings or copies the trigger settings to the search settings.	
Syntax	:SEARCH:COpy {SEARCHtotrigger TRIGgertosearch}	
Parameter	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 | ? }

Parameter/Return parameter	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 →

:SEARCH:TRIGger:TYPe

→ Query

Description Sets or queries the search trigger type.

Syntax :SEARCH:TRIGger:TYPe { EDGe | PULSEWidth | RUNT | RISEFall | FFTPeak | LOGic | BUS | ? }

Parameter/Return parameter	EDGe	Edge trigger
	PULSEWidth	Pulse width trigger
	RUNT	Runt trigger
	RISEFall	Rise and Fall trigger

	FFTPeak	FFT Peak trigger
	LOGic	Logic trigger
	BUS	Bus trigger

**Example** :SEARCH:TRIGger:TYPE EDGE  
Sets the search trigger to the edge type.

:SEARCH:TRIGger:SOURce (Set) →  
→ (Query)

**Description** Sets or queries the search trigger source.

**Syntax** :SEARCH:TRIGger:SOURce {CH1 | CH2 | CH3 | CH4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | ? }

<b>Parameter/Return parameter</b>	CH1 to CH4	Channel 1 to Channel 4
	D0 to D15	Digital channels D0 to D15

**Example** :SEARCH:TRIGger:SOURce CH1  
Sets the search trigger source as CH1.

:SEARCH:TRIGger:EDGE:SLOP (Set) →  
→ (Query)

**Description** Sets or queries the search trigger slope.

**Syntax** :SEARCH:TRIGger:EDGE:SLOP { RISE | FALL | EITHER | ? }

**Related commands** :SEARCH:TRIGger:TYPE

<b>Parameter</b>	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.


Set →  
→ Query

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


Set →  
→ Query

**:SEARCH:TRIGger:HLEVel**

Description	Sets or queries the high level search trigger.	
 Note	Applicable for Rise and Fall/Pulse Runt search triggers.	
Syntax	:SEARCH:TRIGger:HLEVel { <NRf>   ?}	
Related commands	:SEARCH:TRIGger:TYPe	
Parameter	<NRf>	High level value.
Return parameter	<NR3>	Returns the high level search trigger.
Example	:SEARCH:TRIGger:HLEVel 3.30E-1 Sets the high level search trigger to 330mV/mA.	

Set →  
 → Query

**:SEARCH:TRIGger:LLeVel**

Description	Sets or queries the low level search trigger.	
 Note	Applicable for Rise and Fall/Pulse Runt triggers.	
Syntax	:SEARCH:TRIGger:LLeVel { <NRf>   ?}	
Related commands	:SEARCH:TRIGger:TYPe	
Parameter	<NRf>	Low level value.
Return parameter	<NR3>	Returns the low level.
Example	:SEARCH:TRIGger:LLeVel -3.30E-3 Sets the low level search trigger to 330mV/mA.	

Set →  
 → Query

**: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	
Parameter	POSitive	Positive polarity
	NEGative	Negative polarity
Return parameter	Returns the pulse width polarity.	
Example	:SEARCH:TRIGger:PULSEWidth:POLarity POSitive Sets the pulse width polarity to positive.	

Set →  
 → Query

**: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	
Parameter	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.	
	<span style="border: 1px solid black; border-radius: 10px; padding: 2px 5px;">Set</span> → → <span style="border: 1px solid black; border-radius: 10px; padding: 2px 5px;">Query</span>	
	<b>:SEARCH:TRIGger:PULSe:WHEN</b>	

Description	Sets or queries the pulse width search trigger conditions.	
-------------	--	--

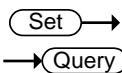
Syntax :SEARCH:TRIGger:PULSe:WHEn {MOREthan | LESSthan | EQual | UNEQual | ?}

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 (≠).



**:SEARCH:TRIGger:PULSe:TIME**

Description Sets or queries the pulse width search trigger time.

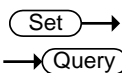
Syntax :SEARCH:TRIGger:PULSe:TIME {<NRf> | ?}

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.



**: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 →

:SEARCH:TRIGger:RUNT:TIME → Query

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 →

:SEARCH:TRIGger:RISEFall:WHEn → Query

Description Sets or queries the rise and fall search trigger conditions.

Syntax :SEARCH:TRIGger:RISEFall:WHEn {MOREthan | LESSthan | Equal | UNEQual | ? }

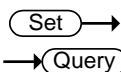
Related commands	:SEARCH:TRIGger:TYPE
	:SEARCH:TRIGger:RISEFall:TIME



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
	SPI	SPI mode
	UART	UART mode
	CAN	CAN mode
	LIN	LIN mode

PARAllel Parallel mode

Example :SEARCH:TRIGger:BUS:TYPe?  
UART

:SEARCH:TRIGger:BUS:B1:I2C:CONDition Set →  
← Query

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.

Set →  
 → Query

**: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.	

Set →  
 → Query

**:SEARCH:TRIGger:BUS:B1:I2C:ADDRess:TYPE**

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:ADDRess:TYPE {GENeralcall   STARtbyte   HSmode   EEPROM   CBUS   ? }	
Related commands	:SEARCH:TRIGger:BUS:B1:I2C:CONDition	
Parameter	GENeralcall	Set a general call address (0000 000 0).
	STARtbyte	Set a start byte address. (0000 000 1)
	HSmode	Set a high-speed mode address. (0000 1xx x)

EEPROM	Set an EEPROM address. (1010 xxx x)
CBUS	Set a CBUS address. (0000 001 x)

Return Parameter Returns the address type

Example :SEARCH:TRIGger:BUS:B1:I2C:ADDRes:TYPe?  
CBUS

Set →  
 → Query

**:SEARCH:TRIGger:BUS:B1:I2C:ADDRes:VALue**

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	<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 in binary.

Example 1 :SEARCH:TRIGger:BUS:B1:I2C:ADDRes:MODE  
ADDR7  
  
:SEARCH:TRIGger:BUS:B1:I2C:ADDRes:VALue  
"xxx0101"  
  
Sets the address to XXX0101

Example 2 :SEARCH:TRIGger:BUS:B1:I2C:ADDRes:VALue?  
XXX0101

**:SEARCH:TRIGger:BUS:B1:I2C:ADDRes** (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 I<sup>2</sup>C search trigger is set to trigger on Address or Address/Data

**Syntax** :SEARCH:TRIGger:BUS:B1:I2C:ADDRes:DIRection { READ | WRITE | NOCARE | ? }

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


<b>Parameter</b>	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:ADDRes: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 I<sup>2</sup>C 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

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

<b>Return parameter</b>	<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:CONDition (Set) →  
 → (Query)

Description Sets or queries the UART search triggering condition.

Syntax	:SEARCH:TRIGger:BUS:B1:UART:CONDition { RXSTArt   RXDATA   RXENDPacket   TXSTArt   TXDATA   TXENDPacket   TXPARIttyerr   RXPARIttyerr   ? }	
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.
	RXPARIttyerr	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.
	TXPARIttyerr	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 Set → Query

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	<sting>	The number of characters in the string depends on the data size setting. The string must be enclosed in double quotes, "string".  x = don't care 1 = binary 1 0 = binary 0

Return Parameter Returns the data value.


Example1 :SEARCH:TRIGger:BUS:B1:UART:CONDition RXDATA  
:SEARCH:TRIGger:BUS:B1:UART:RX:DATA:SIZE 1  
:SEARCH:TRIGger:BUS:B1:UART:RX:DATA:VALue "1x1x0101"  
Sets the value to 1x1x0101

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



Set →  
 → Query

---

Description	Sets or queries the number of bytes for UART data.	
 Note	This setting only applies when the UART search trigger is set to trigger on 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.	

Set →  
 → Query

---

Description	Sets or queries the search triggering data value for the UART bus when the bus is set to trigger on Tx Data.	
Syntax	:SEARCH:TRIGger:BUS:B1:UART:TX:DATA:VALue {<string>   ?}	
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

:SEARCH:TRIGger:BUS:B1:SPI:CONDition Set →  
→ Query

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 :SEARCH:TRIGger:BUS:B1:SPI:CONDition MISO  
 Sets the SPI bus to trigger on MISO.

:SEARCH:TRIGger:BUS:B1:SPI:DATA:SIZE Set →  
→ Query

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.	

Set →

:SEARCH:TRIGger:BUS:B1:SPI:DATA:MISO:VALue → Query

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

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

**Related commands** :SEARCH:TRIGger:BUS:B1:SPI: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** :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

<b>Parameter</b>	<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.

**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

Set →

:SEARCH:TRIGger:BUS:B1:CAN:CONDition → Query

**Description** Sets or returns the CAN search trigger condition.

**Syntax** :SEARCH:TRIGger:BUS:B1:CAN:CONDition {SOFFRAMEtype|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	
	:SEARCH:TRIGger:BUS:B1:CAN:FRAMEType	
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.

Set →

:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:MODE → 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

Set →

:SEARCH:TRIGger:BUS:B1:CAN:IDentifier:VALue → Query

Description Sets or returns the identifier string used for the CAN search trigger.



