RIGOL

Programming Guide

DS1000Z-E Series
Digital Oscilloscope

Aug. 2019
RIGOL (SUZHOU) TECHNOLOGIES INC.

Guaranty and Declaration

Copyright

Trademark Information

RIGOL is a registered trademark of **RIGOL** (SUZHOU) TECHNOLOGIES INC.

Publication Number

PGA27100-1110

Software Version

00.06.01

Software upgrade might change or add product features. Please acquire the latest version of the manual from **RIGOL** website or contact **RIGOL** to upgrade the software.

Notices

- **RIGOL** products are covered by P.R.C. and foreign patents, issued and pending.
- **RIGOL** reserves the right to modify or change parts of or all the specifications and pricing policies at the company's sole decision.
- Information in this publication replaces all previously released materials.
- Information in this publication is subject to change without notice.
- **RIGOL** shall not be liable for either incidental or consequential losses in connection with the furnishing, use, or performance of this manual, as well as any information contained.
- Any part of this document is forbidden to be copied, photocopied, or rearranged without prior written approval of **RIGOL**.

Product Certification

RIGOL guarantees that this product conforms to the national and industrial standards in China as well as the ISO9001:2015 standard and the ISO14001:2015 standard. Other international standard conformance certifications are in progress.

Contact Us

If you have any problem or requirement when using our products or this manual, please contact **RIGOL**.

E-mail: service@rigol.com
Website: www.rigol.com

Document Overview

This manual provides guidance on how to use the SCPI commands in programming to realize remote control of **RIGOL** DS1000Z-E series digital oscilloscope through the remote interface. DS1000Z-E can communicate with a PC through the USB or the LAN bus.

Main Topics in this Manual:

Chapter 1 Programming Overview
This chapter introduces how to build the remote communication between DS1000Z-E series digital oscilloscope and the PC. It also introduces the remote control methods as well as the syntax, symbols, parameters and abbreviation rules of the SCPI commands.

Chapter 2 Command System
This chapter introduces the syntax, function, parameter and using instruction of each command.

Chapter 3 Programming Demos
This chapter lists some programming demos to illustrate how to use commands to realize the common functions of the oscilloscope in the development environments of Excel, Matlab, LabVIEW, Visual Basic 6.0 and Visual C++ 6.0.

Tip
For the newest version of this manual, please download it from **RIGOL** official website (www.rigol.com).

Format Conventions in this Manual:

1. **Key**
   The function key on the front panel is denoted by the format of "Key Name (Bold) + Text Box" in the manual. For example, **Utility** denotes the "Utility" key on the front panel.

2. **Menu**
   The menu item is denoted by the format of "Menu Word (Bold) + Character Shading" in the manual. For example, **System** denotes the "System" item under **Utility**.

3. **Operation Step**
   The next step of the operation is denoted by an arrow "→" in the manual. For example, **Utility** → **System** denotes that first press **Utility** on the front panel and then press **System**.

Content Conventions in this Manual:

DS1000Z-E series includes the following models. Unless otherwise noted, this manual takes DS1202Z-E as an example to illustrate the command system of DS1000Z-E series.

Model	Analog Bandwidth	Number of Analog Channels
DS1202Z-E	200 MHz	2

Contents

Chapter 1 Programming Overview

This chapter introduces how to build the remote communication between DS1000Z-E series digital oscilloscope and the PC. It also introduces the remote control methods as well as the syntax, symbols, parameters and abbreviation rules of the SCPI commands.

Main topics of this chapter:

To Build Remote Communication

This oscilloscope can communicate with a PC through the USB or the LAN bus. This section introduces how to control the oscilloscope remotely through the USB interface using Ultra Sigma in details.

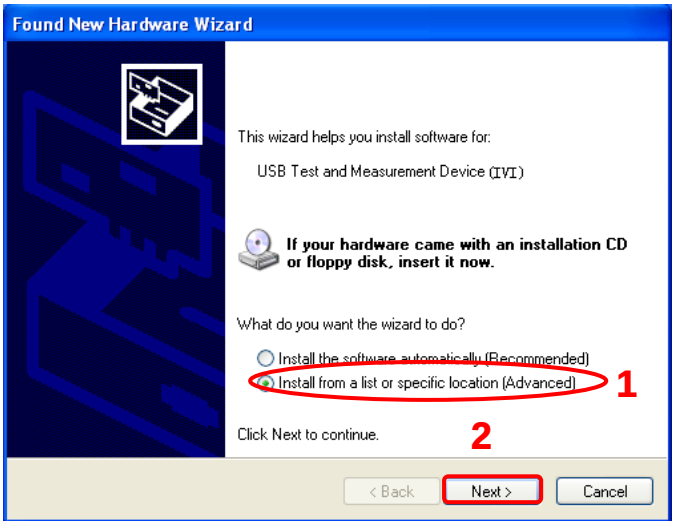
Operation Steps:

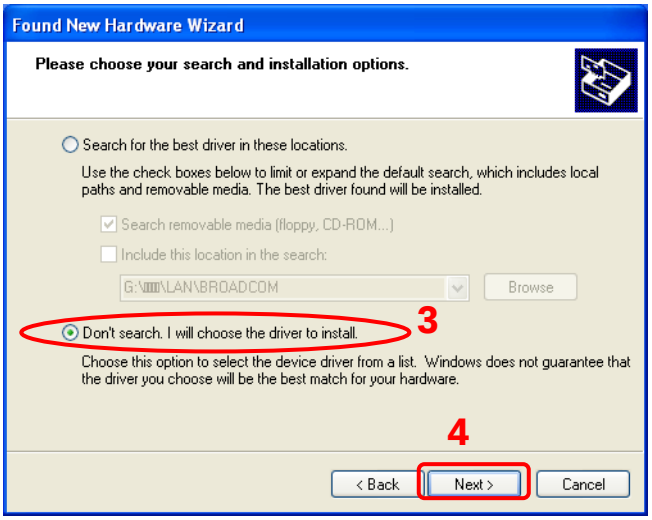
1. **Install the Ultra Sigma common PC software**
   Download the Ultra Sigma common PC software from **RIGOL** official website (www.rigol.com) and install it according to the instructions.

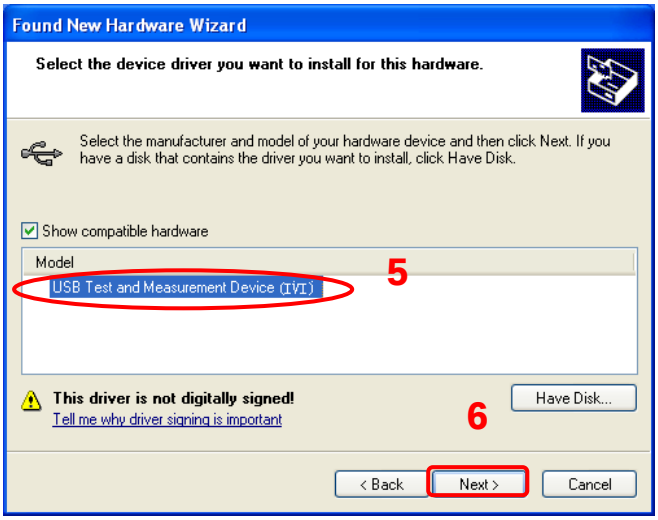
2. **Connect the instrument and PC and configure the interface parameters of the instrument**
   DS1000Z-E can communicate with a PC through the USB or the LAN bus. This manual takes the USB interface as an example.

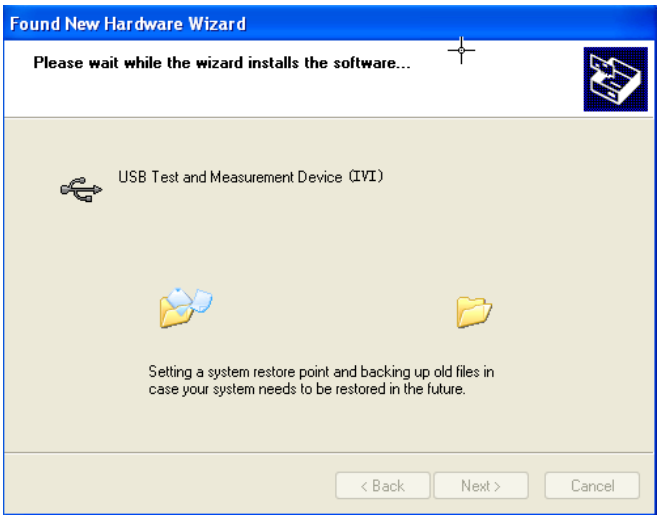
   (1) Connect the devices
   Connect the USB Device interface at the real panel of the oscilloscope and the USB Host interface of the PC using a USB cable.

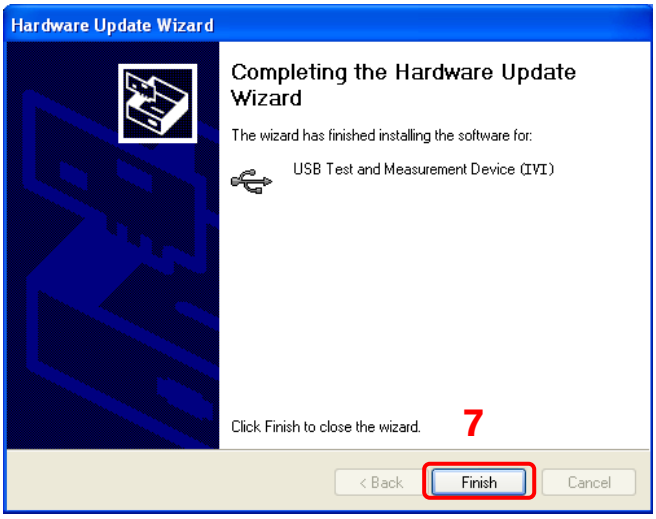
   (2) Install the USB driver
   This oscilloscope is a USB-TMC device. After you connect the oscilloscope to the PC and turn both on for the first time (the oscilloscope is automatically configured to the USB interface; at the same time, make sure that **Utility** → **IO Setting** → **USB Device** is set to "Computer"), the **Found New Hardware Wizard** as shown in the figure below is displayed on the PC. Please install the "USB Test and Measurement Device (IVI)" driver following the directions in the wizard. The steps are as follows.




Found New Hardware Wizard
Please choose your search and installation options.
Search for the best driver in these locations.
Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.
Search removable media (floppy, CD-ROM...)
Include this location in the search:
G:\LAN\BROADCOM
Browse
3
Don't search. I will choose the driver to install.
Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.
4
< Back
Next >
Cancel


Found New Hardware Wizard
Select the device driver you want to install for this hardware.
Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.
Show compatible hardware
Model
USB Test and Measurement Device (IVI)
5
This driver is not digitally signed!
Tell me why driver signing is important
Have Disk...
6
< Back
Next >
Cancel


Found New Hardware Wizard
Please wait while the wizard installs the software...
USB Test and Measurement Device (IVI)
Setting a system restore point and backing up old files in case your system needs to be restored in the future.
< Back
Next >
Cancel



(3) Search for device resource

Start up the **Ultra Sigma** and the software will automatically search for the instrument resources currently connected to the PC. You can also click USB-TMC to search for the resources.

(4) View the device resources

The resources found will appear under the "RIGOL Online Resource" directory and the model number and USB interface information of the instrument will also be displayed.
For example, DS1202Z-E (USB0::0x1AB1::0x04CE::DS1ZD170800001::INSTR).

(5) Control the instrument remotely

Right click the resource name "DS1202Z-E (USB0::0x1AB1::0x04CE::DS1ZD170800001::INSTR)" and select "SCPI Panel Control" to turn on the remote command control panel through which you can send commands and read data.

Remote Control Methods

1. User-defined Programming

Users can use SCPI (Standard Commands for Programmable Instruments) commands to program and control the oscilloscope. For details, refer to the introductions in "**Chapter 3 Programming Demos**".

2. Send SCPI Commands via PC Software

You can control the oscilloscope remotely by sending SCPI commands via PC software. Ultra Sigma provided by **RIGOL** is recommended.

SCPI Command Overview

SCPI (Standard Commands for Programmable Instruments) is a standardized instrument programming language that is built upon the standard IEEE 488.1 and IEEE 488.2 and conforms to various standards (such as the floating point operation rule in IEEE 754 standard, ISO 646 7-bit coded character for information interchange (equivalent to ASCII programming)). The SCPI commands provide a hierarchical tree structure and consist of multiple subsystems. Each command subsystem consists of a root keyword and one or more sub-keywords.

Syntax

The command string usually starts with ":"; the keywords are separated by ":" and are followed by the parameter settings available; "?" is added at the end of the command string to indicate query; the command keywords and the first parameter are separated by space.

For example,
:ACQuire:TYPE <type>
:ACQuire:TYPE?
ACQuire is the root keyword of the command. TYPE is the second-level keyword. The command string starts with ":" which is also used to separate the multiple-level keywords. <type> represents the parameters available for setting. "?" represents query. The command keywords :ACQuire:TYPE and parameter <type> are separated by a space.

"," is generally used for separating multiple parameters contained in the same command, for example,
[:TRACe[<n>]]:DATA:VALue volatile,<points>,<data>

Symbol Description

The following symbols will not be sent with the commands.

1. **Braces { }**
   The parameters enclosed in the braces are optional and are usually separated by the vertical bar "|". When using the command, one of the parameters must be selected.