Note

Only applicable when the search triggering condition is set to DATA 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".
--------------------------------	----------	--


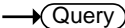
String contents:

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

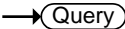
:SEARCH:TRIGger:BUS:B1:CAN:IDentifier   
 :DIRection 

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

:SEARCH:TRIGger:BUS:B1:CAN:DATA:SIZE (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	

Set →  
 → Query

---

:SEARCH:TRIGger:BUS:B1:CAN:DATA:VALue

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> ?}	
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:DATA:SIZE 1 :SEARCH:TRIGger:BUS:B1:CAN:DATA:VALue "01010X1X" :SEARCH:TRIGger:BUS:B1:CAN:DATA:VALue? >01010X1X	

:SEARCH:TRIGger:BUS:B1:LIN:CONDition (Set) →  
→ (Query)

**Description**      Sets or returns the LIN search trigger condition.

**Syntax**            :SEARCH:TRIGger:BUS:B1:LIN:CONDition  
                         {SYNCField|IDentifier|DATA|IDANDDATA|WAKEup|S  
                         LEEP|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 (Set) →  
→ (Query)

**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 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:LIN:DATa:QUALifier?
>EQUAL
:SEARCH:TRIGger:BUS:B1:LIN:DATa:QUALifier
MOREthan
:SEARCH:TRIGger:BUS:B1:LIN:DATa:QUALifier?
>MORETHAN
```

:SEARCH:TRIGger:BUS:B1:LIN:DATa:SIZE (Set) →  
→ (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

Set →

:SEARCH:TRIGger:BUS:B1:LIN:ERRTYPE

→ Query


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	

Set →  
 → Query

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

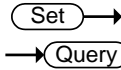
Description	Sets or returns the identifier string to be used for the LIN search trigger.
-------------	--

 Note	Only applicable when the condition is set to ID or IDANDDATA.
--	---

Syntax	:SEARCH:TRIGger:BUS:B1:LIN:IDentifier:VALue {<string> ?}	
--------	---	--

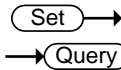
Parameter/ Return parameter	<string>	The size of the string depends on the data size setting. The string must be enclosed in double quotes, "string".
		String contents: x = don't care 1 = binary 1 0 = binary 0

Example	:SEARCH:TRIGger:BUS:B1:LIN:CONDition ID :SEARCH:TRIGger:BUS:B1:LIN:IDentifier:VALue "00X1X01X" :SEARCH:TRIGger:BUS:B1:LIN:IDentifier:VALue? >01100X1X01X	
---------	--	--



**:SEARCH:FFTPeak:MEtHod**

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:SIInfo**

Description Sets or returns the State Info to “Mark” or “Peak”.

Related Commands :SEARCH:TRIGger:TYPE

Syntax :SEARCH:FFTPeak:SIInfo {MARK | PEAK | ?}

Parameter/ Return parameter	MARK	Sets the State Info to Mark.
	PEAK	Sets the State Info to Peak.

```
Example      :SEARCH:FFTPeak:SIInfo?
             >PEAK
             :SEARCH:FFTPeak:SIInfo mark
             :SEARCH:FFTPeak:SIInfo?
             >MARK
```

**: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;

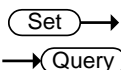
---



## Label Commands

:CHANnel<X>:LABel.....	225
:CHANnel<X>:LABel:DISPlay .....	226
:REF<X>:LABel .....	226
:REF<X>:LABel:DISPlay .....	227
:BUS1:LABel.....	228
:BUS1:LABel:DISPlay .....	228
:D<x>:LABel .....	229
:D<x>:LABel:DISPlay.....	230
:DIGital:ANAlog:A<x>:LABel .....	230
:DIGital:ANAlog:A<x>:LABel:DISPlay .....	231
:SET<X>:LABel .....	232



### :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

:CHANnel<X>:LABel:DISPlay  

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 :CHANnel1:LABel "CH1"  
 :CHANnel1:LABel:DISPlay ON  
 :CHANnel1:LABel:DISPlay?  
 ON  
 Sets the channel 1 label to "CH1" and then turns the label display on. The query return shows that the label is on.

:REF<X>:LABel  

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

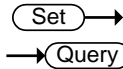


Description	Turns the label on/off for the selected reference waveform or returns its status.	
Syntax	:REF<X>:LABel:DISPlay { OFF   ON   ? }	
Related commands	:REF<X>:LABel	
Parameter	<X>	Reference waveform 1, 2, 3, 4
	OFF	Turns the file label off for the selected reference waveform.
	ON	Turns the file 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.

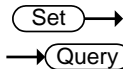


**:BUS1:LABel**

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



**: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 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 :BUS1:LABel "Bus"  
 :BUS1:LABel:DISPlay ON  
 :BUS1:LABel:DISPlay?  
 ON  
 Sets the label for the bus to "Bus" and then turns the label display on. The query return shows that the label is on.



Description Sets or returns the waveform label for digital channels.

Syntax :D<x>:LABel {<string> | ?}

Related commands :D<x>:LABel:DISPlay

Parameter <x> Digital channel number D0~D15  
 <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	<code>&lt;string&gt;</code>	Returns the label for the bus. No return indicates that there has not been a file label assigned for bus.
------------------	-----------------------------	---

Example           :D1:LABel "D1"  
                   Sets the digital channel 1 label as "D1".

Set →  
 → Query

**:D<x>:LABel:DISPlay**

Description	Turns the label on/off for the selected digital channel or returns its status.
-------------	--

Syntax	:D<x>:LABel:DISPlay { OFF   ON   ? }
--------	--------------------------------------

Related commands	:D<x>:LABel
------------------	-------------

Parameter	OFF	Turns the file label off for the selected digital channel.
	ON	Turns the file label on for the selected digital channel.

Return parameter	Returns the status of the file label for the digital channel (ON, OFF).
------------------	---

Example           :D1:LABel "D1"  
                   :D1:LABel?  
                   >D1  
                   :D1:LABel:DISPlay ON  
                   D1:LABel:DISPlay?  
                   ON  
                   Sets the label for the D1 channel to "D1" and then turns the label display on. The query return shows that the label is on.

Set →  
 → Query

**:DIGital:ANALog:A<x>:LABel**

Description	Sets or returns the label for analog waveforms.
-------------	---

Syntax	:DIGital:ANALog:A<x>:LABel {<string>   ?}
--------	---

Related commands :DIGital:ANALog:A<x>:LABel:DISPlay

Parameter	<x>	Analog waveform number (1~2).
	<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.

Example :DIG:ANA:A1:LAB "A1"  
Sets the analog waveform 1 label as "A1".



Description Turns the label on/off for the analog waveform or returns its status.

Syntax :DIGital:ANALog:A<x>:LABel:DISPlay { OFF | ON | ? }

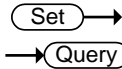
Related commands :DIGital:ANALog:A<x>:LABel

Parameter	OFF	Turns the file label off for the analog waveform.
	ON	Turns the file label on for the analog waveform.

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

Example :DIGital:ANALog:A1:LABel "A1"  
:DIGital:ANALog:A1:LABel:DISPlay ON  
:DIGital:ANALog:A1:LABel:DISPlay?  
ON

Sets the label for the analog waveform to "A1" and then turns the label display on. The query return shows that the label is on.



**:SET<X>:LABel**

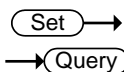
Description	Sets or returns the file label for the selected setup.	
Syntax	:SET<X>:LABel {<string>   ?}	
Related commands	:SET<X>:LABel:DISPlay	
Parameter	<X>	Setup number 1 to 20
	<string>	The string must be no more than 8 characters and only contain alphanumeric characters in addition to period, dash 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	:SET1:LABel "SET1_lab" Sets the label for setup 1 as "SET1_lab".	
Example2	:SET1:LABel? SET1_lab	



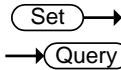
## Segment Commands

:SEGMENTS:STATE .....	233
:SEGMENTS:CURRENT .....	234
:SEGMENTS:TOTALNUM .....	234
:SEGMENTS:TIME .....	235
:SEGMENTS:DISPALL .....	235
:SEGMENTS:MEASURE:MODE .....	235
:SEGMENTS:MEASURE:PLOT:SOURCE .....	236
:SEGMENTS:MEASURE:PLOT:DIVIDE .....	236
:SEGMENTS:MEASURE:PLOT:SELECT .....	237
:SEGMENTS:MEASURE:PLOT:RESULTS.....	237
:SEGMENTS:MEASURE:TABLE:SOURCE .....	238
:SEGMENTS:MEASURE:TABLE:SELECT.....	238
:SEGMENTS:MEASURE:TABLE:LIST.....	239
:SEGMENTS:MEASURE:TABLE:SAVE .....	239
:SEGMENTS:SAVE .....	240
:SEGMENTS:SAVE:SOURCE .....	240
:SEGMENTS:SAVE:SELECT:START .....	240
:SEGMENTS:SAVE:SELECT:END .....	241

### :SEGMENTS:STATE



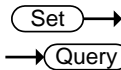
Description	Turns the segmented memory function on/off or queries its state.	
Syntax	:SEGMENTS:STATE { OFF   ON   ? }	
Related commands	:RUN :STOP	
Parameter/ Return parameter	OFF	Turns the segmented memory off.
	ON	Turns the segmented memory on.
Example1	:SEGMENTS:STATE ON Turns segmented memory on.	



**:SEGMents:CURRent**

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.



**: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**

Set →

→ Query

---

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**

Set →

→ Query

---

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 Set →  
→ Query

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 Set →  
→ Query

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.

Set →

**:SEGMents:MEASure:PLOT:SElect**

→ Query

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

**:SEGMents:MEASure:TABLE:SOURce**

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

**:SEGMents:MEASure:TABLE:SElect**

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.

**:SEGMents:MEASure:TABLE:LIST → Query**

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 MDO-2074EG, 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 →

**:SEGMents:SAVe:SOURce**

→ Query

Description	Sets or queries the source segment waveform to save.	
Syntax	:SEGMents:SAVe: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~CH4	Channel 1 to 4.
	D0~D15	Digital channels D0~D15
Example	:SEGMents:SAVe:SOURce CH1 >Sets the source to CH1.	

Set →

**:SEGMents:SAVe:SELEct:STARt**

→ Query

Description	Sets or queries the starting segment to save from. The number of possible segments depends on the record length.
Syntax	:SEGMents:SAVe:SELEct:STARt {SETTOMIN   SETTOMAX   <NR1>   ? }



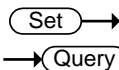
Related commands	:SEGMENTS:TOTALNUM	
Parameter/ Return parameter	SETTOMIN	Sets the starting segment to min segment.
	SETTOMAX	Sets the starting segment to the max segment.
	<NR1>	Sets the segment to 1~29000
Example	:SEGMENTS:SAVE:SELECT:START 2 Sets the starting segment to segment number 2.	
	:SEGMENTS:SAVE:SELECT:END	
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

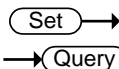
The DVM commands are only available when the optional DVM software is installed.

:DVM:STATE .....	242
:DVM:SOURce .....	242
:DVM:MODE .....	243
:DVM:VALue .....	243

### :DVM:STATE



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.	



### :DVM:SOURce

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.	

→ Query

---

**:DVM:VALue**

---

Description	Returns the measurement value of the selected mode.	
Syntax	:DVM:VALue?	
Related commands	:DVM:SOURce :DVM:STATE :DVD: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.....	244
:GONogo:EXECute .....	244
:GONogo:FUNCTion.....	245
:GONogo:NGCount .....	245
:GONogo:NGDefine.....	245
:GONogo:SOURce .....	246
:GONogo:VIOLation .....	246
:GONogo:SCRipt .....	246
:TEMPlate:MODE .....	247
:TEMPlate:MAXimum.....	247
:TEMPlate:MINimum.....	247
:TEMPlate:POSition:MAXimum.....	248
:TEMPlate:POSition:MINimum .....	248
:TEMPlate:SAVe:MAXimum .....	248
:TEMPlate:SAVe:MINimum .....	249
:TEMPlate:TOLerance.....	249
:TEMPlate:SAVe:AUTo .....	249

### :GONogo:CLear

Set →

Description Clears the Go/NoGo counter.

Syntax :GONogo:CLear

### :GONogo:EXECute

Set →

→ Query

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** Set →

Description Initializes the Go/NoGo APP. This must be run after the Go/NoGo APP has been started.

Syntax :GONogo:FUNCTION

**:GONogo:NGCount** → Query

Description Returns the Go/NoGo counter.

Syntax :GONogo:NGCount{?}

Return parameter Returns a string in the following format “number of violations,total tests”

Example :GONogo:NGCount?  
> 3,25  
Indicates that 3 violations occurred over 25 tests.

**:GONogo:NGDefine** Set →  
→ Query

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/ Return Parameter	STOP The waveform will be frozen. CONTInue Ignore the violation.
Example	:GONogo:VIOLation STOP Sets violation action to STOP.

Set →

---

:GONogo:SCRipt	
Description	Activates/Deactivates the Go/NoGo APP or queries its state.