2. **Vertical Bar |**
   The vertical bar is used to separate multiple parameters and one of the parameters must be selected when using the command.

3. **Square Brackets []**
   The content in the square brackets can be omitted.

4. **Triangle Brackets < >**
   The parameter enclosed in the triangle brackets must be replaced by an effective value.

Parameter Type

1. **Bool**
   The parameter could be ON, OFF, 1, or 0. For example,
   :MEASure:ADISplay <bool>
   :MEASure:ADISplay?
   Wherein,
   <bool> can be set to {{1|ON}|{0|OFF}}.
   The query returns 1 or 0.

2. Discrete

The parameter could be any of the values listed. For example,
:ACQuire:TYPE <type>
:ACQuire:TYPE?
Wherein,
<type> can be set to NORMal|AVERages|PEAK|HRESolution.
The query returns the abbreviations (NORM, AVER, PEAK, or HRES).

3. Integer

Unless otherwise noted, the parameter can be any integer (NR1 format) within the effective value range. Note that do not set the parameter to a decimal, otherwise errors will occur. For example,
:DISPlay:GBRightness <brightness>
:DISPlay:GBRightness?
Wherein,
<brightness> can be set to any integer between 0 and 100.
The query returns an integer between 0 and 100.

4. Real

The parameter can be any real number within the effective value range and this command accepts decimal (NR2 format) and scientific notation (NR3 format) parameter input. For example,
:TRIGger:TIMeout:TIMe <NR3>
:TRIGger:TIMeout:TIMe?
Wherein,
<NR3> can be set to any real number between 1.6e-08 (namely 16ns) to 1e+01 (namely 10s).
The query returns a real number in scientific notation.

5. ASCII String

The parameter should be the combinations of ASCII characters.
For example,
:SYSTem:OPTion:INSTall <license>
Wherein,
<license> can be set to PDUY9N9QTS9PQSWPLAETRD3UJHYA.

Command Abbreviation

All the commands are case-insensitive and you can use any of them. If abbreviation is used, all the capital letters in the command must be written completely. For example,
:MEASure:ADISplay? can be abbreviated to :MEAS:ADIS?.

Chapter 2 Command System

This chapter introduces the syntax, function, parameter, and using instruction of each DS1000Z-E command.

Main topics of this chapter:

Note:

1. Unless otherwise noted, this manual takes DS1202Z-E as an example to introduce the commands.
2. For parameter setting commands (for example, the time, frequency, and amplitude), the oscilloscope can only accept numbers and set the parameters using the default units; it cannot recognize the units sent with the parameters. For the default unit of each parameter, please refer to the description in each command in the following introductions.

:AUToscale

Syntax :AUToscale

Description Enable the waveform auto setting function. The oscilloscope will automatically adjust the vertical scale, horizontal timebase, and trigger mode according to the input signal to realize optimum waveform display. This command is equivalent to pressing the **AUTO** key on the front panel.

Explanation

- Theoretically, waveform auto setting function requires that the frequency of sine is no lower than 41Hz; the duty cycle should be greater than 1% and the amplitude must be at least 20mVpp for square (the probe ratio is 1X).
- When the pass/fail function is enabled (see the :MASK:ENABle command), if you sent this command, the oscilloscope will disable the pass/fail function firstly and then execute the waveform auto setting function.
- When the waveform record function is enabled or during the playback of the recorded waveform, this command is invalid.

:CLEar

Syntax :CLEar

Description Clear all the waveforms on the screen. If the oscilloscope is in the RUN state, waveform will still be displayed. This command is equivalent to pressing the **CLEAR** key on the front panel.

Related Command :DISPlay:CLEar

:RUN
:STOP

Syntax :RUN

:STOP

Description The :RUN command starts the oscilloscope and the :STOP command stops the oscilloscope. These commands are equivalent to pressing the **RUN/STOP** key on the front panel.

Explanation When the waveform record function is enabled or during the playback of the recorded waveform, these commands are invalid.

:SINGle

Syntax :SINGle

Description Set the oscilloscope to the single trigger mode. This command is equivalent to any of the following two operations: pressing the **SINGLE** key on the front panel and sending the :TRIGger:SWEep SINGle command.

Explanation

- In the single trigger mode, the oscilloscope triggers once when the trigger conditions are met and then stops.
- When the waveform record function is enabled or during the playback of the recorded waveform, this command is invalid.

Related Commands :TFORce

:RUN

:STOP

:TFORce

Syntax :TFORce

Description Generate a trigger signal forcefully. This command is only applicable to the normal and single trigger modes (see the :TRIGger:SWEep command) and is equivalent to pressing the **FORCE** key in the trigger control area on the front panel.

:ACQuire Commands

The :ACQuire commands are used to set and query the memory depth, acquisition mode and the number of averages as well as query the current sample rate of the oscilloscope.

Command List [1]:

- :ACQuire:AVERages
- :ACQuire:MDEPth
- :ACQuire:TYPE
- :ACQuire:SRATe?

Note[1]: In the "Command List" in this manual, the parameters in the setting commands and the query commands are not included and you can refer to the complete introductions of the commands in the text according to the keywords.

:ACQuire:AVERages

Syntax :ACQuire:AVERages <count>

:ACQuire:AVERages?

Description Set or query the number of averages under the average acquisition mode.

Parameter

Name	Type	Range	Default
<count>	Integer	2^n (n is an integer from 1 to 10)	2

Explanation

- You can send the :ACQuire:TYPE command to set the acquisition mode.
- In the average acquisition mode, greater number of averages can lower the noise and increase the vertical resolution, but will also slow the response of the displayed waveform to the waveform changes.

Return Format The query returns an integer between 2 and 1024.

Example

```
:ACQuire:AVERages 128      /*Set the number of averages to 128*/
:ACQuire:AVERages?         /*The query returns 128*/
```

:ACQuire:MDEPth

Syntax :ACQuire:MDEPth <mdep>

:ACQuire:MDEPth?

Description Set or query the memory depth of the oscilloscope (namely the number of waveform points that can be stored in a single trigger sample). The default unit is pts (points).

Parameter

Name	Type	Range	Default
<mdep>	Discrete	Refer to **Explanation**	AUTO

Explanation

- When a single channel is enabled, the range of <mdep> is {AUTO|12000|120000|1200000|12000000|24000000}.
  When dual channels are enabled, the range of <mdep> is {AUTO|6000|60000|600000|6000000|12000000}.
- The following equation describes the relationship among memory depth, sample rate, and waveform length:
  Memory Depth = Sample Rate x Waveform Length
  Wherein, the Waveform Length is the product of the horizontal timebase (set by the :TIMebase[:MAIN]:SCALe command) times the number of grids in the horizontal direction on the screen (12 for DS1000Z-E).
  When AUTO is selected, the oscilloscope will select the memory depth automatically according to the current sample rate.

Return Format The query returns the actual number of points (integer) or AUTO.

Example

```
:ACQuire:MDEPth 12000      /*Set the memory depth to 12000pts*/
:ACQuire:MDEPth?           /*The query returns 12000*/
```

Related Command :ACQuire:SRATe?

:ACQuire:TYPE

Syntax :ACQuire:TYPE <type>

:ACQuire:TYPE?

Description Set or query the acquisition mode of the oscilloscope.

Parameter

Name	Type	Range	Default
<type>	Discrete	{NORMal|AVERages|PEAK|HRESolution}	NORMal

Explanation

- NORMal: in this mode, the oscilloscope samples the signal at equal time interval to rebuild the waveform. For most of the waveforms, the best display effect can be obtained using this mode.
- AVERages: in this mode, the oscilloscope averages the waveforms from multiple samples to reduce the random noise of the input signal and improve the vertical resolution. The number of averages can be set by the:ACQuire:AVERages command. Greater number of averages can lower the noise and increase the vertical resolution, but will also slow the response of the displayed waveform to the waveform changes.
- PEAK (Peak Detect): in this mode, the oscilloscope acquires the maximum and minimum values of the signal within the sample interval to get the envelope of the signal or the narrow pulse of the signal that might be lost. In this mode, signal confusion can be prevented but the noise displayed would be larger.
- HRESolution (High Resolution): this mode uses a kind of ultra-sample technique to

average the neighboring points of the sample waveform to reduce the random noise on the input signal and generate much smoother waveforms on the screen. This is generally used when the sample rate of the digital converter is higher than the storage rate of the acquisition memory.

Return Format The query returns NORM, AVER, PEAK, or HRES.

Example
```
:ACQuire:TYPE AVERages      /*Select the average acquisition mode*/
:ACQuire:TYPE?              /*The query returns AVER*/
```

:ACQuire:SRATe?

Syntax :ACQuire:SRATe?

Description Query the current sample rate. The default unit is Sa/s.

Explanation

- Sample rate is the sample frequency of the oscilloscope, namely the waveform points sampled per second.
- The following equation describes the relationship among memory depth, sample rate, and waveform length:

  Memory Depth = Sample Rate x Waveform Length

  Wherein, the Memory Depth can be set using the :ACQuire:MDEPth command, and the Waveform Length is the product of the horizontal timebase (set by the :TIMebase[:MAIN]:SCALe command) times the number of the horizontal scales (12 for DS1000Z-E).

Return Format The query returns the sample rate in scientific notation.

Example
```
:ACQuire:SRATe?     /*The query returns 2.000000e+09*/
```

:CALibrate Commands

Command List:

- :CALibrate:QUIT
- :CALibrate:STARt

:CALibrate:QUIT

Syntax :CALibrate:QUIT

Description Exit the self-calibration at any time.

Related Command :CALibrate:STARt

:CALibrate:STARt

Syntax :CALibrate:STARt

Description The oscilloscope starts to execute self-calibration.

Explanation

- The self-calibration operation can make the oscilloscope quickly reach its optimum working state to obtain the most accurate measurement values.
- During the self-calibration, all the channels of the oscilloscope must be disconnected from the inputs.
- The functions of most of the keys are disabled during the self-calibration. You can send the :CALibrate:QUIT command to quit the self-calibration.

:CHANnel<n> Commands

The :CHANnel<n> commands are used to set or query the vertical system parameters of the analog channels, such as the bandwidth limit, coupling, vertical scale, and vertical offset.

Command List:

:CHANnel<n>:BWLimit

Syntax :CHANnel<n>:BWLimit <type>

:CHANnel<n>:BWLimit?

Description Set or query the bandwidth limit parameter of the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<type>	Discrete	{20M|OFF}	OFF

Explanation

- OFF: disable the bandwidth limit and the high frequency components of the signal under test can pass the channel.
  20M: enable the bandwidth limit and the high frequency components of the signal under test that exceed 20 MHz are attenuated.
- Enabling the bandwidth limit can reduce the noise, but can also attenuate the high frequency components.

Return Format The query returns 20M or OFF.

Example

```
:CHANnel1:BWLimit 20M      /*Enable the 20MHz bandwidth limit*/
:CHANnel1:BWLimit?         /*The query returns 20M*/
```

:CHANnel<n>:COUPling

Syntax :CHANnel<n>:COUPling <coupling>

:CHANnel<n>:COUPling?

Description Set or query the coupling mode of the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<coupling>	Discrete	{AC|DC|GND}	DC

Explanation

- AC: the DC components of the signal under test are blocked.
- DC: the DC and AC components of the signal under test can both pass the channel.
- GND: the DC and AC components of the signal under test are both blocked.

Return Format The query returns AC, DC, or GND.

Example

```
:CHANnel1:COUPling AC      /*Select the AC coupling mode*/
:CHANnel1:COUPling?        /*The query returns AC*/
```

:CHANnel<n>:DISPlay

Syntax :CHANnel<n>:DISPlay <bool>

:CHANnel<n>:DISPlay?

Description Enable or disable the specified channel or query the status of the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	CH1: 1|ON CH2: 0|OFF

Return Format The query returns 1 or 0.

Example

```
:CHANnel1:DISPlay ON       /*Enable CH1*/
:CHANnel1:DISPlay?         /*The query returns 1*/
```

:CHANnel<n>:INVert

Syntax :CHANnel<n>:INVert <bool>

:CHANnel<n>:INVert?

Description Enable or disable the waveform invert of the specified channel or query the status of the waveform invert of the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation When waveform invert is turned off, the waveform display is normal; when waveform invert is turned on, the waveform voltage values are inverted.

Return Format The query returns 1 or 0.

Example
```
:CHANnel1:INVert ON      /*Enable the waveform invert of CH1*/
:CHANnel1:INVert?        /*The query returns 1*/
```

:CHANnel<n>:OFFSet

Syntax :CHANnel<n>:OFFSet <offset>

:CHANnel<n>:OFFSet?

Description Set or query the vertical offset of the specified channel. The default unit is V.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<offset>	Real	Related to the current vertical scale and probe ratio When the probe ratio is 1X, vertical scale≥500mV/div: -100V to +100V vertical scale<500mV/div: -2V to +2V When the probe ratio is 10X, vertical scale≥5V/div: -1000V to +1000V vertical scale<5V/div: -20V to +20V	0V (the probe ratio is 10X)

Return Format The query returns the vertical offset in scientific notation.

Example
```
:CHANnel1:OFFSet 0.01    /*Set the vertical offset of CH1 to 10mV*/
:CHANnel1:OFFSet?        /*The query returns 1.000000e-02*/
```

:CHANnel<n>:RANGe

Syntax :CHANnel<n>:RANGe <range>

:CHANnel<n>:RANGe?

Description Set or query the vertical range of the specified channel. The default unit is V.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<range>	Real	Related to the probe ratio When the probe ratio is 1X: 8mV to 80V When the probe ratio is 10X: 80mV to 800V	8V (the probe ratio is 10X)

Explanation This command indirectly modifies the vertical scale of the specified channel (Vertical Scale = Vertical Range/8). The vertical scale can be set by the :CHANnel<n>:SCALe command.

Return Format The query returns the vertical range in scientific notation.

Example

```
:CHANnel1:RANGe 8        /*Set the vertical range of CH1 to 8V*/
:CHANnel1:RANGe?         /*The query returns 8.000000e+00*/
```

:CHANnel<n>:TCAL

Syntax :CHANnel<n>:TCAL <val>

:CHANnel<n>:TCAL?

Description Set or query the delay calibration time of the specified channel to calibrate the zero offset of the corresponding channel. The default unit is s.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<val>	Real	-100ns to 100ns	0.00s

Explanation <val> can only be set to the specific values in the specified step. If the parameter you sent is not one of the specific values, the parameter will be set to the nearest specific values automatically. The step varies with the horizontal timebase (set by the :TIMebase[:MAIN]:SCALe command), as shown in the table below.

Horizontal Timebase	Step of the Delay Calibration Time
5ns	100ps
10ns	200ps
20ns	400ps
50ns	1ns
100ns	2ns
200ns	4ns
500ns	10ns
1μs to 10μs	20ns

Note: When the horizontal timebase is equal to or greater than 10μs, the delay calibration time cannot be adjusted.

Return Format The query returns the delay calibration time in scientific notation.

Example

```
:CHANnel1:TCAL 0.00000002      /*Set the delay calibration time to 20ns*/
:CHANnel1:TCAL?                /*The query returns 2.000000e-08*/
```

:CHANnel<n>:SCALe

Syntax :CHANnel<n>:SCALe <scale>

:CHANnel<n>:SCALe?

Description Set or query the vertical scale of the specified channel. The default unit is V.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<scale>	Real	Related to the current probe ratio When the probe ratio is 1X: 1mV to 10V When the probe ratio is 10X (default): 10mV to 100V	1V (the probe ratio is 10X)

Explanation

- The range of the vertical scale is related to the current probe ratio (set by the :CHANnel<n>:PROBe command).
- You can use the :CHANnel<n>:VERNier command to enable or disable the fine adjustment of the vertical scale. By default, the fine adjustment is off. At this point, you can only set the vertical scale in 1-2-5 step, namely 10mV, 20mV, 50mV, 100mV, ..., 100V (the probe ratio is 10X). When the fine adjustment is on, you can further adjust the vertical scale within a relatively smaller range to improve the vertical resolution. If the amplitude of the input waveform is a little bit greater than the full scale under the current scale and the amplitude would be a little bit lower if the next scale is used, fine adjustment can be used to improve the display amplitude of the waveform to view the signal details.

Return Format The query returns the vertical scale in scientific notation.

Example

```
:CHANnel1:SCALe 1     /*Set the vertical scale of CH1 to 1V*/
:CHANnel1:SCALe?      /*The query returns 1.000000e+00*/
```

:CHANnel<n>:PROBe

Syntax :CHANnel<n>:PROBe <atten>

:CHANnel<n>:PROBe?

Description Set or query the probe ratio of the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<atten>	Discrete	{0.01|0.02|0.05|0.1|0.2|0.5|1|2|5|10|20|50|100|200|500|1000}	10

Explanation

- Setting the probe ratio refers to multiply the signal sampled with the specified ratio and then display the result (the actual amplitude of the signal will not be affected).
- Setting the probe ratio will affect the range of the vertical scale.

Return Format The query returns the probe ratio in scientific notation.

Example

```
:CHANnel1:PROBe 10     /*Set the probe ratio of CH1 to 10X*/
:CHANnel1:PROBe?       /*The query returns 1.000000e+01*/
```

Related Command :CHANnel<n>:SCALe

:CHANnel<n>:UNITs

Syntax :CHANnel<n>:UNITs <units>

:CHANnel<n>:UNITs?

Description Set or query the amplitude display unit of the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<units>	Discrete	{VOLTage|WATT|AMPere|UNKNown}	VOLTage

Return Format The query returns VOLT, WATT, AMP, or UNKN.

Example

```
:CHANnel1:UNITs VOLTage      /*Set the amplitude display unit of CH1 to V*/
:CHANnel1:UNITs?             /*The query returns VOLT*/
```

:CHANnel<n>:VERNier

Syntax :CHANnel<n>:VERNier <bool>

:CHANnel<n>:VERNier?

Description Enable or disable the fine adjustment of the vertical scale of the specified channel, or query the fine adjustment status of the vertical scale of the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation By default, the fine adjustment is off. At this point, you can only set the vertical scale in 1-2-5 step, namely 10mV, 20mV, 50mV, 100mV...100V (the probe ratio is 10X). When the fine adjustment is on, you can further adjust the vertical scale within a relatively smaller range to improve the vertical resolution. If the amplitude of the input waveform is a little bit greater than the full scale under the current scale and the amplitude would be a little bit lower if the next scale is used, fine adjustment can be used to improve the display amplitude of the waveform to view the signal details.

Return Format The query returns 1 or 0.

Example

```
:CHANnel1:VERNier ON      /*Enable the fine adjustment function of the vertical scale of
                            CH1*/
:CHANnel1:VERNier?        /*The query returns 1*/
```

Related Command :CHANnel<n>:SCALe

:CURSor Commands

The :CURSor commands are used to measure the X-axis value (such as time) and Y-axis value (such as voltage) of the waveform displayed on the screen.

Command List:

:CURSor:MODE

Syntax :CURSor:MODE <mode>

:CURSor:MODE?

Description Set or query the cursor measurement mode.

Parameter

Name	Type	Range	Default
<mode>	Discrete	{OFF|MANual|TRACk|AUTO|XY}	OFF

Explanation

- OFF: disable the cursor measurement function.
- MANual: enable the manual cursor measurement mode.
- TRACk: enable the track cursor measurement mode.
- AUTO: enable the auto cursor measurement mode.
- XY: enable the XY cursor measurement mode. This mode is valid only when the horizontal timebase mode is XY.

Return Format The query returns OFF, MAN, TRAC, AUTO, or XY.

Example

```
:CURSor:MODE MANual      /*Enable the manual cursor measurement mode*/
:CURSor:MODE?            /*The query returns MAN*/
```

Related Commands

:CURSor:MANual

:CURSor:TRACk

:CURSor:XY

:TIMebase:MODE

:CURSor:MANual

Command List:

:CURSor:MANual:TYPE

Syntax :CURSor:MANual:TYPE <type>

:CURSor:MANual:TYPE?

Description Set or query the cursor type in manual cursor measurement mode.

Parameter

Name	Type	Range	Default
<type>	Discrete	{X|Y}	X

Explanation

- X: select the X type cursors. The X type cursors are a vertical solid line (cursor A) and a vertical dotted line (cursor B) and are usually used to measure the time parameters.
- Y: select the Y type cursors. The Y type cursors are a horizontal solid line (cursor A) and a horizontal dotted line (cursor B) and are usually used to measure the voltage parameters.
- When the channel source of manual cursor measurement is set to LA (:CURSor:MANual:SOURce), the cursor type cannot be set to Y.

Return Format The query returns X or Y.

Example

```
:CURSor:MANual:TYPE Y      /*select the Y type cursors*/
:CURSor:MANual:TYPE?       /*The query returns Y*/
```

:CURSor:MANual:SOURce

Syntax :CURSor:MANual:SOURce <source>

:CURSor:MANual:SOURce?

Description Set or query the channel source of the manual cursor measurement mode.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2|MATH}	CHANnel1

Explanation Only the channel that is enabled currently can be selected.

Return Format The query returns CHAN1, CHAN2, or MATH.

Example
```
:CURSor:MANual:SOURce CHANnel2      /*Set the channel source to CH2*/
:CURSor:MANual:SOURce?              /*The query returns CHAN2*/
```

:CURSor:MANual:TUNit

Syntax :CURSor:MANual:TUNit <unit>

:CURSor:MANual:TUNit?

Description Set or query the horizontal unit in the manual cursor measurement mode.

Parameter

Name	Type	Range	Default
<unit>	Discrete	{S|HZ|DEGRee|PERCent}	S

Explanation

- S: AX, BX, and BX-AX in the measurement results are in "s" and 1/|dX| is in "Hz".
- HZ: AX, BX, and BX-AX in the measurement results are in "Hz" and 1/|dX| is in "s".
- DEGRee: AX, BX, and BX-AX are in "degree".
- PERCent: AX, BX, and BX-AX are expressed in percentage.

Return Format The query returns S, HZ, DEGR, or PERC.

Example
```
:CURSor:MANual:TUNit DEGRee      /*Set the horizontal unit to "degree"*/
:CURSor:MANual:TUNit?            /*The query returns DEGR*/
```

:CURSor:MANual:VUNit

Syntax :CURSor:MANual:VUNit <unit>

:CURSor:MANual:VUNit?

Description Set or query the vertical unit in the manual cursor measurement mode.

Parameter

Name	Type	Range	Default
<unit>	Discrete	{PERCent|SOURce}	SOURce

Explanation

- PERCent: AY, BY, and BY-AY in the measurement results are expressed in percentage.
- SOURce: the units of AY, BY, and BY-AY in the measurement results will be automatically set to the unit of the current source.

Return Format The query returns PERC or SOUR.

Example
```
:CURSor:MANual:VUNit PERCent     /*Set the system to express AY, BY, and BY-AY in the
                                   measurement results in percentage*/
:CURSor:MANual:VUNit?            /*The query returns PERC*/
```

Related Command :CHANnel<n>:UNITs

:CURSor:MANual:AX

Syntax :CURSor:MANual:AX <x>

:CURSor:MANual:AX?

Description Set or query the horizontal position of cursor A in the manual cursor measurement mode.

Parameter

Name	Type	Range	Default
<x>	Integer	5 to 594	100

Explanation The horizontal and vertical positions of the cursor are defined by the pixel coordinate of the screen. The pixel coordinate of the screen ranges from (0,0) to (600,400). Wherein, (0,0) is located at the left top corner of the screen and (600,400) is located at the right bottom corner of the screen. The horizontal pixel range is from 0 to 600 and the vertical pixel range is from 0 to 400.

Return Format The query returns an integer between 5 and 594.

Example

```
:CURSor:MANual:AX 200      /*Set the horizontal position of cursor A to 200*/
:CURSor:MANual:AX?         /*The query returns 200*/
```

:CURSor:MANual:BX

Syntax :CURSor:MANual:BX <x>

:CURSor:MANual:BX?

Description Set or query the horizontal position of cursor B in the manual cursor measurement mode.

Parameter

Name	Type	Range	Default
<x>	Integer	5 to 594	500

Explanation The horizontal and vertical positions of the cursor are defined by the pixel coordinate of the screen. The pixel coordinate of the screen ranges from (0,0) to (600,400). Wherein, (0,0) is located at the left top corner of the screen and (600,400) is located at the right bottom corner of the screen. The horizontal pixel range is from 0 to 600 and the vertical pixel range is from 0 to 400.

Return Format The query returns an integer between 5 and 594.

Example

```
:CURSor:MANual:BX 200      /*Set the horizontal position of cursor B to 200*/
:CURSor:MANual:BX?         /*The query returns 200*/
```

:CURSor:MANual:AY

Syntax :CURSor:MANual:AY <y>

:CURSor:MANual:AY?

Description Set or query the vertical position of cursor A in the manual cursor measurement mode.

Parameter

Name	Type	Range	Default
<y>	Integer	5 to 394	100

Explanation The horizontal and vertical positions of the cursor are defined by the pixel coordinate of

the screen. The pixel coordinate of the screen ranges from (0,0) to (600,400). Wherein, (0,0) is located at the left top corner of the screen and (600,400) is located at the right bottom corner of the screen. The horizontal pixel range is from 0 to 600 and the vertical pixel range is from 0 to 400.

Return Format The query returns an integer between 5 and 394.

Example
```
:CURSor:MANual:AY 200      /*Set the vertical position of cursor A to 200*/
:CURSor:MANual:AY?         /*The query returns 200*/
```

:CURSor:MANual:BY

Syntax :CURSor:MANual:BY <y>

:CURSor:MANual:BY?

Description Set or query the vertical position of cursor B in the manual cursor measurement mode.

Parameter

Name	Type	Range	Default
<y>	Integer	5 to 394	300

Explanation The horizontal and vertical positions of the cursor are defined by the pixel coordinate of the screen. The pixel coordinate of the screen ranges from (0,0) to (600,400). Wherein, (0,0) is located at the left top corner of the screen and (600,400) is located at the right bottom corner of the screen. The horizontal pixel range is from 0 to 600 and the vertical pixel range is from 0 to 400.

Return Format The query returns an integer between 5 and 394.

Example
```
:CURSor:MANual:BY 200      /*Set the vertical position of cursor B to 200*/
:CURSor:MANual:BY?         /*The query returns 200*/
```

:CURSor:MANual:AXValue?

Syntax :CURSor:MANual:AXValue?

Description Query the X value of cursor A in the manual cursor measurement mode. The unit depends on the horizontal unit currently selected.

Return Format The query returns the X value of cursor A in scientific notation.