Syntax	:GONogo:SCRipt {OFF   ON   ?}
Parameter/ Return Parameter	ON Turns Go/NoGo APP on. OFF Turns the Go/NoGo APP off.
Example	:GONogo:SCRipt? >ON The Go/NoGo script is on.

Set →  
 → Query

**:TEMPlate:MODE**

Description	Sets or returns the Go/NoGo template mode.	
Syntax	:TEMPlate:MODE {MAXimum MINimum AUTO ?}	
Parameter/ Return Parameter	MAXimum	Maximum template
	MINimum	Minimum template
	AUTO	Auto template
Example	:TEMPlate:MODE AUTO Sets the template mode to AUTO.	

Set →  
 → Query

**:TEMPlate:MAXimum**

Description	Defines or queries which waveform memory (REF1 or W1~W20) is set to the maximum template.	
Syntax	:TEMPlate:MAXimum {REF1 W1~W20 ?}	
Parameter/ Return Parameter	REF1	Reference one
	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 (REF2 or W1~W20) is set to the minimum template.	
Syntax	:TEMPlate:MINimum {REF2 W1~W20 ?}	
Parameter/ Return Parameter	REF2	Reference one
	W1~W20	Waveform memory 1 to 20

Example :TEMPlate:MINimum REF2  
Saves the minimum template to REF2.

Set →

**:TEMPlate:POSition:MAXimum**

→ Query

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 →

**:TEMPlate:POSition:MINimum**

→ Query

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.

**:TEMPlate:SAVE:MAXimum**

Set →

Description Saves the maximum template.

Syntax :TEMPlate:SAVE:MAXimum



**:TEMPlate:SAVe:MINimum** Set →

Description Saves the maximum template.

Syntax :TEMPlate:SAVe:MINimum

Set →

**:TEMPlate:TOLerance** → Query

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 →

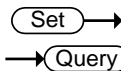
Description Saves the AUTO template (maximum and minimum templates).

Syntax :TEMPlate:SAVe:AUTO

## Mask Commands

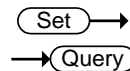
:MASK:STATe.....	250
:MASK:RATio .....	250
:MASK:SOURce .....	251
:MASK:VIOLation .....	251
:MASK:VIOLation:SAVe.....	251
:MASK:AUTo.....	252
:MASK:AUTo:SOURce .....	252
:MASK:AUTo:UNITs .....	252
:MASK:USER:UNITs .....	253
:MASK:USER:AREa<x1>:POINT<x2>.....	253
:MASK:USER:AREa<x1>:POINT<x2>:STATe .....	254
:MASK:USER:CREATE .....	254
:MASK:USER:SAVe .....	255
:MASK:USER:LOAD .....	255

### :MASK:STATe



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:RATio

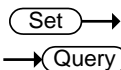


Description	Resets the mask violation ratio.
-------------	----------------------------------

Syntax :MASK:RATio {RESET}  
:MASK:RATio?

Parameter RESET Reset

Example :MASK:RATio RESET  
Resets the ratio.



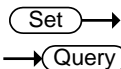
**:MASK:SOURce**

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.



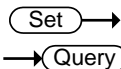
**:MASK:VIOLation**

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**

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 fastly 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 Set →  
→ Query

Description Sets or returns the units for auto created mask.

Syntax	:MASK:AUTO:UNITs {DIVIions CURRent} :MASK:AUTO:UNITs?	
Parameter	DIVIions	Divisions units.
	CURRent	Current units.
Example	:MASK:AUTO:UNITs DIVIions 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 {DIVIions CURRent} :MASK:USER:UNITs?	
Parameter	DIVIions	Divisions units.
	CURRent	Current units.
Example	:MASK:USER:UNITs DIVIions 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** (Set) →  
→ (Query)

**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?

<b>Parameter</b>	<b>AREa&lt;x1&gt;</b>	Number of the customized mask. <x1>:1~8.
	<b>POINT&lt;x2&gt;</b>	Number of the points set up the mask area. <x2>:1~10.
	<b>ON</b>	Enable the point in the mask area.
	<b>OFF</b>	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:CREATE** (Set) →

**Description** Create or removes the customized mask.

**Syntax** :MASK:USER:CREATE {ON|OFF}

<b>Parameter</b>	<b>ON</b>	Creates the mask.
	<b>OFF</b>	Removes the mask.

**Example** :MASK:USER:CREATE ON  
Create a costumized mask.

**:MASK:USER:SAVe**

Set →

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**

Set →

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.

## AWG Commands

The Arbitrary Wave Generator is available on the MDO-2000EX only.

:AWG:UTIL.....	257
:AWG:UTIL:AMPcpl .....	257
:AWG:UTIL:FREQcpl.....	258
:AWG:UTIL:FREQcpl:OFFSet .....	258
:AWG:UTIL:FREQcpl:RATio.....	259
:AWG:UTIL:TRACking.....	259
:AWG<x>:AMPlitude.....	259
:AWG<x>:FREQuency.....	260
:AWG<x>:FUNCTion .....	260
:AWG<x>:OFFSet .....	261
:AWG<x>:OUTPut:LOAd:IMPEDance .....	261
:AWG<x>:OUTPut:STATE .....	261
:AWG<x>:PHAsE.....	262
:AWG<x>:PULSe:DUTYcycle .....	262
:AWG<x>:RAMP:SYMmetry .....	262
:AWG<x>:MODulation:STATE .....	263
:AWG<x>:MODulation:TYPe.....	263
:AWG<x>:MODulation:AM:DEPth .....	263
:AWG<x>:MODulation:AM:FREQ.....	264
:AWG<x>:MODulation:AM:SHApe.....	264
:AWG<x>:MODulation:AM:PHAsE .....	264
:AWG<x>:MODulation:AM:DUTYcycle .....	265
:AWG<x>:MODulation:AM:SYMmetry .....	265
:AWG<x>:MODulation:AM:RATE .....	265
:AWG<x>:MODulation:FM:DEV .....	266
:AWG<x>:MODulation:FM:FREQ .....	266
:AWG<x>:MODulation:FM:SHApe .....	267
:AWG<x>:MODulation:FM:PHAsE .....	267
:AWG<x>:MODulation:FM:DUTYcycle.....	268
:AWG<x>:MODulation:FM:SYMmetry .....	268
:AWG<x>:MODulation:FM:RATE.....	268



:AWG<x>:MODulation:FSK:FREQ ..... 269  
 :AWG<x>:MODulation:FSK:RATE ..... 269  
 :AWG<x>:SWEep:STATE ..... 270  
 :AWG<x>:SWEep:TYPe ..... 270  
 :AWG<x>:SWEep:START ..... 270  
 :AWG<x>:SWEep:STOP ..... 271  
 :AWG<x>:SWEep:TIme ..... 271  
 :AWG<x>:SWEep:SPAN ..... 271  
 :AWG<x>:SWEep:CENTer ..... 272  
 :AWG<x>:ARBitrary:EDIT:NUMPoint ..... 272  
 :AWG<x>:ARBitrary:EDIT:FUNCTion ..... 272  
 :AWG<x>:ARBitrary:SAVe:WAVEform ..... 273  
 :AWG<x>:ARBitrary:LOAd:WAVEform ..... 273  
 :AWG<x>:ARBitrary:EDIT:COpy ..... 274  
 :AWG<x>:ARBitrary:EDIT:CLear ..... 274  
 :AWG<x>:ARBitrary:EDIT:LINE ..... 275  
 :AWG<x>:ARBitrary:EDIT:SCALE ..... 275  
 :AWG<x>:ARBitrary:EDIT:POINt ..... 275  
 :AWG<x>:ARBitrary:EDIT:POINt:ADD ..... 276  
 :AWG<x>:ARBitrary:EDIT:POINt:DELEte ..... 276

**: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

**:AWG:UTIL:AMPCpl** Set →  
→ 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 →

→ Query

**:AWG:UTIL:FREQCpl**


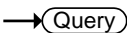
Description	Set or return the type of frequency couple.	
Syntax	:AWG:UTIL:FREQCpl {OFF  OFFSet  RATio} :AWG:UTIL:FREQCpl?	
Parameter	OFF	Turn off the Frequency Couple.
	OFFSet	The frequency of Gen. 1 and Gen. 2 are coupled by offset.
	RATio	The frequency of Gen. 1 and Gen. 2 are coupled by ratio.
Example	:AWG:UTIL:FREQCpl RATio :AWG:UTIL:FREQCpl? >RATIO	

Set →

→ Query

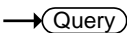
**:AWG:UTIL:FREQCpl:OFFSet**

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

**: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	<code>&lt;NRf&gt;</code>	Amplitude in Volts. (50Ω impedance 0.1~2.5V) (High Z impedance 0.2~5V)
------------------	--------------------------	--

Example :AWG1:AMP 1

**:AWG<x>:FREQuency**

Set →  
→ Query

Description Sets or returns the waveform frequency.

Syntax :AWG<x>:FREQuency {<NRf> | ?}

Parameter/ Return parameter	<code>&lt;x&gt;</code> <code>&lt;NRf&gt;</code>	Channel number 1~2. Frequency in Hertz.
--------------------------------	--	--

Example :AWG1:FREQ 2000

**:AWG<x>:FUNction**

Set →  
→ Query

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/ Return parameter	<code>&lt;x&gt;</code>	Channel number 1~2.
	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/ Return parameter	<x>	Channel number 1~2
	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.

:AWG<x>:OUTPut:STATE (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 →

**:AWG<x>:PHase**

→ 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 →

**:AWG<x>:PULSe:DUTYcycle**

→ 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/	<x>	Channel number 1~2.
Return parameter	<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.

Set →

:AWG<x>:MODulation:AM:DEPth

→ 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?  
>1.2000e+02

:AWG<x>:MODulation:AM:FREQ Set →  
→ Query

Description Sets or returns the AM modulation frequency.

Syntax :AWG<x>:MODulation:AM:FREQ {<NRF> | ?}

Parameter/	<x>	Channel number 1~2.
Return parameter	<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/	<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:AM:SHA RAMP  
Sets a ramp shape to the AM modulating waveform.

:AWG<x>:MODulation:AM:PHase Set →  
→ 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 →

→ Query

**:AWG<x>:MODulation:AM:DUTYcycle**

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 →

→ Query

**:AWG<x>:MODulation:AM:SYMMetry**

Description	Sets or returns the symmetry of the AM modulation (ramp wave shape only).	
Syntax	:AWG<x>:MODulation:AM:SYMMetry {<NRf>   ?}	
Parameter/ Return parameter	<x>	Channel number 1~2.
	<NRf>	Symmetry in percentage 0~100%.
Example	:AWG1:MOD:AM:SYM 50  Sets the symmetry of the AM modulating waveform to 50%.	

Set →

→ Query

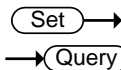
**:AWG<x>:MODulation:AM:RATE**

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/ Return parameter	<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:AM:RATE RATE5K  
Sets the noise rate of the AM modulating waveform to 5kHz.



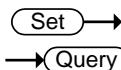
:AWG<x>:MODulation:FM:DEV

Description Sets or returns the deviation of the FM modulation.

Syntax :AWG<x>:MODulation:FM:DEV {<NRF> | ?}

Parameter/ Return parameter	<x>	Channel number 1~2.
	<NRF>	Frequency deviation in Hertz.

Example :AWG1:MOD:FM:DEV?  
>2.000000000e+02



:AWG<x>:MODulation:FM:FREQ

Description Sets or returns the frequency of the FM modulation.

Syntax	:AWG<x>:MODulation:FM:FREQ {<NRf>   ?}	
Parameter/ Return parameter	<x>	Channel number 1~2.
	<NRf>	Frequency in Hertz.