Example
```
:CURSor:MANual:AXValue?    /*The query returns -4.000000e-06*/
```

Related Commands :CURSor:MANual:AX

:CURSor:MANual:TUNit

:CURSor:MANual:AYValue?

Syntax :CURSor:MANual:AYValue?

Description Query the Y value of cursor A in the manual cursor measurement mode. The unit depends on the vertical unit currently selected.

Return Format When the signal source is CHANnel1|CHANnel2|MATH, the query returns the Y value of cursor A in scientific notation.

Example
```
:CURSor:MANual:AYValue?    /*The query returns 2.000000e+00*/
```

Related Commands :CURSor:MANual:AY

:CURSor:MANual:VUNit

:CURSor:MANual:BXValue?

Syntax :CURSor:MANual:BXValue?

Description Query the X value of cursor B in the manual cursor measurement mode. The unit depends on the horizontal unit currently selected.

Return Format The query returns the X value of cursor B in scientific notation.

Example :CURSor:MANual:BXValue? /*The query returns 4.000000e-06*/

Related Commands :CURSor:MANual:BX

:CURSor:MANual:TUNit

:CURSor:MANual:BYValue?

Syntax :CURSor:MANual:BYValue?

Description Query the Y value of cursor B in the manual cursor measurement mode. The unit depends on the vertical unit currently selected.

Return Format When the signal source is CHANnel1|CHANnel2|MATH, the query returns the Y value of cursor B in scientific notation.

Example :CURSor:MANual:BYValue? /*The query returns -2.000000e+00*/

Related Commands :CURSor:MANual:BY

:CURSor:MANual:VUNit

:CURSor:MANual:XDELta?

Syntax :CURSor:MANual:XDELta?

Description Query the difference between the X values of cursor A and cursor B (BX-AX) in the manual cursor measurement mode. The unit depends on the horizontal unit currently selected.

Return Format The query returns the difference in scientific notation.

Example :CURSor:MANual:XDELta? /*The query returns 8.000000e-06*/

Related Commands :CURSor:MANual:AX

:CURSor:MANual:BX

:CURSor:MANual:TUNit

:CURSor:MANual:IXDELta?

Syntax :CURSor:MANual:IXDELta?

Description Query the reciprocal of the absolute value of the difference between the X values of cursor A and cursor B (1/|dX|) in the manual cursor measurement mode. The unit depends on the horizontal unit currently selected.

Return Format The query returns 1/|dX| in scientific notation.

Example :CURSor:MANual:IXDELta? /*The query returns 1.250000e+05*/

Related Commands :CURSor:MANual:AX

:CURSor:MANual:BX

:CURSor:MANual:TUNit

:CURSor:MANual:YDELta?

Syntax :CURSor:MANual:YDELta?

Description Query the difference between the Y values of cursor A and cursor B (BY-AY) in the manual cursor measurement mode. The unit depends on the vertical unit currently selected.

Return Format When the signal source is CHANnel1|CHANnel2|MATH, the query returns the difference in scientific notation.

Example
```
:CURSor:MANual:YDELta?     /*The query returns -4.000000e+00*/
```

Related Commands :CURSor:MANual:AY
:CURSor:MANual:BY
:CURSor:MANual:VUNit

:CURSor:TRACk

Command List:

- :CURSor:TRACk:SOURce1
- :CURSor:TRACk:SOURce2
- :CURSor:TRACk:AX
- :CURSor:TRACk:BX
- :CURSor:TRACk:AY?
- :CURSor:TRACk:BY?
- :CURSor:TRACk:AXValue?
- :CURSor:TRACk:AYValue?
- :CURSor:TRACk:BXValue?
- :CURSor:TRACk:BYValue?
- :CURSor:TRACk:XDELta?
- :CURSor:TRACk:YDELta?
- :CURSor:TRACk:IXDELTA?

:CURSor:TRACk:SOURce1

Syntax :CURSor:TRACk:SOURce1 <source>

:CURSor:TRACk:SOURce1?

Description Set or query the channel source of cursor A in the track cursor measurement mode.

Parameter

Name	Type	Range	Default
<source>	Discrete	{OFF|CHANnel1|CHANnel2|MATH}	CHANnel1

Explanation Only the channels enabled can be selected as the channel source.

Return Format The query returns OFF, CHAN1, CHAN2, or MATH.

Example
```
:CURSor:TRACk:SOURce1 CHANnel2     /*Set the channel source to CH2*/
:CURSor:TRACk:SOURce1?             /*The query returns CHAN2*/
```

:CURSor:TRACk:SOURce2

Syntax :CURSor:TRACk:SOURce2 <source>

:CURSor:TRACk:SOURce2?

Description Set or query the channel source of cursor B in the track cursor measurement mode.

Parameter

Name	Type	Range	Default
<source>	Discrete	{OFF|CHANnel1|CHANnel2|MATH}	CHANnel1

Explanation Only the channels enabled can be selected as the channel source.

Return Format The query returns OFF, CHAN1, CHAN2, or MATH.

Example
```
:CURSor:TRACk:SOURce2 CHANnel2    /*Set the channel source to CH2*/
:CURSor:TRACk:SOURce2?            /*The query returns CHAN2*/
```

:CURSor:TRACk:AX

Syntax :CURSor:TRACk:AX <x>

:CURSor:TRACk:AX?

Description Set or query the horizontal position of cursor A in the track cursor measurement mode.

Parameter

Name	Type	Range	Default
<x>	Integer	5 to 594	100

Explanation The horizontal and vertical positions of the cursor are defined by the pixel coordinate of the screen. The pixel coordinate of the screen ranges from (0,0) to (600,400). Wherein, (0,0) is located at the left top corner of the screen and (600,400) is located at the right bottom corner of the screen. The horizontal pixel range is from 0 to 600 and the vertical pixel range is from 0 to 400.

Return Format The query returns an integer between 5 and 594.

Example
```
:CURSor:TRACk:AX 200    /*Set the horizontal position of cursor A to 200*/
:CURSor:TRACk:AX?       /*The query returns 200*/
```

:CURSor:TRACk:BX

Syntax :CURSor:TRACk:BX <x>

:CURSor:TRACk:BX?

Description Set or query the horizontal position of cursor B in the track cursor measurement mode.

Parameter

Name	Type	Range	Default
<x>	Integer	5 to 594	500

Explanation The horizontal and vertical positions of the cursor are defined by the pixel coordinate of the screen. The pixel coordinate of the screen ranges from (0,0) to (600,400). Wherein, (0,0) is located at the left top corner of the screen and (600,400) is located at the right bottom corner of the screen. The horizontal pixel range is from 0 to 600 and the vertical pixel range is from 0 to 400.

Return Format The query returns an integer between 5 and 594.

Example
```
:CURSor:TRACk:BX 200    /*Set the horizontal position of cursor B to 200*/
:CURSor:TRACk:BX?       /*The query returns 200*/
```

:CURSor:TRACk:AY?

Syntax :CURSor:TRACk:AY?

Description Query the vertical position of cursor A in the track cursor measurement mode.

Explanation
- The horizontal and vertical positions of the cursor are defined by the pixel coordinate of the screen. The pixel coordinate of the screen ranges from (0,0) to (600,400). Wherein, (0,0) is located at the left top corner of the screen and (600,400) is located at the right bottom corner of the screen. The horizontal pixel range is from 0 to 600 and the vertical pixel range is from 0 to 400.
- When cursor A exceeds the vertical range of the screen display, the query always returns 4294967295.

Return Format The query returns an integer.

Example :CURSor:TRACk:AY? /*The query returns 284*/

:CURSor:TRACk:BY?

Syntax :CURSor:TRACk:BY?

Description Query the vertical position of cursor B in the track cursor measurement mode.

Explanation
- The horizontal and vertical positions of the cursor are defined by the pixel coordinate of the screen. The pixel coordinate of the screen ranges from (0,0) to (600,400). Wherein, (0,0) is located at the left top corner of the screen and (600,400) is located at the right bottom corner of the screen. The horizontal pixel range is from 0 to 600 and the vertical pixel range is from 0 to 400.
- When cursor B exceeds the vertical range of the screen display, the query always returns 4294967295.

Return Format The query returns an integer.

Example :CURSor:TRACk:BY? /*The query returns 200*/

:CURSor:TRACk:AXValue?

Syntax :CURSor:TRACk:AXValue?

Description Query the X value of cursor A in the track cursor measurement mode. The default unit is s.

Return Format The query returns the X value of cursor A in scientific notation.

Example :CURSor:TRACk:AXValue? /*The query returns -4.000000e-06*/

Related Command :CURSor:TRACk:AX

:CURSor:TRACk:AYValue?

Syntax :CURSor:TRACk:AYValue?

Description Query the Y value of cursor A in the track cursor measurement mode. The unit is the same as the channel unit currently selected.

Return Format The query returns the Y value of cursor A in scientific notation.

Example :CURSor:TRACk:AYValue? /*The query returns -4.000000e-01*/

Related Commands :CHANnel<n>:UNITs

:CURSor:TRACk:AY?

:CURSor:TRACk:BXValue?

Syntax :CURSor:TRACk:BXValue?

Description Query the X value of cursor B in the track cursor measurement mode. The default unit is s.

Return Format The query returns the X value of cursor B in scientific notation.

Example :CURSor:TRACk:BXValue? /*The query returns 4.000000e-06*/

Related Command :CURSor:TRACk:BX

:CURSor:TRACk:BYValue?

Syntax :CURSor:TRACk:BYValue?

Description Query the Y value of cursor B in the track cursor measurement mode. The unit is the same as the channel unit currently selected.

Return Format The query returns the Y value of cursor B in scientific notation.

Example :CURSor:TRACk:BYValue? /*The query returns 4.000000e-01*/

Related Commands :CHANnel<n>:UNITs

:CURSor:TRACk:BY?

:CURSor:TRACk:XDELta?

Syntax :CURSor:TRACk:XDELta?

Description Query the difference between the X values of cursor A and cursor B (BX-AX) in the track cursor measurement mode. The default unit is s.

Return Format The query returns the difference in scientific notation.

Example :CURSor:TRACk:XDELta? /*The query returns 8.000000e-06*/

Related Commands :CURSor:TRACk:AX

:CURSor:TRACk:BX

:CURSor:TRACk:YDELta?

Syntax :CURSor:TRACk:YDELta?

Description Query the difference between the Y values of cursor A and cursor B (BY-AY) in the track cursor measurement mode. The unit is the same as the channel unit currently selected.

Return Format The query returns the difference in scientific notation.

Example :CURSor:TRACk:YDELta? /*The query returns 8.000000e-01*/

Related Commands :CURSor:TRACk:AY?
:CURSor:TRACk:BY?

:CURSor:TRACk:IXDELTA?

Syntax :CURSor:TRACk:IXDELTA?

Description Query the reciprocal of the absolute value of the difference between the X values of cursor A and cursor B (1/|dX|) in the track cursor measurement mode. The default unit is Hz.

Return Format The query returns 1/|dX| in scientific notation.

Example :CURSor:TRACk:IXDELTA? /*The query returns 1.250000e+05*/

Related Commands :CURSor:TRACk:AX
:CURSor:TRACk:BX

:CURSor:AUTO

Command List:

- :CURSor:AUTO:ITEM
- :CURSor:AUTO:AX?
- :CURSor:AUTO:BX?
- :CURSor:AUTO:AY?
- :CURSor:AUTO:BY?
- :CURSor:AUTO:AXValue?
- :CURSor:AUTO:AYValue?
- :CURSor:AUTO:BXValue?
- :CURSor:AUTO:BYValue?

:CURSor:AUTO:ITEM

Syntax :CURSor:AUTO:ITEM <item>
:CURSor:AUTO:ITEM?

Description The auto cursor function can measure 37 waveform parameters. Using this command, you can select the parameters to be measured by the auto cursor from the five parameters enabled last or query the parameters currently measured by the auto cursor.

Parameter

Name	Type	Range	Default
<item>	Discrete	{OFF|ITEM1|ITEM2|ITEM3|ITEM4|ITEM5}	OFF

Explanation

- This command is only valid when the auto cursor mode is selected. You can select the auto cursor measurement mode using the :CURSor:MODE command.
- The 37 waveform parameters are listed below (see the detailed introduction in :MEASure Commands). The parameters can be enabled by the :MEASure:ITEM command.
  Period, Frequency, Rise Time, Fall Time, +Width, -Width, +Duty, -Duty, +Pulses, -Pulses, +Edges, -Edges, tVmax, tVmin, +Rate, +Rate, Delay 1→2, Delay 1→2, Phase 1→2, Phase 1→2, Vmax, Vmin, Vpp, Vtop, Vbase, Vamp, Vupper, Vmid, Vlower, Vavg, Vrms, Overshoot, Preshoot, Area, Period Area, Period Vrms, Variance.

Return Format The query returns OFF, ITEM1, ITEM2, ITEM3, ITEM4, or ITEM5.

Example
```
:CURSor:AUTO:ITEM ITEM3      /*Use auto cursor to measure ITEM3*/
:CURSor:AUTO:ITEM?           /*The query returns ITEM3*/
```

:CURSor:AUTO:AX?

Syntax :CURSor:AUTO:AX?

Description Query the horizontal position of cursor A in auto cursor measurement.

Explanation

- The horizontal and vertical positions of the cursor are defined by the pixel coordinate of the screen. The pixel coordinate of the screen ranges from (0,0) to (600,400). Wherein, (0,0) is located at the left top corner of the screen and (600,400) is located at the right bottom corner of the screen. The horizontal pixel range is from 0 to 600 and the vertical pixel range is from 0 to 400.
- In auto cursor measurement, cursor A of X type is not required for some of the measurement items. At this point, the query returns 4294967295.

Return Format The query returns an integer between 5 and 594.

:CURSor:AUTO:BX?

Syntax :CURSor:AUTO:BX?

Description Query the horizontal position of cursor B in auto cursor measurement.

Explanation

- The horizontal and vertical positions of the cursor are defined by the pixel coordinate of the screen. The pixel coordinate of the screen ranges from (0,0) to (600,400). Wherein, (0,0) is located at the left top corner of the screen and (600,400) is located at the right bottom corner of the screen. The horizontal pixel range is from 0 to 600 and the vertical pixel range is from 0 to 400.
- In auto cursor measurement, cursor B of X type is not required for some of the measurement items. At this point, the query returns 4294967295.

Return Format The query returns an integer between 5 and 594.

:CURSor:AUTO:AY?

Syntax :CURSor:AUTO:AY?

Description Query the vertical position of cursor A in auto cursor measurement.

Explanation
- The horizontal and vertical positions of the cursor are defined by the pixel coordinate of the screen. The pixel coordinate of the screen ranges from (0,0) to (600,400). Wherein, (0,0) is located at the left top corner of the screen and (600,400) is located at the right bottom corner of the screen. The horizontal pixel range is from 0 to 600 and the vertical pixel range is from 0 to 400.
- In auto cursor measurement, cursor A of Y type is not required for some of the measurement items. At this point, the query returns 4294967295.

Return Format The query returns an integer between 5 and 394.

:CURSor:AUTO:BY?

Syntax :CURSor:AUTO:BY?

Description Query the vertical position of cursor B in auto cursor measurement.

Explanation
- The horizontal and vertical positions of the cursor are defined by the pixel coordinate of the screen. The pixel coordinate of the screen ranges from (0,0) to (600,400). Wherein, (0,0) is located at the left top corner of the screen and (600,400) is located at the right bottom corner of the screen. The horizontal pixel range is from 0 to 600 and the vertical pixel range is from 0 to 400.
- In auto cursor measurement, cursor B of Y type is not required for some of the measurement items. At this point, the query returns 4294967295.

Return Format The query returns an integer between 5 and 394.

:CURSor:AUTO:AXValue?

Syntax :CURSor:AUTO:AXValue?

Description Query the X value of cursor A in auto cursor measurement. The unit depends on the horizontal unit currently selected.

Explanation In auto cursor measurement, cursor A of X type is not required for some of the measurement items. At this point, the query returns 9.9E37.

Return Format The query returns the X value of cursor A in scientific notation.

Example :CURSor:AUTO:AXValue? /*The query returns -4.000000e-06*/

:CURSor:AUTO:AYValue?

Syntax :CURSor:AUTO:AYValue?

Description Query the Y value of cursor A in auto cursor measurement. The unit depends on the vertical unit currently selected.

Explanation In auto cursor measurement, cursor A of Y type is not required for some of the measurement items. At this point, the query returns 9.9E37.

Return Format The query returns the Y value of cursor A in scientific notation.

Example :CURSor:AUTO:AYValue? /*The query returns -4.000000e-01*/

:CURSor:AUTO:BXValue?

Syntax :CURSor:AUTO:BXValue?

Description Query the X value of cursor B in auto cursor measurement. The unit depends on the horizontal unit currently selected.

Explanation In auto cursor measurement, cursor B of X type is not required for some of the measurement items. At this point, the query returns 9.9E37.

Return Format The query returns the X value of cursor B in scientific notation.

Example :CURSor:AUTO:BXValue? /*The query returns -4.000000e-06*/

:CURSor:AUTO:BYValue?

Syntax :CURSor:AUTO:BYValue?

Description Query the Y value of cursor B in auto cursor measurement. The unit depends on the vertical unit currently selected.

Explanation In auto cursor measurement, cursor B of Y type is not required for some of the measurement items. At this point, the query returns 9.9E37.

Return Format The query returns the Y value of cursor B in scientific notation.

Example :CURSor:AUTO:BYValue? /*The query returns 4.000000e-01*/

:CURSor:XY

The :CURSor:XY commands can only be used when the horizontal timebase mode is XY.

Command List:

:CURSor:XY:AX

Syntax :CURSor:XY:AX <x>

:CURSor:XY:AX?

Description Set or query the horizontal position of cursor A in the XY cursor measurement mode.

Parameter

Name	Type	Range	Default
<x>	Integer	5 to 394	100

Explanation In the XY timebase mode, the horizontal and vertical positions of the cursor are defined by the pixel coordinate of the XY display area. The pixel coordinate of the screen ranges from (0,0) to (400,400). Wherein, (0,0) is located at the right top corner and (400,400) is located at the left bottom corner. The horizontal and vertical pixel ranges are both from 0 to 400.

Return Format The query returns an integer between 5 and 394.

Example

```
:CURSor:XY:AX 200      /*Set the horizontal position of cursor A to 200*/
:CURSor:XY:AX?         /*The query returns 200*/
```

:CURSor:XY:BX

Syntax :CURSor:XY:BX <x>

:CURSor:XY:BX?

Description Set or query the horizontal position of cursor B in the XY cursor measurement mode.

Parameter

Name	Type	Range	Default
<x>	Integer	5 to 394	300

Explanation In the XY timebase mode, the horizontal and vertical positions of the cursor are defined by the pixel coordinate of the XY display area. The pixel coordinate of the screen ranges from (0,0) to (400,400). Wherein, (0,0) is located at the right top corner and (400,400) is located at the left bottom corner. The horizontal and vertical pixel ranges are both from 0 to 400.

Return Format The query returns an integer between 5 and 394.

Example
```
:CURSor:XY:BX 200      /*Set the horizontal position of cursor B to 200*/
:CURSor:XY:BX?         /*The query returns 200*/
```

:CURSor:XY:AY

Syntax :CURSor:XY:AY <y>

:CURSor:XY:AY?

Description Set or query the vetical position of cursor A in the XY cursor measurement mode.

Parameter

Name	Type	Range	Default
<x>	Integer	5 to 394	100

Explanation In the XY timebase mode, the horizontal and vertical positions are defined by the pixel coordinate of the XY display area. The pixel coordinate of the screen ranges from (0,0) to (400,400). Wherein, (0,0) is located at the right top corner and (400,400) is located at the left bottom corner. The horizontal and vertical pixel ranges are both from 0 to 400.

Return Format The query returns an integer between 5 and 394.

Example
```
:CURSor:XY:AY 200      /*Set the vertical position of cursor A to 200*/
:CURSor:XY:AY?         /*The query returns 200*/
```

:CURSor:XY:BY

Syntax :CURSor:XY:BY <y>

:CURSor:XY:BY?

Description Set or query the vertical position of cursor B in the XY cursor measurement mode.

Parameter

Name	Type	Range	Default
<x>	Integer	5 to 394	300

Explanation In the XY timebase mode, the horizontal and vertical positions are defined by the pixel coordinate of the XY display area. The pixel coordinate of the XY display area ranges from (0,0) to (400,400). Wherein, (0,0) is located at the right top corner and (400,400) is located at the left bottom corner. The horizontal and vertical pixel ranges are both from 0 to 400.

Return Format The query returns an integer between 5 and 394.

Example
```
:CURSor:XY:BY 200      /*Set the vertical position of cursor B to 200*/
:CURSor:XY:BY?         /*The query returns 200*/
```

:CURSor:XY:AXValue?

Syntax :CURSor:XY:AXValue?

Description Query the X value of cursor A in the XY cursor measurement mode. The unit depends on the amplitude unit of the corresponding channel.

Return Format The query returns the X value of cursor A in scientific notation.

Example
```
:CURSor:XY:AXValue?    /*The query returns 2.000000e+00*/
```

Related Command :CHANnel<n>:UNITs

:CURSor:XY:AYValue?

Syntax :CURSor:XY:AYValue?

Description Query the Y value of cursor A in the XY cursor measurement mode. The unit depends on the amplitude unit of the corresponding channel.

Return Format The query returns the Y value of cursor A in scientific notation.

Example :CURSor:XY:AYValue? /*The query returns 2.000000e+00*/

Related Command :CHANnel<n>:UNITs

:CURSor:XY:BXValue?

Syntax :CURSor:XY:BXValue?

Description Query the X value of cursor B in the XY cursor measurement mode. The unit depends on the amplitude unit of the corresponding channel.

Return Format The query returns the X value of cursor B in scientific notation.

Example :CURSor:XY:BXValue? /*The query returns -2.000000e+00*/

Related Command :CHANnel<n>:UNITs

:CURSor:XY:BYValue?

Syntax :CURSor:XY:BYValue?

Description Query the Y value of cursor B in the XY cursor measurement mode. The unit depends on the amplitude unit of the corresponding channel.

Return Format The query returns the Y value of cursor B in scientific notation.

Example :CURSor:XY:BYValue? /*The query returns -2.000000e+00*/

Related Command :CHANnel<n>:UNITs

:DECoder Commands

The :DECoder commands are used to execute decoding settings and operations.

Command List:

:DECoder<n>:MODE

Syntax :DECoder<n>:MODE <mode>

:DECoder<n>:MODE?

Description Set or query the decoder type.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<mode>	Discrete	{PARallel|UART|SPI|IIC}	PARallel

Explanation PARallel, UART, SPI, and IIC correspond to parallel decoding, RS232 decoding, SPI decoding, and I2C decoding respectively.

Return Format The query returns PAR, UART, SPI, or IIC.

Example

```
:DECoder1:MODE SPI      /*Set the decoder type to SPI*/
:DECoder1:MODE?         /*The query returns SPI*/
```

:DECoder<n>:DISPlay

Syntax :DECoder<n>:DISPlay <bool>

:DECoder<n>:DISPlay?

Description Turn on or off the decoder or query the status of the decoder.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Return Format The query returns 1 or 0.

Example
```
:DECoder1:DISPlay ON      /*Turn on Decoder 1*/
:DECoder1:DISPlay?        /*The query returns 1*/
```

:DECoder<n>:FORMat

Syntax :DECoder<n>:FORMat <fmt>

:DECoder<n>:FORMat?

Description Set or query the bus display format.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<fmt>	Discrete	{HEX|ASCii|DEC|BIN|LINE}	ASCii

Explanation

- HEX: hexadecimal; DEC: decimal; BIN: binary
- In LINE format, the actual values of the bus are displayed in binary form and the order is consistent with the bus transmission order. This format is only valid for serial buses which include LSB and MSB endian. If MSB endian is selected, LINE format is the same as binary.

Return Format The query returns HEX, ASC, DEC, BIN, or LINE.

Example
```
:DECoder1:FORMat HEX      /*Set the bus display format to hexadecimal*/
:DECoder1:FORMat?         /*The query returns HEX*/
```

:DECoder<n>:POSition

Syntax :DECoder<n>:POSition <pos>

:DECoder<n>:POSition?

Description Set or query the vertical position of the bus on the screen.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<pos>	Integer	50 to 350	Decoder 1: 350 Decoder 2: 300

Explanation The screen is divided into 400 parts vertically which are marked as 0 to 400 from top to bottom respectively. The range of <pos> is from 50 to 350.

Return The query returns an integer between 50 and 350.

Format

Example
```
:DECoder1:POSition 300      /*Set the vertical position of the bus to 300*/
:DECoder1:POSition?         /*The query returns 300*/
```

:DECoder<n>:THREshold:CHANnel1
:DECoder<n>:THREshold:CHANnel2

Syntax
```
:DECoder<n>:THREshold:CHANnel1 <thre>
:DECoder<n>:THREshold:CHANnel1?
:DECoder<n>:THREshold:CHANnel2 <thre>
:DECoder<n>:THREshold:CHANnel2?
```

Description Set or query the threshold level of the specified analog channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<thre>	Real	(-4 x VerticalScale - VerticalOffset) to (4 x VerticalScale - VerticalOffset)	--

Explanation

- VerticalScale is the vertical scale of the specified analog channel. VerticalOffset is the vertical position of the specified analog channel.
- By default, the auto threshold function of the analog channels of the oscilloscope is turned on. To set the threshold level manually, send the :DECoder<n>:THREshold:AUTO command to turn off the auto threshold function.

Return Format The query returns the level in scientific notation.

Example
```
:DECoder1:THREshold:CHANnel1 1.5      /*Set the threshold level of CH1 to 1.5V*/
:DECoder1:THREshold:CHANnel1?         /*The query returns 1.500000e+00*/
```

:DECoder<n>:THREshold:AUTO

Syntax
```
:DECoder<n>:THREshold:AUTO <bool>
:DECoder<n>:THREshold:AUTO?
```

Description Turn on or off the auto threshold function of the analog channels, or query the status of the auto threshold function of the analog channels.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	1|ON

Explanation By default, the auto threshold function of the analog channels of the oscilloscope is turned on. To set the threshold level manually, send this command to turn off the auto threshold function and use the commands in **Related Commands** to set the threshold manually.

Return Format The query returns 1 or 0.

Example
```
:DECoder1:THREshold:AUTO OFF      /*Turn off the auto threshold function*/
```

:DECoder1:THREshold:AUTO? /*The query returns 0*/

Related Commands :DECoder<n>:THREshold:CHANnel1

:DECoder<n>:THREshold:CHANnel2

:DECoder<n>:CONFig:LABel

Syntax :DECoder<n>:CONFig:LABel <bool>

:DECoder<n>:CONFig:LABel?