Example	:AWG1:MOD:FM:FREQ 1000 Sets the frequency of the FM modulating waveform to 1kHz.	

Set →  
→ Query

**:AWG<x>:MODulation:FM:SHApe**

Description	Sets or returns the shape of the FM modulation.	
Syntax	:AWG<x>:MODulation:FM: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: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>   ?}	
Parameter/ Return parameter	<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>   ?}	
Parameter/ Return parameter	<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>   ?}	
Parameter/ Return parameter	<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 →  
 → Query

**:AWG<x>:MODulation:FM:RATE**

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   ?}	
Parameter/ Return parameter	<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 →  
 → Query

**:AWG<x>:MODulation:FSK:FREQ**

**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.

Set →  
 → Query

**:AWG<x>:MODulation:FSK:RATE**

**Description**       Sets or returns the FSK modulation rate.

**Syntax**            :AWG<x>:MODulation:FSK:RATE {<NRF> | ?}

<b>Parameter/ Return parameter</b>	<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.	

Set →  
 → Query

**:AWG<x>:SWEep:STOP**

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.	

Set →  
 → Query

**:AWG<x>:SWEep:TIME**

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.	

Set →  
 → Query

**: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>   ?}	
Parameter/ Return parameter	<x>	Channel number 1~2.
	<NRF>	Span of the sweep in Hertz.
Example	:AWG1:SWE:SPAN 1100 Sets the span of the sweep to 1.1kHz.	

:AWG<x>:SWEep:CENTer (Set) →  
→ (Query)

**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>	<x>	Channel number 1~2.
	<NRF>	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 (Set) →  
→ (Query)

**Description**      Sets or returns the number of points of an arbitrary waveform.

**Syntax**            :AWG<x>:ARBITrary:EDIT:NUMPoint { <NR1> | ?}

<b>Parameter/ Return parameter</b>	<x>	Channel number 1~2.
	<NR1>	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}

<b>Parameter</b>	<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** 

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. Example: "Disk:/xxx.UAW" "USB:/xxx.UAW"

Example :AWG1:ARB:SAVE:WAVE ARB2  
Saves the arbitrary waveform to ARB2.

**:AWG<x>:ARBitrary:LOAd:WAVEform** 

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.

<code>&lt;file path&gt;</code>	Loads the arbitrary waveform from disk or USB at the specified file path. Exemple: "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>}`

<b>Parameter</b>	<code>&lt;x&gt;</code>	Channel number 1~2.
	<code>&lt;STARt&gt;</code>	NR1, point at which the segment to copy starts.
	<code>&lt;LENGth&gt;</code>	NR1, length of the segment to copy.
	<code>&lt;PASTe&gt;</code>	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>}`

<b>Parameter</b>	<code>&lt;x&gt;</code>	Channel number 1~2.
	ALL	Deletes the entire arbitrary waveform.

<code>&lt;STARTt&gt;</code>	NR1, point at which the segment to delete starts.
<code>&lt;LENGTh&gt;</code>	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	<code>&lt;x&gt;</code>	Channel number 1~2.
	<code>&lt;address1&gt;</code>	NR1, the point at which the line starts.
	<code>&lt;data1&gt;</code>	NRf, the value at the starting point.
	<code>&lt;address2&gt;</code>	NR1, the point at which the line ends.
	<code>&lt;data2&gt;</code>	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	<code>&lt;x&gt;</code>	Channel number 1~2.
	<code>&lt;NRf&gt;</code>	Scale 0.1~ 10

Example :AWG1:ARB:EDIT:SCAL 5.5

**:AWG<x>:ARbitrary:EDIT:POINT Set →**

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 Set →**

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 Set →**

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

:DATALOG:STATE .....	277
:DATALOG:SOURce .....	277
:DATALOG:SAVe .....	278
:DATALOG:INTerval .....	278
:DATALOG:DURation .....	279

**:DATALOG:STATE** 



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.	

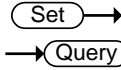
**:DATALOG:SOURce** 



Description	Sets or queries the data logging source channel.	
Syntax	:DATALOG:SOURce { CH1~CH4   D0~D15   all   ? }	
Related commands	:DATALOG:STATE :DATALOG:SAVe :DATALOG:INTerval :DATALOG:DURation	
Parameter/Return	CH1 ~CH4	Channel 1, 2, 3 or 4

parameter	D0~D15	Digital channels D0~D15
	all	All displayed channels.

Example :DATALOG:SOURce CH1  
Sets the source to CH1.



**:DATALOG:SAVE**

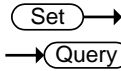
Description Sets or queries the save format as image or waveform.

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.



**:DATALOG:INTerval**

Description Sets or queries the interval time between each recording.

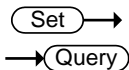
Syntax :DATALOG:INTerval <NRF>  
:DATALOG:INTerval?

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.....280  
 :REMOTEDisk:PATHName.....280  
 :REMOTEDisk:USERName .....280  
 :REMOTEDisk:PASSWord .....281  
 :REMOTEDisk:MOUNT .....281  
 :REMOTEDisk:AUTOMount .....282

:REMOTEDisk:IPADDRESS (Set) →  
→ (Query)

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 (Set) →  
→ (Query)

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 (Set) →  
→ (Query)

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.	

:REMOTEDisk:PASSWord 

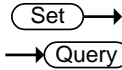


Description	Sets or queries the account password for the remote disk.	
Syntax	:REMOTEDisk:PASSWord {<string>   ? }	
Parameter/Return parameter	<string>	Username password enclosed in double quotes eg., "Password".
Example	:REMOTEDisk:PASSWord "Password" Sets the account password as Password.	

: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:IPADDDress "172.16.5.154" :REMOTEDisk:PATHName "remote_disk" :REMOTEDisk:USERName "guest" :REMOTEDisk:PASSWord "password" :REMOTEDisk:MOUNT ON Sets the remote disk parameters and mounts the remote disk.	



**:REMOTEDisk:AUTOMount**

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

## DMM Commands

:DMM.....	283
:DMM:STATE.....	283
:DMM:VALue.....	284
:DMM:HOLD.....	284
:DMM:MMIN.....	284
:DMM:MODE.....	285
:DMM:MODE:RANGe.....	285
:DMM:TEMPerature:UNITs.....	286
:DMM:TEMPerature:TYPe.....	286
:DMM:TEMPerature:SIM.....	287

**: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	ON	Turns the DMM function on.

parameter      OFF      Turns the DMM function off.

Example      :DMM:STATE ON  
 Turns the DMM function on.

:DMM:VALue      

Description      Returns the measurement value.

Syntax      :DMM:VALue?

Related commands      :MEASUrement:DISPlay

Return parameter      <string>      Returns the measurement or value on the display as a string.

Example      :DMM:VALue?  
 0.000  
 Returns the value on the DMM display.



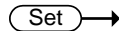
:DMM:HOLD      

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 (Set) →  
→ (Query)

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.

:DMM:MODE:RANGe (Set) →  
→ (Query)

Description Sets or queries the DMM measurement range.

Syntax :DMM:MODE:RANGe (AUto|<NRF>)  
:DMM:MODE:RANGe?

Related commands :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 | ? }

Parameter/Return parameter	Celsius	Degrees Celsius
	Fahrenheit	Degrees Fahrenheit

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

Set →

:DMM:TEMPerature:TYPE

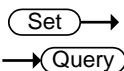
→ Query

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

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

Parameter/Return parameter	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.



**:DMM:TEMPerature:SIM**

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

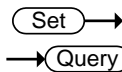
Parameter/Return parameter <NRf> 0.0~50.0 for Celsius degrees;  
 32~ 122.0 for Fahrenheit degrees

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

## Spectrum Analyzer Commands

:SA:STATE .....	288
:SA:LIST .....	289
:SA:MEMory .....	289
:SA:MEMory:SOURce .....	291
:SA:SOURce .....	292
:SA:SPECTRUMTrace .....	292
:SElect:NORMal .....	292
:SElect:MAXHold .....	293
:SElect:MINHold .....	293
:SElect:AVErage .....	294
:SA:AVErage:NUMAVg .....	294
:SA:DETECTionmethod:MODE .....	294
:SA:DETECTionmethod:MAXHold .....	295
:SA:DETECTionmethod:MINHold .....	295
:SA:DETECTionmethod:NORMal .....	296
:SA:DETECTionmethod:AVErage .....	296
:SA:FREQuency .....	297
:SA:SPAN .....	297
:SA:START .....	298
:SA:STOP .....	298
:SA:RBW:MODE .....	298
:SA:RBW .....	299
:SA:SPANRbwratio .....	299
:SA:WINDow .....	300
:SA:UNIts .....	300
:SA:SCAle .....	301
:SA:POSition .....	301

: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** → Query

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** → Query

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 by raw waveform block data.
	<waveform block data>	<string> Returns the spectrum analyzer settings . Format: parameter(1),setting(1);parameter(2),setting(2)...parameter(n),setting(n);Wa

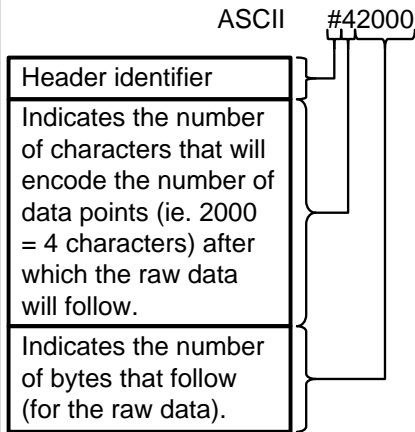
veform Data;

<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.....

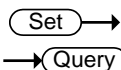
The actual value of a data point can

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



**: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 →

:SA:SOURce

→ Query

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:SPECTRUMTrace

Set →

Description Resets all spectrum traces.

Syntax SA:SPECTRUMTrace {RESET}

Parameter RESET Reset the trace

Example :SA:SPECTRUMTrace RESET  
 Reset all the trace of spectrum analyzer.

Set →

:SElect:NORMal

→ Query

Description Sets or returns the frequency domain Normal trace display on or off in the frequency domain graticule.

Syntax :SElect:NORMal {ON|OFF}  
 :SElect:NORMal?