Description Turn on or off the label display function, or query the status of the label display function.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	1|ON

Explanation When this function is turned on, the bus label will be displayed at the lower-upper side of the bus (when the bus display is turned on).

Return Format The query returns 1 or 0.

Example
```
:DECoder1:CONFig:LABel ON      /*Turn on the label display function*/
:DECoder1:CONFig:LABel?        /*The query returns 1*/
```

:DECoder<n>:CONFig:LINE

Syntax :DECoder<n>:CONFig:LINE <bool>

:DECoder<n>:CONFig:LINE?

Description Turn on or off the bus display function, or query the status of the bus display function.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	1|ON

Explanation When this function is enabled, the bus will be displayed on the screen. You can send the :DECoder<n>:POSition command to adjust the vertical display position of the bus.

Return Format The query returns 1 or 0.

Example
```
:DECoder1:CONFig:LINE OFF      /*Turn off the bus display function*/
:DECoder1:CONFig:LINE?         /*The query returns 0*/
```

:DECoder<n>:CONFig:FORMat

Syntax :DECoder<n>:CONFig:FORMat <bool>

:DECoder<n>:CONFig:FORMat?

Description Turn on or off the format display function, or query the status of the format display function.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	1|ON

Explanation When this function is turned on, the current bus display format will be displayed at the right of the label display (when the bus display is turned on). You can send the :DECoder<n>:FORMat command to set the bus display format.

Return Format The query returns 1 or 0.

Example
```
:DECoder1:CONFig:FORMat OFF      /*Turn off the format display function*/
:DECoder1:CONFig:FORMat?         /*The query returns 0*/
```

:DECoder<n>:CONFig:ENDian

Syntax :DECoder<n>:CONFig:ENDian <bool>

:DECoder<n>:CONFig:ENDian?

Description Turn on or off the endian display function in serial bus decoding, or query the status of the endian display function in serial bus decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation

- This command is invalid in parallel decoding.
- When this function is enabled, the current bus endian will be displayed at the right of the format display (when the bus display is turned on).

Return Format The query returns 1 or 0.

Example
```
:DECoder1:CONFig:ENDian ON       /*Turn on the endian display function*/
:DECoder1:CONFig:ENDian?         /*The query returns 1*/
```

:DECoder<n>:CONFig:WIDth

Syntax :DECoder<n>:CONFig:WIDth <bool>

:DECoder<n>:CONFig:WIDth?

Description Turn on or off the width display function, or query the status of the width display function.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation When this function is enabled, the width of each frame of data will be displayed at the right of the endian display (when the bus display is turned on).

Return Format The query returns 1 or 0.

Example
```
:DECoder1:CONFig:WIDth ON        /*Turn on the width display function*/
:DECoder1:CONFig:WIDth?          /*The query returns 1*/
```

:DECoder<n>:CONFig:SRATe?

Syntax :DECoder<n>:CONFig:SRATe?

Description Query the current digital sample rate.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--

Explanation The digital sample rate is related to the data source currently selected. By default, the data source is "Trace"; at this point, the digital sample rate is related to the horizontal time base.

Return Format The query returns the digital sample rate in scientific notation.

Example :DECoder1:CONFig:SRATe? /*The query returns 1.000000e+08*/

:DECoder<n>:UART

The :DECoder<n>:UART commands are used to set the RS232 decoding parameters.

Command List:

:DECoder<n>:UART:TX

Syntax :DECoder<n>:UART:TX <tx>

:DECoder<n>:UART:TX?

Description Set or query the TX channel source of RS232 decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<tx>	Discrete	{CHANnel1|CHANnel2|OFF}	CHANnel1

Explanation When OFF is selected, no TX channel source will be set. The TX channel source and RX channel source (:DECoder<n>:UART:RX) cannot be both set to OFF.

Return Format The query returns CHAN1, CHAN2, or OFF.

Example

```
:DECoder1:UART:TX CHANnel2      /*Set the TX channel source to CH2*/
:DECoder1:UART:TX?              /*The query returns CHAN2*/
```

:DECoder<n>:UART:RX

Syntax :DECoder<n>:UART:RX <rx>

:DECoder<n>:UART:RX?

Description Set or query the RX channel source of RS232 decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<rx>	Discrete	{CHANnel1|CHANnel2|OFF}	OFF

Explanation When OFF is selected, no RX channel source will be set. The RX channel source and TX channel source (:DECoder<n>:UART:TX) cannot be both set to OFF.

Return Format The query returns CHAN1, CHAN2, or OFF.

Example
```
:DECoder1:UART:RX CHANnel2      /*Set the RX channel source to CH2*/
:DECoder1:UART:RX?              /*The query returns CHAN2*/
```

:DECoder<n>:UART:POLarity

Syntax
```
:DECoder<n>:UART:POLarity <pol>
:DECoder<n>:UART:POLarity?
```

Description Set or query the polarity of RS232 decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<pol>	Discrete	{NEGative|POSitive}	POSitive

Explanation

- NEGative: negative polarity (), namely high level is 0 and low level is 1. The RS232 standard uses negative polarity.
- POSitive: positive polarity (), namely high level is 1 and low level is 0.

Return Format The query returns NEG or POS.

Example
```
:DECoder1:UART:POLarity NEGative    /*Set the polarity of RS232 decoding to
                                      negative*/
:DECoder1:UART:POLarity?            /*The query returns NEG*/
```

:DECoder<n>:UART:ENDian

Syntax
```
:DECoder<n>:UART:ENDian <endian>
:DECoder<n>:UART:ENDian?
```

Description Set or query the endian of RS232 decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<endian>	Discrete	{LSB|MSB}	LSB

Return Format The query returns LSB or MSB.

Example
```
:DECoder1:UART:ENDian MSB       /*Set the endian of RS232 decoding to MSB*/
:DECoder1:UART:ENDian?          /*The query returns MSB*/
```

:DECoder<n>:UART:BAUD

Syntax
```
:DECoder<n>:UART:BAUD <baud>
:DECoder<n>:UART:BAUD?
```

Description Set or query the buad rate of RS232 decoding. The default unit is bps (baud per second).

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<baud>	Integer	110 to 20M	9600

Return Format The query returns the current baud rate in integer.

Example
```
:DECoder1:UART:BAUD 57600     /*Set the buad rate of RS232 decoding to 57600bps*/
:DECoder1:UART:BAUD?          /*The query returns 57600*/
```

:DECoder<n>:UART:WIDTh

Syntax :DECoder<n>:UART:WIDTh <wid>

:DECoder<n>:UART:WIDTh?

Description Set or query the width of each frame of data in RS232 decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<wid>	Integer	5 to 8	8

Return Format The query returns an integer between 5 and 8.

Example
```
:DECoder1:UART:WIDTh 7      /*Set the data width in RS232 decoding to 7*/
:DECoder1:UART:WIDTh?       /*The query returns 7*/
```

:DECoder<n>:UART:STOP

Syntax :DECoder<n>:UART:STOP <stop>

:DECoder<n>:UART:STOP?

Description Set or query the stop bit after each frame of data in RS232 decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<stop>	Discrete	{1|1.5|2}	1

Return Format The query returns 1, 1.5, or 2.

Example
```
:DECoder1:UART:STOP 1.5      /*Set the stop bit in RS232 decoding to 1.5*/
:DECoder1:UART:STOP?         /*The query returns 1.5*/
```

:DECoder<n>:UART:PARity

Syntax :DECoder<n>:UART:PARity <parity>

:DECoder<n>:UART:PARity?

Description Set or query the even-odd check mode of the data transmission in RS232 decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<parity>	Discrete	{NONE|EVEN|ODD}	NONE

Return Format The query returns NONE, EVEN, or ODD.

Example
```
:DECoder1:UART:PARity ODD      /*Set the even-odd check mode in RS232 decoding to odd*/
:DECoder1:UART:PARity?         /*The query returns ODD*/
```

:DECoder<n>:IIC

The :DECoder<n>:IIC commands are used to set the I2C decoding parameters.

Command List:

:DECoder<n>:IIC:CLK

Syntax :DECoder<n>:IIC:CLK <clk>

:DECoder<n>:IIC:CLK?

Description Set or query the signal source of the clock channel in I2C decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<clk>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example

```
:DECoder1:IIC:CLK CHANnel1      /*Set the signal source of the clock channel in I2C
                                  decoding to CH1*/
:DECoder1:IIC:CLK?              /*The query returns CHAN1*/
```

:DECoder<n>:IIC:DATA

Syntax :DECoder<n>:IIC:DATA <dat>

:DECoder<n>:IIC:DATA?

Description Set or query the signal source of the data channel in I2C decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<dat>	Discrete	{CHANnel1|CHANnel2}	CHANnel2

Return Format The query returns CHAN1 or CHAN2.

Example

```
:DECoder1:IIC:DATA CHANnel1     /*Set the signal source of the data channel in I2C
                                  decoding to CH1*/
:DECoder1:IIC:DATA?             /*The query returns CHAN1*/
```

:DECoder<n>:IIC:ADDRess

Syntax :DECoder<n>:IIC:ADDRess <addr>

:DECoder<n>:IIC:ADDRess?

Description Set or query the address mode of I2C decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<addr>	Discrete	{NORMal|RW}	NORMal

Explanation

- NORMal: the address bits (:TRIGger:IIC:AWIDth) does not include the R/W bit.
- RW: the address bits (:TRIGger:IIC:AWIDth) includes the R/W bit.

Return Format The query returns NORM or RW.

Example

```
:DECoder1:IIC:ADDRess RW      /*Set the address of I2C decoding to include the R/W bit*/
:DECoder1:IIC:ADDRess?        /*The query returns RW*/
```

:DECoder<n>:SPI

The :DECoder<n>:SPI commands are used to set the SPI decoding parameters.

Command List:

:DECoder<n>:SPI:CLK

Syntax :DECoder<n>:SPI:CLK <clk>

:DECoder<n>:SPI:CLK?

Description Set or query the signal source of the clock channel in SPI decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<clk>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example

```
:DECoder1:SPI:CLK CHANnel1      /*Set the signal source of the clock channel in SPI
                                  decoding to CH1*/
:DECoder1:SPI:CLK?              /*The query returns CHAN1*/
```

:DECoder<n>:SPI:MISO

Syntax :DECoder<n>:SPI:MISO <miso>

:DECoder<n>:SPI:MISO?

Description Set or query the MISO channel source in SPI decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<miso>	Discrete	{CHANnel1|CHANnel2|OFF}	OFF

Explanation When OFF is selected, no MISO channel source will be set. The MISO channel source and MOSI channel source (:DECoder<n>:SPI:MOSI) cannot be both set to OFF.

Return Format The query returns CHAN1, CHAN2, or OFF.

Example
```
:DECoder1:SPI:MISO CHANnel1    /*Set the MISO channel source in SPI decoding to CH1*/
:DECoder1:SPI:MISO?            /*The query returns CHAN1*/
```

:DECoder<n>:SPI:MOSI

Syntax
```
:DECoder<n>:SPI:MOSI <mosi>
:DECoder<n>:SPI:MOSI?
```

Description Set or query the MOSI channel source in SPI decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<mosi>	Discrete	{CHANnel1|CHANnel2|OFF}	CHANnel2

Explanation When OFF is selected, no MOSI channel source will be set. The MOSI channel source and MISO channel source (:DECoder<n>:SPI:MISO) cannot be both set to OFF.

Return Format The query returns CHAN1, CHAN2, or OFF.

Example
```
:DECoder1:SPI:MOSI CHANnel1    /*Set the MOSI channel source in SPI decoding to CH1*/
:DECoder1:SPI:MOSI?            /*The query returns CHAN1*/
```

:DECoder<n>:SPI:CS

Syntax
```
:DECoder<n>:SPI:CS <cs>
:DECoder<n>:SPI:CS?
```

Description Set or query the CS channel source in SPI decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<cs>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Explanation This command is only valid in the CS mode (:DECoder<n>:SPI:MODE).

Return Format The query returns CHAN1 or CHAN2.

Example
```
:DECoder1:SPI:CS CHANnel2      /*Set the CS channel source in SPI decoding to CH2*/
:DECoder1:SPI:CS?              /*The query returns CHAN2*/
```

:DECoder<n>:SPI:SELect

Syntax :DECoder<n>:SPI:SELect <CsNcs>

:DECoder<n>:SPI:SELect?

Description Set or query the CS polarity in SPI decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<CsNcs>	Discrete	{NCS|CS}	NCS

Explanation

- NCS: low level is valid (). The instrument starts transmitting data when the CS is low.
- CS: high level is valid (). The instrument starts transmitting data when the CS is high.
- This command is only valid in the CS mode (:DECoder<n>:SPI:MODE).

Return Format The query returns NCS or CS.

Example

```
:DECoder1:SPI:SELect CS      /*Set the CS polarity to high level is valid*/
:DECoder1:SPI:SELect?        /*The query returns CS*/
```

:DECoder<n>:SPI:MODE

Syntax :DECoder<n>:SPI:MODE <CsTmo>

:DECoder<n>:SPI:MODE?

Description Set or query the frame synchronization mode of SPI decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<CsTmo>	Discrete	{CS|TIMeout}	TIMeout

Explanation

- CS: it contains a chip select line (CS). You can perform frame synchronization according to CS. At this point, you need to send the :DECoder<n>:SPI:CS and :DECoder<n>:SPI:SELect commands to set the CS channel source and polarity.
- TIMeout: you can perform frame synchronization according to the timeout time. At this point, you need to send the :DECoder<n>:SPI:TIMeout command to set the timeout time.

Return Format The query returns CS or TIM.

Example

```
:DECoder1:SPI:MODE CS    /*Set the frame synchronization mode of SPI decoding to
                           CS*/
:DECoder1:SPI:MODE?      /*The query returns CS*/
```

:DECoder<n>:SPI:TIMeout

Syntax :DECoder<n>:SPI:TIMeout <tmo>

:DECoder<n>:SPI:TIMeout?

Description Set or query the timeout time in the timeout mode of SPI decoding. The default unit is s.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<tmo>	Real	Refer to **Explanation**	1.00us

Explanation

- The timeout time should be greater than the maximum pulse width of the clock and lower than the idle time between frames.
- This command is only valid in the timeout mode (:DECoder<n>:SPI:MODE).

Return Format The query returns the timeout time in scientific notation.

Example

```
:DECoder1:SPI:TIMeout 0.000005      /*Set the timeout time to 5us*/
:DECoder1:SPI:TIMeout?              /*The query returns 5.000000e-06*/
```

:DECoder<n>:SPI:POLarity

Syntax

```
:DECoder<n>:SPI:POLarity <pol>
:DECoder<n>:SPI:POLarity?
```

Description Set or query the polarity of the SDA data line in SPI decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<pol>	Discrete	{NEGative|POSitive}	POSitive

Explanation

- NEGative:  . The low level is 1.
- POSitive:  . The high level is 1.

Return Format The query returns NEG or POS.

Example

```
:DECoder1:SPI:POLarity NEG ative    /*Set the polarity of the SDA data line in SPI
                                      decoding to negative*/
:DECoder1:SPI:POLarity?             /*The query returns NEG*/
```

:DECoder<n>:SPI:EDGE

Syntax

```
:DECoder<n>:SPI:EDGE <edge>
:DECoder<n>:SPI:EDGE?
```

Description Set or query the clock type when the instrument samples the data line in SPI decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<edge>	Discrete	{RISE|FALL}	RISE

Return Format The query returns RISE or FALL.

Example

```
:DECoder1:SPI:EDGE FALL     /*Set the instrument to sample data on the falling edge
                              of the clock in SPI decoding*/
:DECoder1:SPI:EDGE?         /*The query returns FALL*/
```

:DECoder<n>:SPI:ENDian

Syntax :DECoder<n>:SPI:ENDian <endian>

:DECoder<n>:SPI:ENDian?

Description Set or query the endian of the SPI decoding data.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<endian>	Discrete	{LSB|MSB}	MSB

Return Format The query returns LSB or MSB.

Example
```
:DECoder1:SPI:ENDian MSB      /*Set the data endian of the SPI decoding to MSB*/
:DECoder1:SPI:ENDian?         /*The query returns MSB*/
```

:DECoder<n>:SPI:WIDTh

Syntax :DECoder<n>:SPI:WIDTh <wid>

:DECoder<n>:SPI:WIDTh?

Description Set or query the number of bits of each frame of data in SPI decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<wid>	Integer	8 to 32	8

Return Format The query returns an integer between 8 and 32.

Example
```
:DECoder1:SPI:WIDTh 16      /*Set the data width in SPI decoding to 16*/
:DECoder1:SPI:WIDTh?        /*The query returns 16*/
```

:DECoder<n>:PARallel

The :DECoder<n>:PARallel commands are used to set the parallel decoding parameters.

Command List:

- :DECoder<n>:PARallel:CLK
- :DECoder<n>:PARallel:EDGE
- :DECoder<n>:PARallel:WIDTh
- :DECoder<n>:PARallel:BITX
- :DECoder<n>:PARallel:SOURce
- :DECoder<n>:PARallel:POLarity
- :DECoder<n>:PARallel:NREJect
- :DECoder<n>:PARallel:NRTime
- :DECoder<n>:PARallel:CCOMpensation
- :DECoder<n>:PARallel:PLOT

:DECoder<n>:PARallel:CLK

Syntax :DECoder<n>:PARallel:CLK <clk>

:DECoder<n>:PARallel:CLK?

Description Set or query the CLK channel source of parallel decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<clk>	Discrete	{CHANnel1|CHANnel2|OFF}	CHANnel1

Return Format The query returns CHAN1, CHAN2, or OFF.

Example

```
:DECoder1:PARallel:CLK CHANnel1      /*Set the CLK channel source of parallel
                                       decoding to CH1*/
:DECoder1:PARallel:CLK?              /*The query returns CHAN1*/
```

:DECoder<n>:PARallel:EDGE

Syntax :DECoder<n>:PARallel:EDGE <edge>

:DECoder<n>:PARallel:EDGE?

Description Set or query the edge type of the clock channel when the instrument samples the data channel in parallel decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<edge>	Discrete	{RISE|FALL|BOTH}	RISE

Explanation If no clock channel is selected (:DECoder<n>:PARallel:CLK), the instrument will sample when the channel data jumps.

Return Format The query returns RISE, FALL, or BOTH.

Example
```
:DECoder1:PARallel:EDGE BOTH      /*Set the oscilloscope to sample data on any edge
                                    of the clock channel in parallel decoding*/
:DECoder1:PARallel:EDGE?          /*The query returns BOTH*/
```

:DECoder<n>:PARallel:WIDTh

Syntax :DECoder<n>:PARallel:WIDTh <wid>

:DECoder<n>:PARallel:WIDTh?

Description Set or query the data width (namely the number of bits of each frame of data) of the parallel bus.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<wid>	integer	1 to 16	8

Explanation After setting the data width using this command, send the :DECoder<n>:PARallel:BITX and :DECoder<n>:PARallel:SOURce commands to select each bit and set the channel source for each bit respectively.

Return Format The query returns an integer between 1 and 16.

Example
```
:DECoder1:PARallel:WIDTh 16       /*Set the data width in parallel decoding to 16*/
:DECoder1:PARallel:WIDTh?         /*The query returns 16*/
```

:DECoder<n>:PARallel:BITX

Syntax :DECoder<n>:PARallel:BITX <bit>

:DECoder<n>:PARallel:BITX?

Description Set or query the data bit that requires a channel source on the parallel bus.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bit>	Integer	0 to (data width - 1)	0

Explanation

- Set the data width using the :DECoder<n>:PARallel:WIDTh command.
- After selecting the desired bit, send the :DECoder<n>:PARallel:SOURce command to set the channel source of this bit.

Return Format The query returns the current data bit in integer.

Example
```
:DECoder1:PARallel:BITX 2         /*Set the current bit to 2*/
:DECoder1:PARallel:BITX?          /*The query returns 2*/
```

:DECoder<n>:PARallel:SOURce

Syntax :DECoder<n>:PARallel:SOURce <src>

:DECoder<n>:PARallel:SOURce?

Description Set ro query the channel source of the data bit currently selected.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<src>	Discrete	{CHANnel1|CHANnel2|}	Related to the bit selected

Explanation Before sending this command, use the :DECoder<n>:PARallel:BITX command to select the desired data bit.

Return Format The query returns CHAN1 or CHAN2.

Example

```
:DECoder1:PARallel:SOURce CHANnel2    /*Set the channel source of the current bit to CH2*/
:DECoder1:PARallel:SOURce?            /*The query returns CHAN2*/
```

Related Command :DECoder<n>:PARallel:WIDTh

:DECoder<n>:PARallel:POLarity

Syntax

```
:DECoder<n>:PARallel:POLarity <pol>
:DECoder<n>:PARallel:POLarity?
```

Description Set ro query the data polarity of parallel decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<pol>	Discrete	{NEGative|POSitive}	POSitive

Explanation

- NEGative:  . The low level is 1.
- POSitive:  . The high level is 1.

Return Format The query returns NEG or POS.

Example

```
:DECoder1:PARallel:POLarity NEGative    /*Set the data polarity of parallel decoding to negative*/
:DECoder1:PARallel:POLarity?            /*The query returns NEG*/
```

:DECoder<n>:PARallel:NREJect

Syntax

```
:DECoder<n>:PARallel:NREJect <bool>
:DECoder<n>:PARallel:NREJect?
```

Description Turn on or off the noise rejection function of parallel decoding, or query the status of the noise rejection function of parallel decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation

- Noise rejection can remove the data without enough duration on the bus to eliminate the glitches of the actual circuit.

- When the noise rejection is turned on, sending the :DECoder<n>:PARallel:NRTime command can set the desired rejection time.

Return Format The query returns 1 or 0.

Example
```
:DECoder1:PARallel:NREJect ON      /*Turn on the noise rejection function*/
:DECoder1:PARallel:NREJect?        /*The query returns 1*/
```

:DECoder<n>:PARallel:NRTime

Syntax :DECoder<n>:PARallel:NRTime <time>

:DECoder<n>:PARallel:NRTime?

Description Set or query the noise rejection time of parallel decoding. The default unit is s.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<time>	Real	0.00s to 100ms	0.00s

Explanation Before sending this command, send the :DECoder<n>:PARallel:NREJect command to turn on the noise rejection function.

Return Format The query returns the noise rejection time in scientific notation.

Example
```
:DECoder1:PARallel:NRTime 0.01     /*Set the noise rejection time to 10ms*/
:DECoder1:PARallel:NRTime?         /*The query returns 1.000000e-02*/
```

:DECoder<n>:PARallel:CCOMpensation

Syntax :DECoder<n>:PARallel:CCOMpensation <comp>

:DECoder<n>:PARallel:CCOMpensation?

Description Set or query the clock compensation time of parallel decoding. The default unit is s.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<comp>	Real	-100ms to 100ms	0.00s

Explanation

- Setting the compensation time can make fine adjustment of the pahse deviation between the clock line and data line.
- This command is invalid when the CLK channel source is set to OFF (:DECoder<n>:PARallel:CLK).

Return Format The query returns the compensation time in scientific notation.

Example
```
:DECoder1:PARallel:CCOMpensation 0.01     /*Set the compensation time to 10ms*/
:DECoder1:PARallel:CCOMpensation?         /*The query returns 1.000000e-02*/
```

:DECoder<n>:PARallel:PLOT

Syntax :DECoder<n>:PARallel:PLOT <bool>

:DECoder<n>:PARallel:PLOT?

Description Turn on or off the curve function of parallel decoding, or query the status of the curve function of parallel decoding.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation When this function is turned on, the variation trend of the bus data is displayed in vector diagram form.

Return Format The query returns 1 or 0.

Example

```
:DECoder1:PARallel:PLOT ON      /*Turn on the curve function*/
:DECoder1:PARallel:PLOT?        /*The query returns 1*/
```

:DISPlay Commands

The :DISPlay commands can be used to set the waveform display mode, persistence time, waveform intensity, screen grid type and grid brightness.

Command List:

:DISPlay:CLEar

Syntax :DISPlay:CLEar

Description Clear all the waveforms on the screen.

Explanation

- If the oscilloscope is in the RUN state, waveform will still be displayed.
- This command is equivalent to pressing the **CLEAR** key on the front panel. Sending the :CLEar command can also clear all the waveforms on the screen.

Related Command :RUN

:DISPlay:DATA?

Syntax :DISPlay:DATA? [<color>,<invert>,<format>]

Description Read the data stream of the image currently displayed on the screen and set the color, invert display, and format of the image acquired.

Parameter

Name	Type	Range	Default
<color>	Bool	{ON|OFF}	ON
<invert>	Bool	{{1|ON}|{0|OFF}}	0|OFF
<format>	Discrete	{BMP24|BMP8|PNG|JPEG|TIFF}	BMP24

Explanation

- <color>: color of the image; ON denotes color and OFF denotes intensity graded color.
  <invert>: the invert function; 1|ON denotes turning on the invert function and 0|OFF denotes turning off the invert function.
- When [<color>,<invert>,<format>] is omitted, by default, the image color (:STORage:IMAGe:COLor) and the status of the invert function (:STORage:IMAGe:INVERT) currently selected are used and the image format is set to BMP24.
- The command is sent from the PC to the instrument through the VISA interface. The instrument responds to the command and directly returns the data stream of the image currently displayed to the buffer area of the PC.

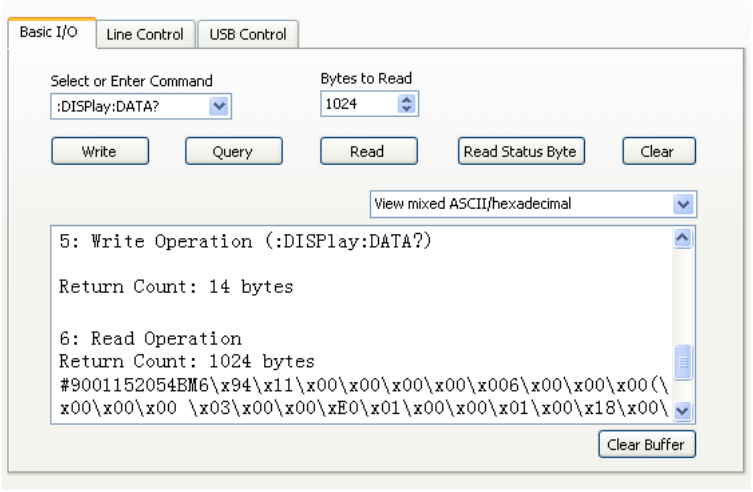
Return Format The format of the data stream is as follows.