Parameter/Return parameter	ON	Turns the normal trace display on.
	OFF	Turns the normal trace display off.

Example :SElect:NORMAL ON  
Sets the normal trace display on.

Set →

:SElect:MAXHold

→ Query

Description Sets or returns the frequency domain Max Hold trace display on or off in the frequency domain graticule.

Syntax :SElect:MAXHold {ON|OFF}  
:SElect:MAXHold?

Parameter/Return parameter	ON	Turns the Max Hold trace display on.
	OFF	Turns the Max Hold trace display off.

Example :SElect:MAXHold OFF  
Sets the Max Hold trace display off.

Set →

:SElect:MINHold

→ Query

Description Sets or returns the frequency domain Min Hold trace display on or off in the frequency domain graticule.

Syntax :SElect:MINHold {ON|OFF}  
:SElect:MINHold?

Parameter/Return parameter	ON	Turns the Min Hold trace display on.
	OFF	Turns the Min Hold trace display off.

Example : SElect:MINHold OFF  
Sets the Min Hold trace display off.

Set →  
→ Query

**:SElect:AVErage**

Description	Sets or returns the frequency domain Average trace display on or off in the frequency domain graticule.	
Syntax	:SElect:AVErage {ON OFF} :SElect: AVErage?	
Parameter/Return parameter	ON	Turns the Average trace display on.
	OFF	Turns the Average trace display off.
Example	: SElect: AVErage ON Sets the Average trace display on.	

Set →  
→ Query

**:SA:AVErage:NUMAVg**

Description	Sets or returns the number of acquisitions to be used when creating the Average frequency domain trace.	
Syntax	:SA:AVErage:NUMAVg {<NR1>} :SA:AVErage:NUMAVg?	
Parameter/Return parameter	<NR1>	The range is 2 – 256, in exponential increments.
Example	:SA:AVErage:NUMAVg 128 Sets the Average number to 128.	

Set →  
→ Query

**:SA:DETECTionmethod:MODE**

Description	Sets or returns the detection within the oscilloscope occurs automatically or manually.	
Syntax	:SA:DETECTionmethod:MODE {AUTO MANual} :SA:DETECTionmethod:MODE?	

Related commands :SA:DETECTIonmethod:MAXHold,:SA:DETECTIonmethod:MINHold  
:SA:DETECTIonmethod:NORMAl,:SA:DETECTIonmethod:AVERage

Parameter/Return parameter	AUTo	Automatically mode
	MANual	Manually mode

Example :SA:DETECTIonmethod:MODE AUTO  
Sets the detection mode to automatic.

Set →  
 → Query

**:SA:DETECTIonmethod:MAXHold**

Description Sets or returns the detection method of max Hold frequency domain trace.

Syntax :SA:DETECTIonmethod:MAXHold {PLUSpeak|MINUSpeak|SAMPLE|AVERage}  
:SA: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.

Example :SA:DETECTIonmethod:MAXHold AVERage  
Sets the detection method to average.

Set →  
 → Query

**:SA:DETECTIonmethod:MINHold**

Description Sets or returns the detection method of min Hold frequency domain trace.

Syntax :SA:DETECTIonmethod:MINHold {PLUSpeak|MINUSpeak|SAMPLE|AVERage}  
:SA: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.

Example :SA:DETECTionmethod:MINHold AVErage  
Sets the detection method to average.

Set →

:SA:DETECTionmethod:NORMal

→ Query

Description Sets or returns the detection method of normal frequency domain trace.

Syntax :SA:DETECTionmethod:NORMal  
{PLUSpeak|MINUSpeak|SAMple|AVErage}  
:SA: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.

Example :SA:DETECTionmethod:NORMal AVErage  
Sets the detection method to average.

Set →

:SA:DETECTionmethod:AVErage

→ Query

Description Sets or returns the detection method of average frequency domain trace.

Syntax :SA:DETECTionmethod:AVErage  
{PLUSpeak|MINUSpeak|SAMple|AVErage}  
:SA: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.

Example :SA:DETECTionmethod:AVErage AVErage  
Sets the detection method to average.

Set →

:SA:FREQUENCY

→ Query

Description Sets or returns the frequency (or center frequency) of the acquisition system.

Syntax :SA:FREQUENCY {<NRF>|CENTER}  
:SA:FREQUENCY?

Parameter/Return parameter	<NRF>	Sets the frequency by user.
	CENTER	Sets the frequency to center.

Example SA:FREQUENCY 3.0E+06  
Sets the center frequency to 3 MHz.

Set →

:SA:SPAN

→ Query

Description Sets or returns the span frequency setting.

Syntax :SA:SPAN <NRF>  
:SA:SPAN?

Parameter/Return parameter	<NRF>	Sets the span frequency by user.
----------------------------	-------	----------------------------------

Example SA:SPAN 25E+06  
Sets the span frequency to 25 MH.

Set →  
 → Query

**:SA:START**

Description	Sets or returns the start frequency setting.	
Syntax	:SA:START <NRf> :SA:START?	
Parameter/Return parameter	<NRf>	Sets the start frequency by user.
Example	SA:START -9.5E+06 Sets the start frequency to -9.5 MHz.	

Set →  
 → Query

**:SA:STOP**

Description	Sets or returns the stop frequency setting.	
Syntax	:SA:STOP <NRf> :SA:STOP?	
Parameter/Return parameter	<NRf>	Sets the stop frequency by user.
Example	SA:START 100E+06 Sets the stop frequency to 100MHz.	

Set →  
 → Query

**:SA:RBW:MODE**

Description	Sets or returns the resolution bandwidth (RBW) mode, either automatic or manual.	
Syntax	:SA:RBW:MODE {AUTO MANual} :SA:RBW:MODE?	
Parameter/Return parameter	AUTO	Automatically mode.
	MANual	Manually mode.
Example	SA:RBW:MODE AUTO Sets the mode to automatic.	

Set →  
 → Query

**:SA:RBW**

---

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:RBW <NRf> :SA:RBW?	
Related commands	SA:RBW:MODE	
Parameter/Return parameter	<NRf>	Sets the RBW by user.
Example	Sets SA:RBW 2.0E+04 Query SA:RBW? Return 1.825017e+04 If the RBW set to 20kHz, the query will return the nearest value (1.825017e+04).	

Set →  
 → Query

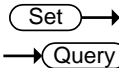
**:SA:SPANRbwratio**

---

Description	Sets or returns the resolution bandwidth (RBW) when the RBW mode has been set to AUTO (using the command SA:RBW:MODE).	
Syntax	:SA:SPANRbwratio {RATIO1K RATIO2K RATIO5K RATIO10K RATIO20K RATIO50K RATIO100K RATIO200K <NRf>} :SA:SPANRbwratio?	
Related commands	SA:RBW:MODE	
Parameter/Return parameter	<NRf>	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

**Example** :SA:SPANRbwratio RATIO2K  
 Sets the ratio to 2000:1.  
 Sets :SA:SPANRbwratio 2000  
 Query :SA:SPANRbwratio?  
 Return RATIO2K



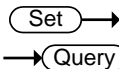
**:SA:WINDow**

**Description** Sets or returns the windowing function, which is only used for traces.

**Syntax** :SA:WINDow {RECTangular|HAMming|HANning|BLACKman}:SA: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

**Example** :SA:WINDow HANning  
 Sets to the hanning window.



**:SA:UNIts**

**Description** Sets or returns the vertical units.

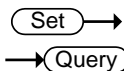
**Syntax** :SA:UNIts {DBV|LINEAR|DBM}  
 :SA:UNIts?

<b>Parameter/ Return parameter</b>	DBV	Sets to DBV unit
	LINEAR	Sets to Linear unit

**DBM** Sets to DBM unit

Example :SA:UNIts DBM  
Sets the unit to DBM unit.

**:SA:SCALE**



Description Sets or returns the overall vertical scale.

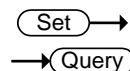
Syntax :SA:SCALE <NRf>  
:SA:SCALE?

Related commands :SA: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)

Example :SA:SCALE 2  
Sets the scale to 2.

**:SA:POSition**



Description Sets or returns the overall vertical position.

Syntax :SA:POSition <NRf>  
:SA:POSition?

Parameter/ Return parameter <NRf> Vertical position range: +/-12

Example :SA:POSition 3  
Sets the vertical position to 3.



## Power Supply Commands

:POWERSupply:OUTPut<X> .....	302
:POWERSupply:CONFigure .....	302
:POWERSupply:OUTPut<X>:VOLTage .....	303
:POWERSupply:OUTPut<X>:RECONFigure .....	303
:POWERSupply:OUTPut<X>:OCP .....	303

:POWERSupply:OUTPut<X> 


Description	Sets or returns the power supply output.	
Syntax	:POWERSupply:OUTPut<X> {ON OFF} :POWERSupply:OUTPut<X>?	
Parameter/ Return parameter	OFF	Turns off the power supply output
	ON	Turns on the power supply output
	<X>	Range 1~2, Select the output 1 or output 2
Example	:POWERSupply:OUTPut1 ON Turn on output 1.	

:POWERSupply:CONFigure 

  


Description	Configur the power supply. *It is must delay about 6 sec after power supply config.	
Syntax	:POWERSupply:CONFigure {ON ?}	
Parameter	ON	Configur the power supply.
Example	:POWERSupply:CONFigure 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>	Range 1.0~5.0, sets the voltage
	<X>	Range 1~2, Select the output 1 or output 2
Example	:POWERSupply:OUTPut1:VOLTage 3.3 Sets the power supply output 1 to 3.3V.	

:POWERSupply:OUTPut<X>:RECONFigure (Set) →

Description	Reconfigure the power supply when OCP occurred.	
Syntax	:POWERSupply:OUTPut<X>:RECONFigure {ON}	
Parameter/ Return parameter	ON	Reconfigure the power supply
	<X>	Range 1~2, Select the output 1 or output 2
Example	:POWERSupply:OUTPut1:RECONFigure ON Reconfigure the power supply output 1.	

:POWERSupply:OUTPut<X>:OCP → (Query)

Description	Returns the power supply OCP.	
Syntax	:POWERSupply:OUTPut<X>:OCP?	
Parameter/ Return parameter	<X>	Range 1~2, Select the output 1 or output 2
Example	:POWERSupply:OUTPut1:OCP? Return the OCP status for output 1.	

# 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.	



## Digital Commands

:D<x>:DISPlay .....	305
:D<x>:POSition .....	305
:DISPlay:DIGital:HEIght .....	306
:DIGital:GROU<x>:THReshold .....	306
:DIGital:ANALog:A<x>:DISPlay .....	307
:DIGital:ANALog:A<x>:RATio .....	307
:DIGital:ANALog:A<x>:POSition .....	308
:D<x>:MEMory .....	308
:DIGital:MEMory .....	310

:D<x>:DISPlay (Set) →  
→ (Query)

**Description** Turns a digital channel <x> on/off or returns its status.

**Syntax** :D<x>:DISPlay {OFF | ON | ?}

<b>Parameter / Return parameter</b>	OFF	Turns off a digital channel
	ON	Turns on a digital channel
	D<x>	Digital channel number D0 ~ D15

**Example** :D0:DISP OFF

:D<x>:POSition (Set) →  
→ (Query)

**Description** Sets or returns the position level for digital channel <x>.

**Syntax** :D<x>:POSition {<NRf> | ?}

<b>Parameter / Return parameter</b>	D<x>	Digital channel number D0 ~ D15
	<NRf>	Vertical scale position

```
Example      :D0:POS ?
              >1.87 DIV
              :D0:POS 0
```

Set →  
 → Query

**:DISPlay:DIGital:HEIght**

Description	Sets or returns the number of available digital waveform position slots.	
Syntax	:DISPlay:DIGital:HEIght {SMALL   MEDium   LARge   ?}	
Parameter / Return parameter	SMALL	Sets the height to small mode (digital channels: 16 max)
	MEDium	Sets the height to medium mode (digital channels: 16 max)
	LARge	Sets the height to large mode (digital channels: 8 max)

```
Example      :DIS:DIG:HEI ?
              >LARGE
              :DIS:DIG:HEI SMA
```

Set →  
 → Query

**:DIGital:GROU<x>:THReshold**

Description	Sets or returns the digital threshold for a group.	
Syntax	:DIGital:GROU<x> {ECL   TTL   PECL   CMOS5   CMOS3   CMOS2   <NR3>   ?}	
Parameter / Return parameter	ECL	Sets the threshold to a preset ECL high level (-1.3V)
	TTL	Sets the threshold to a preset TTL high level (1.4V)
	PECL	Sets the threshold to a preset PECL high level (3.7V)
	CMOS5	Sets the threshold to a preset CMOS5 (5.0V) high level (2.5V)

CMOS3	Sets the threshold to a preset CMOS3 (3.3V) high level (1.65V)
CMOS2	Sets the threshold to a preset CMOS2 (2.5V) high level (1.25V)
<NR3>	Sets the threshold to a preset ECL high level (-1.3V)
GROUP<x>	Group number 1~4 (16 channels) or 1~2 (8 channels) GROUP1: digital channels D0~D3 GROUP2: digital channels D4~D7 GROUP3: digital channels D8~D11 GROUP4: digital channels D12~D15

Example :DIG:GROUP1:THR ?  
>-1.300e+00  
:DIG:GROUP1:THR TTL

Set →  
→ Query

**:DIGital:ANALog:A<x>:DISPlay**

Description Turns the analog waveform <x> on/off or returns its status.

Syntax :DIGital:ANALog:A<x>:DISPlay {OFF | ON | ?}

Parameter / Return parameter	OFF	Turns off the analog waveform
	ON	Turns on the analog waveform
	A<x>	Analog waveform number 1~2

Example :DIG:ANA:A1:DISP OFF

Set →  
→ Query

**:DIGital:ANALog:A<x>:RATio**

Description Sets or returns the analog waveform vertical scale ratio.

Syntax :DIGital:ANALog:A<x>:RATio {<NRf> | ?}

Parameter / Return parameter	<NRf> A<x>	Vertical scale ratio (0.1, 0.2, ...1). Analog waveform number 1~2
------------------------------	---------------	--

Example :DIG:ANA:A1:RAT 0.1

:DIGital:ANALog:A<x>:POSition Set →  
→ Query

Description	Sets or returns the analog waveform vertical scale position.	
-------------	--	--

Syntax :DIGital:ANALog:A<x>:POSition {<NRf> | ?}

Parameter / Return parameter	<NRf> A<x>	Vertical scale position (0, 0.1, 0.2, ...8). Analog waveform number 1~2
------------------------------	---------------	--

Example :DIG:ANA:A1:POS 4.5

:D<x>:MEMory → Query

Description	Returns the data in acquisition memory for the selected digital channel.	
-------------	--	--

Syntax :D<x>:MEMory?

Related commands	:ACQuire:RECOrdlength :HEADer	
------------------	----------------------------------	--

Parameter	D<x>	Digital channel number D0 ~ D15
-----------	------	---------------------------------

Return parameter

Returns acquisition settings followed by raw waveform block data.

<string>

<string>

Returns the acquisition settings for the selected digital channel.

Format:

parameter(1),setting(1);parameter(2),setting(2)...parameter(n),setting(n);  
Waveform Data;

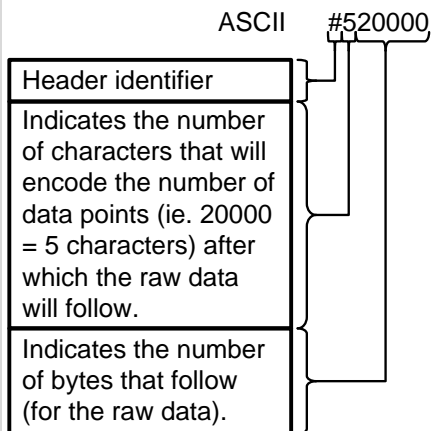
<waveform block data>

<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 logical level (0 or 1) of a data point of the digital channel, i.e. logical 0 is 0x0000 and logical 1 is 0x0001.

Waveform Raw Data Example:

Header raw data.....

Hex:

35 32 30 30 30 30 30 00 00 00 00 00 00  
00 01 00 01 .....

ASCII/Decimal:

#520000 0000000101 ....

The raw data contains 20000 bytes (=10000 points); point 1 is logical 0, point 2 is logical 0, point 3 is logical 0, point 4 is logical 1, point 5 is logical 1, etc...

Example :D1:MEM?  
 FORMAT,2.0A;Display,1;Memory  
 Length,10000;IntpDistance,0; Trigger Address,0;  
 Threshold Used,1.400E+00;Source,D1;Vertical  
 Units,V; Label1,;Firmware,V1.25b10; Horizontal  
 Units,S;Horizontal Scale,1.000E-04; Horizontal  
 Position,0.000E+00;Horizontal Mode,Main;SincET  
 Mode,Real Time; Sampling Period,1.000E-07;Time,22-  
 Sep-16 19:42:28;  
 Waveform Data;  
 #520000.....follows waveform block  
 data.....

**:DIGital:MEMory**

→ Query

Description	Returns the data in acquisition memory for all the digital channels.
Syntax	:DIGital:MEMory?
Related commands	:ACQuire:RECOrdlength :HEADer

Return parameter	Returns acquisition settings followed by raw waveform block data.				
<string>	<p>&lt;string&gt;</p> <p>Returns the acquisition settings for all the digital channels.</p> <p>Format: parameter(1),setting(1);parameter(2),setting(2)...parameter(n),setting(n); Waveform Data;</p>				
<waveform block data>	<p>&lt;waveform block data&gt;</p> <p>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>				
	<div style="text-align: right; margin-right: 20px;">ASCII #520000</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Header identifier</td> <td rowspan="3" style="text-align: center; vertical-align: middle;">}                 <div style="border-left: 1px solid black; border-right: 1px solid black; height: 100px; margin: 0 auto;"></div> </td> </tr> <tr> <td style="padding: 5px;">Indicates the number of characters that will encode the number of data points (ie. 20000 = 5 characters) after which the raw data will follow.</td> </tr> <tr> <td style="padding: 5px;">Indicates the number of bytes that follow (for the raw data).</td> </tr> </table>	Header identifier	} <div style="border-left: 1px solid black; border-right: 1px solid black; height: 100px; margin: 0 auto;"></div>	Indicates the number of characters that will encode the number of data points (ie. 20000 = 5 characters) after which the raw data will follow.	Indicates the number of bytes that follow (for the raw data).
Header identifier	} <div style="border-left: 1px solid black; border-right: 1px solid black; height: 100px; margin: 0 auto;"></div>				
Indicates the number of characters that will encode the number of data points (ie. 20000 = 5 characters) after which the raw data will follow.					
Indicates the number of bytes that follow (for the raw data).					
	<p>Raw Data: The sixteen bits composing each consecutive two bytes encode the logical level (0 or 1) of all the digital channels for one data point. For a given two bytes, the least significant</p>				

bit is channel 0 and the most significant bit is channel 15.

Waveform Raw Data Example:

Header raw data.....

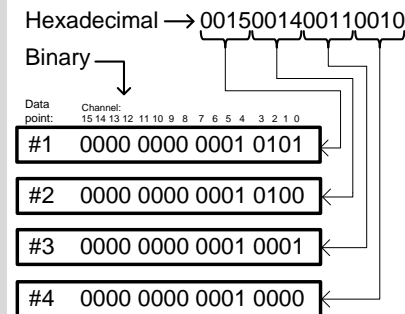
Hex:

35 32 30 30 30 30 00 15 00 14 00 11  
00 10 .....

ASCII/Decimal:

#520000 raw data ....

The raw data contains 20000 bytes (=10000 points) with the following logical level for each channels per data point:



Example

```
:DIG:MEM?
Format,2.0E;Display,1111111111111111;Memory
Length,10000;IntpDistance,0;Trigger
Address,0;Threshold12_15,1.40V;Threshold8_11,1.40
V;Threshold4_7,1.40V;Threshold0_3,1.40V;Vertical
Units,V;Label15,;Label14,;Label13,;Label12,;Label11,;L
abel10,;Label9,;Label8,;Label7,;Label6,;Label5,;Label4,
;Label3,;Label2,;Label1,;Label0,;Firmware,V1.25b10;H
orizontal Units,S;Horizontal Scale,1.000E-
04;Horizontal Position,0.000E+00;Horizontal
Mode,Main;SincET Mode,Real Time;Sampling
Period,1.000E-07;Time,22-Sep-16 10:44:28;
Waveform Data;
#520000.....follows waveform block
data.....
```



# APPENDIX

## Error messages

Description      The following error messages may be returned from the :SYSTem:ERRor? query. For details see page 167.

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"

# INDEX

ACQuire	
FILTer	
TRACking .....	48
ACQuire	
AVERAge .....	42
FILTer	
FREQUency .....	46
FILTer.....	46
SOURce .....	45
MEMory.....	43
MODE.....	43
ACQuire	
STATe .....	48
ACQuire	
INTERpolation.....	48
ACQuire	
RECOrdlength .....	49
ACQuire	
HEADer .....	49
AUTORSET	
MODE.....	50
AUTOSet.....	50
BUS1 .....	175
CAN	
BITRate.....	186
PROBe .....	186
SAMPLepoint.....	186
SOURce .....	185
DISplay	
FORMAt.....	185
I2C	
ADDResS	
RWINCLude.....	176
SCLK	
SOURce.....	177
SDA	
SOURce.....	177, 178
INPut .....	176
LABEl.....	228, 307
DISPlay .....	228, 229, 230, 231, 307, 310
LIN	
BITRate .....	187
IDFORmat .....	187
POLARity .....	188
SAMPLepoint.....	188
SOURce.....	188
STANDard .....	189
SPI	
BITORder .....	183
MISO	
SOURce .....	184
MOSI	
SOURce .....	184
SCLK	
POLARity .....	182
SOURce .....	183
SS	
POLARity .....	182
SOURce .....	183
WORDSize .....	182
STATE .....	175
TYPe .....	175
UART	
BITRate .....	179
DATABits .....	179
EOFPacket.....	180
PACKet .....	180
PARity .....	179
RX	
SOURce .....	181
TX	
SOURce .....	181
CHANnel	
BWLimit.....	51
COUPling.....	52
DESKew .....	52
DISPlay.....	52
EXPand.....	53
IMPedance .....	53
INVert.....	54

LABel .....	225, 305	DATE .....	173
DISPlay .....	226, 305	DISPlay	
POSition .....	54	INTENSITY	
PROBe		BACKLight	
TYPE .....	55	AUTODim	
PROBe		Time .....	78
RATio .....	55	DISPlay	
SCALe .....	55	GRATicule .....	79
CLS .....	38	INTensity	
CURSor		BACKLight .....	78
DDT .....	69	BACKLight	
H1Position .....	70	AUTODim	
H2Position .....	70	ENABle .....	78
HDELta .....	71	GRATicule .....	77
HUNI .....	68	WAVEform .....	77
HUSE .....	68	OUTPut .....	80
MODE .....	67	PERSistence .....	79
SOURce .....	67	WAVEform .....	80
V1Position .....	71	DMM	
VDELta .....	72	HOLD .....	283, 284
VUNI .....	69	MOD .....	285
VUSE .....	69	TEMPerature	
XY		TYPe .....	286, 287, 291, 292, 293, 294
POLar		UNITs .....	286
RADIUS		VALue .....	284, 285
DELta .....	74	DVM	
POSition .....	73	MODE .....	243
THETA		SOURce .....	242
DELta .....	74	STATE .....	242
POSition .....	74	VALue .....	243
PRODuct		Error list .....	313
DELta .....	75	Ethernet	
POSition .....	75	interface .....	7
RATio		ETHERnet	
DELta .....	76	DHCP .....	173
RATio		FORCe .....	121
POSition .....	75	GONogo	
RECTangular		CLEar .....	244
X		EXECute .....	244, 250, 251, 252, 257, 258, 259
DELta .....	72	FUNCTION .....	245, 252, 260
POSition .....	72	NGCount .....	245, 250, 253, 254, 255, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273,
Y			
DELta .....	73		
POSition .....	73		
DATALOG			
DURation ...	279, 294, 295, 296, 297, 298, 299		
INTERval .....	278		
SAVE .....	278		
SOURce .....	277		
STATE .....	277		

274, 275, 299, 300, 301, 302, 303, 304	AMPLitude .....	96
NGDefine.....	AREa.....	102
245	CARea .....	102
SCRIpt.....	CMEan.....	97
246	CRMS.....	101
SOURce.....	FALL.....	88
246	FFFDelay.....	105
VIOLation.....	FFRDelay.....	104
	FOVShoot.....	88
HARDcopy	FPReshoot.....	89
ASSIGN.....	FREQuency .....	89
83	FRFDelay.....	103
MODE.....	FRRDelay.....	103
81	GATing.....	85
PRINTINKSaver .....	HIGH .....	98
82	LFFDelay.....	107
SAVEFORMat .....	LFRDelay .....	106
82	LOW .....	98
SAVEINKSaver.....	LRFDelay .....	106
82	LRRDelay.....	105
START.....	MAX .....	99
81	MEAN .....	97
HEADER .....	METHod .....	86
49	MIN .....	99
IDN? .....	NEDGE.....	95
36	NPULSE .....	94
Interface .....	NWIDth.....	90
5	PDUTy.....	90
LISTer	PEDGE.....	95
DATA.....	PERiod.....	91
185	PHAsE.....	108
LRN?.....	PK2PK .....	100
36	PPULSE .....	94
MARK .....	PWIDth.....	91
190	RISe .....	92
CREATE.....	RMS .....	100
190	ROVShoot .....	93
DELEte.....	RPReshoot.....	93
191	SOURce .....	85
MATH		
ADVanced	MEASurement	
OPERator .....	MEAS	
62	MAXimum .....	113
POSITION .....	MEAN.....	113
64	MINImum .....	114
SCALE.....	SOURCE .....	110
64	STATE.....	111
DISP.....	STDdev .....	115
57	TYPe.....	111
DUAL	VALue.....	112
OPERator .....	REFLevel	
58		
POSITION .....		
59		
SCALE.....		
59		
SOURce .....		
58		
FFT		
HORizontal		
POSITION.....		
62		
SCALE.....		
62		
MAG .....		
60		
POSITION .....		
61		
SCALE.....		
61		
SOURce .....		
60		
WINDow.....		
60		
TYPE.....		
58		
MATHVAR.....		
63		
MEASure		

PERCent	SINFo .....	223
HIGH .....	STATE .....	194
LOW .....	TOTAL .....	194
MID .....	TRIGger	
MID2 .....	BUS	
STATistics .....	B1	
MODE .....	CAN	
WEIghting .....	CONDition .....	212
RCL .....	DATA	
RECALL	QUALifier .....	215
SETUp .....	SIZE .....	216
WAVEform .....	VALue .....	217
REF	FRAMEtype .....	213
DISPlay .....	IDentifier	
LABel .....	DIRection .....	215
DISPlay .....	MODE .....	214
OFFSet .....	VALue .....	214
SCALe .....	I2C	
TIMEbase	ADDRESS	
POSition .....	DIRection .....	205
SCALe .....	MODE .....	203
Remote control	TYPE .....	203
interface configuration .....	VALue .....	204
.....5	CONDition .....	202
REMOVEDisk	DATA	
AUTOMount .....	SIZE .....	205
IPADDRESS .....	VALue .....	206
MOUNT .....	LIN	
PASSWord .....	CONDition .....	218
PATHName .....	DATA	
USERName .....	QUALifier .....	218
RST .....	SIZE .....	219
RUN .....	VALue .....	220
SAV .....	ERRRYPE .....	220
SAVe	IDentifier	
IMAGe .....	VALue .....	221
FILEFormat .....	SPI	
INKSaver .....	CONDition .....	210
SETUp .....	DATA	
WAVEform .....	MISO .....	211
FILEFormat .....	MOSI .....	212
SEARCH	SIZE .....	210
COPY .....	UART	
FFTPeak	CONDition .....	206
METHod .....	RX	
MPEak .....	DATA .....	207, 208
	TX	
	DATA .....	209
	TYPE .....	201
	EDGE	
	SLOP .....	195
	HLEVel .....	196
	LEVel .....	196
	LLEVel .....	197
	PULSe	
	Time .....	199



WHEn .....	198	MAXimum .....	247
PULSEWidth		MINimum .....	247
POLarity .....	197	MODE .....	247
RISEFall		POSITION	
SLOP .....	198	MAXimum .....	248
TIME .....	201	MINimum .....	248
WHEn .....	200	SAVE	
RUNT		AUTo .....	249
POLarity .....	197	MAXimum .....	248
TIME .....	200	MINimum .....	249
WHEn .....	199	TOLerance.....	249
SOURce .....	195	TIMEbase	
TYPe .....	194	EXPand.....	122
SEGMENTS		MODE .....	123
CURRent.....	234	POSITION .....	122
DISPALL.....	235	SCALE.....	122
MEASure		WINDow	
MODE .....	235	POSITION.....	123
PLOT		SCALE.....	124
DIVide.....	236	TRIGGER	
RESults.....	237	ALTErnate.....	140
SElect .....	237	BUS	
SOURce.....	236	B1	
TABLE		CAN	
LIST .....	239	CONDition .....	153
SAVe .....	239	DATA	
SElect .....	238	QUALifier.....	156
SOURce.....	238	SIZE .....	157
SAVE .....	240	VALue.....	157
SElect		FRAMetype .....	154
END .....	241	IDentifier	
STARt.....	240	DIRection .....	155
SOURce .....	240	MODE.....	154
STATE .....	233	VALue.....	155
Time .....	235	I2C	
TOTalnum .....	234	ADDRess	
SET		DIRection .....	146
LABel .....	232, 308	MODE.....	144
SINGLE .....	120	TYPe.....	144
Socket server		VALue.....	145
function check.....	10	CONDition .....	143
Socket server		DATA	
interface .....	9	SIZE .....	146
STOP .....	120	VALue.....	147
SYSTEM		LIN	
ERRor .....	167	CONDition .....	158
LOCK .....	167	DATA	
TEMPLATE		QUALifier.....	159
		SIZE .....	159
		VALue.....	160
		ERRTYPE....	161, 162, 163, 164,
		165	
		IDentifier	

VALue.....	161, 162	HLEVel.....	130
SPI		HOLDoff.....	129
CONDition.....	151	LEVel.....	130
DATA		LLEVel.....	131
MISO		MODE.....	129
VALue.....	152	NREJ.....	129
MOSI		PULSE	
VALue.....	152	TIME.....	139
SIZE.....	151	WHEn.....	138
UART		PULSEWidth	
CONDition.....	147	POLarity.....	133
RX		RISEFall	
DATA		SLOP.....	135
SIZE.....	148	TIME.....	136
VALue.....	149	WHEn.....	136
TX		RUNT	
DATA		POLarity.....	134
SIZE.....	149	TIME.....	135
VALue.....	150	WHEn.....	134
THReshold		SOURce.....	128
CH.....	143	STATe.....	141
TYPE.....	142	TIMEOut	
COUPLE.....	128	TIMER.....	140
DELay		WHEn.....	139
EVENT.....	133	TYPE.....	127
LEVel.....	133	VIDeo	
SLOP.....	132	FIELD.....	137
TIME.....	132	LINE.....	138
TYPE.....	132	POLarity.....	138
EDGE		TYPe.....	137
SLOP.....	131	USB	
EXTERnal		function check.....	6
PRObe		remote control interface.....	5
RATio.....	142		
TYPe.....	141		
FREQuency.....	127		