Component	Size (length)	Example	Explanation
TMC Blockheader	N[1]+2	#9001152054	TMC Blockheader ::= #NXXXXXX is used to describe the length of the data stream. Wherein, # is the start denoter of the data stream; N is less than or equal to 9 and the N figures following it denote the length of the data stream in bytes. For example, #9001152054; wherein, N is 9 and 001152054 denotes that the data stream contains 1152054 bytes of effective data.
Image Data (take BMP24 as an example)	800x480x3+54 =1152054[2]	BM...	Specific image data.

Note[1]: N is the width of the data length in the TMC header. For example, the number "9" behind "#" in #9001152054.
Note[2]: The width is 800, the height is 480, the bit depth is 24 bit = 3 byte, 54 is the size of the bitmap file header.

Example

1. Make sure that the buffer is large enough to receive the data stream, otherwise the program might be abnormal when reading the data stream.
2. The returned data stream contains the TMC data header which should be removed to make the data stream a standard image data stream.
3. When the data size is larger than 1 M and the communication speed of the interface is not fast enough, you need to set an appropriate timeout time.
4. The terminator '\n'(0X0A) at the end of the data should be removed.



:DISPlay:TYPE

Syntax :DISPlay:TYPE <type>

:DISPlay:TYPE?

Description Set or query the display mode of the waveform on the screen.

Parameter

Name	Type	Range	Default
<type>	Discrete	{VECTors|DOTS}	VECTors

Explanation

- VECTors: the sample points are connected by lines. Normally, this mode can provide the most vivid waveform to view the steep edge of the waveform (such as square waveform).
- DOTS: display the sample points directly. You can directly view each sample point and use the cursor to measure the X and Y values of the sample point.

Return Format The query returns VECT or DOTS.

Example

```
:DISPlay:TYPE DOTS      /*Select dots display mode*/
:DISPlay:TYPE?          /*The query returns DOTS*/
```

:DISPlay:GRADing:TIME

Syntax :DISPlay:GRADing:TIME <time>

:DISPlay:GRADing:TIME?

Description Set or query the persistence time. The default unit is s.

Parameter

Name	Type	Range	Default
<time>	Discrete	{MIN|0.1|0.2|0.5|1|5|10|INFinite}	MIN

Explanation

- MIN: set the persistence time to its minimum to view the waveform changing in high refresh rate.
- Specific Values: set the persistence time to one of the values listed above to observe glitch that changes relatively slowly or glitch with low occurrence probability.
- INFinite: in this mode, the oscilloscope displays the newly acquired waveform without clearing the waveform formerly acquired. It can be used to measure noise and jitter as well as capture incidental events.

Return Format The query returns MIN, 0.1, 0.2, 0.5, 1, 5, 10, or INF.

Example

```
:DISPlay:GRADing:TIME 0.1     /*Set the persistence time to 0.1s*/
:DISPlay:GRADing:TIME?        /*The query returns 0.1*/
```

:DISPlay:WBRightness

Syntax :DISPlay:WBRightness <time>

:DISPlay:WBRightness?

Description Set or query the waveform brightness.

Parameter

Name	Type	Range	Default
<time>	Integer	0 to 100	60

Return Format The query returns an integer between 0 and 100.

Example

```
:DISPlay:WBRightness 50       /*Set the waveform brightness to 50%*/
:DISPlay:WBRightness?         /*The query returns 50*/
```

:DISPlay:GRID

Syntax :DISPlay:GRID <grid>

:DISPlay:GRID?

Description Set or query the grid type of screen display.

Parameter

Name	Type	Range	Default
<grid>	Discrete	{FULL|HALF|NONE}	FULL

Explanation FULL: turn the background grid and coordinate on.
HALF: turn the background grid off and coordinate on.
NONE: turn the background grid and coordinate off.

Return Format The query returns FULL, HALF, or NONE.

Example
```
:DISPlay:GRID NONE      /*Turn the background grid and coordinate off*/
:DISPlay:GRID?          /*The query returns NONE*/
```

:DISPlay:GBRightness

Syntax :DISPlay:GBRightness <brightness>

:DISPlay:GBRightness?

Description Set or query the brightness of the screen grid.

Parameter

Name	Type	Range	Default
<brightness>	Integer	0 to 100	50

Return Format The query returns an integer between 0 and 100.

Example
```
:DISPlay:GBRightness 60      /*Set the brightness of the screen grid to 60%*/
:DISPlay:GBRightness?        /*The query returns 60*/
```

:ETABle Commands

The :ETABle commands are used to set the parameters related to the decoding event table.

Command List:

:ETABle<n>:DISP

Syntax :ETABle<n>:DISP <bool>

:ETABle<n>:DISP?

Description Turn on or off the decoding event table, or query the status of the decoding event table.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation This command is only valid when the decoder is turned on (:DECoder<n>:DISPlay).

Return Format The query returns 1 or 0.

Example

```
:ETABle1:DISP ON      /*Turn on the decoding event table*/
:ETABle1:DISP?        /*The query returns 1*/
```

:ETABle<n>:FORMat

Syntax :ETABle<n>:FORMat <fmt>

:ETABle<n>:FORMat?

Description Set or query the data display format of the event table.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<fmt>	Discrete	{HEX|ASCii|DEC}	HEX

Return Format The query returns HEX, ASC, or DEC.

Example

```
:ETABle1:FORMat ASCii     /*Set the data display format of the event table to ASCII*/
:ETABle1:FORMat?          /*The query returns ASC*/
```

:ETABle<n>:VIEW

Syntax :ETABle<n>:VIEW <view>

:ETABle<n>:VIEW?

Description Set or query the display mode of the event table.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<view>	Discrete	{PACKage|DETail|PAYLoad}	PACKage

Explanation

- PACKage: the time and data are displayed in the event table.
- DETail: the detailed data of the specified row is displayed in the event table.
- PAYLoad: all data of the specified column is displayed in the event table.

Return Format The query returns PACK, DET, or PAYL.

Example

```
:ETABle1:VIEW PAYLoad      /*Set the display mode of the event table to PAYLoad*/
:ETABle1:VIEW?             /*The query returns PAYL*/
```

:ETABle<n>:COLumn

Syntax :ETABle<n>:COLumn <col>

:ETABle<n>:COLumn?

Description Set or query the current column of the event table.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<col>	Discrete	{DATA|TX|RX|MISO|MOSI}	--

Explanation

- When different decoder is selected (:DECoder<n>:MODE), the range of <col> differs.

  Parallel decoding: DATA

  RS232 decoding: TX|RX

  I2C decoding: DATA

  SPI decoding: MISO|MOSI

- If the TX or RX channel source in RS232 decoding or the MISO or MOSI channel source in SPI decoding is set to OFF, <col> cannot be set to the corresponding parameter.

Return Format The query returns DATA, TX, RX, MISO, or MOSI.

Example

```
:ETABle1:COLumn DATA       /*Set the current column to DATA*/
:ETABle1:COLumn?           /*The query returns DATA*/
```

Related Commands

:DECoder<n>:UART:TX

:DECoder<n>:UART:RX

:DECoder<n>:SPI:MISO

:DECoder<n>:SPI:MOSI

:ETABle<n>:ROW

Syntax :ETABle<n>:ROW <row>

:ETABle<n>:ROW?

Description Set or query the current row of the event table.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<row>	Integer	1 to the maximum number of rows of the current event table	1

Return Format The query returns the current row in integer. If the current even table is empty, the query returns 0.

Example
```
:ETABle1:ROW 2      /*Set the current row to 2*/
:ETABle1:ROW?       /*The query returns 2*/
```

:ETABle<n>:SORT

Syntax :ETABle<n>:SORT <sort>

:ETABle<n>:SORT?

Description Set or query the display type of the decoding results in the event table.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--
<sort>	Discrete	{ASCend|DESCend}	ASCend

Explanation

- ASCend: the events are displayed in the order in which they occurred.
- DESCend: the events are displayed in the order reverse to the order in which they occurred.

Return Format The query returns ASC or DESC.

Example
```
:ETABle1:SORT DESCend      /*Set the display type of the event table to descend*/
:ETABle1:SORT?             /*The query returns DESC*/
```

:ETABle<n>:DATA?

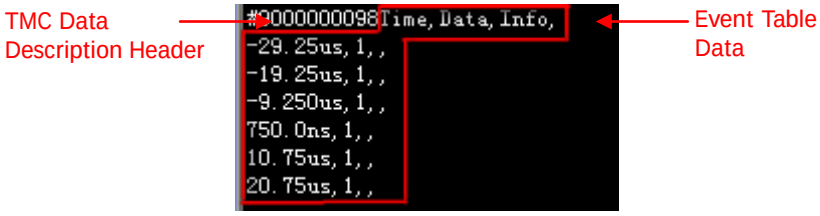
Syntax :ETABle<n>:DATA?

Description Read the current event table data.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	--

Return Format The query returns the event table data in the format as shown in the figure below. Wherein, #9000000098 is the TMC data description header followed by the even table data and its format is #900000dddd. dddd denotes the number of bytes of the valid waveform data following the description header. For example, as shown in the figure below, #9000000098 is the TMC data description header, wherein, 98 denotes that there are 98 bytes of valid data. The content following 98 is the event table data.



Example :ETABle1:DATA? /*The query returns the data as shown in the figure above*/

:FUNCtion Commands

The :FUNCtion commands are used to set the waveform recording and playback parameters.

Command List:

:FUNCtion:WRECord:FEND

Syntax :FUNCtion:WRECord:FEND <frame>

:FUNCtion:WRECord:FEND?

Description Set or query the end frame of waveform recording.

Parameter

Name	Type	Range	Default
<frame>	Integer	1 to the maximum number of frames can be recorded currently	5000

Explanation Use the :FUNCtion:WRECord:FMAX? command to query the maximum number of frames can be recorded currently.

Return Format The query returns the current end frame in integer.

Example

```
:FUNCtion:WRECord:FEND 4096     /*Set the end frame to 4096*/
:FUNCtion:WRECord:FEND?         /*The query returns 4096*/
```

:FUNCtion:WRECord:FMAX?

Syntax :FUNCtion:WRECord:FMAX?

Description Query the maximum number of frames can be recorded currently.

Explanation As the capacity of the waveform memory is fixed, the more the number of points each frame of waveform has, the less the number of waveform frames can be recorded. Thus, the maximum number of frames can be recorded currently is decided by the memory depth currently selected. The less the memory depth, the more the number of waveform frames can be recorded.

Return Format The query returns the maximum number of frames can be recorded currently in integer.

:FUNCtion:WRECord:FINTerval

Syntax :FUNCtion:WRECord:FINTerval <interval>

:FUNCtion:WRECord:FINTerval?

Description Set or query the time interval between frames in waveform recording. The default unit is s.

Parameter

Name	Type	Range	Default
<interval>	Real	100ns to 10s	100ns

Return Format The query returns the time interval currently set in scientific notation.

Example
```
:FUNCtion:WRECord:FINTerval 0.001      /*Set the time interval to 1ms*/
:FUNCtion:WRECord:FINTerval?           /*The query returns 1.000000e-03*/
```

:FUNCtion:WRECord:PROMpt

Syntax :FUNCtion:WRECord:PROMpt <bool>

:FUNCtion:WRECord:PROMpt?

Description Turn on or off the sound prompt when the recording finishes, or query the status of the sound prompt when the recording finishes.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	1|ON

Explanation When the sound prompt is turned on, the instrument exerts a sound promt when the recording finishes no matter whether the system sound (refer to :SYSTem:BEEPer) is turned on or not.

Return Format The query returns 1 or 0.

Example
```
:FUNCtion:WRECord:PROMpt 1      /*Turn on the sound prompt when the recording
                                  finishes*/
:FUNCtion:WRECord:PROMpt?       /*The query returns 1*/
```

:FUNCtion:WRECord:OPERate

Syntax :FUNCtion:WRECord:OPERate <opt>

:FUNCtion:WRECord:OPERate?

Description Start or stop the waveform recording, or query the status of the waveform recording.

Parameter

Name	Type	Range	Default
<opt>	Discrete	{RUN|STOP}	--

Explanation Before sending this command, send the :FUNCtion:WRECord:ENABle command to turn on the waveform recording function. Otherwise, this command is invalid.

Return Format The query returns RUN or STOP.

Example

```
:FUNCtion:WRECord:OPERate RUN     /*Start the waveform recording*/
:FUNCtion:WRECord:OPERate?        /*The query returns RUN if the recording is in progress and returns STOP if the recording has already finished*/
```

:FUNCtion:WRECord:ENABle

Syntax :FUNCtion:WRECord:ENABle <bool>

:FUNCtion:WRECord:ENABle?

Description Turn on or off the waveform recording function, or query the status of the waveform recording function.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation

- The waveform recording function can only be enabled when the horizontal timebase mode is "YT" and the horizontal timebase is lower than 200ms.
- After turning on the waveform recording function, **RUN / STOP** can be used to start or stop the waveform recording. At this point, you can send the :FUNCtion:WRECord:OPERate command to start the recording.

Return Format The query returns 1 or 0.

Example

```
:FUNCtion:WRECord:ENABle 1     /*Turn on the waveform recording function*/
:FUNCtion:WRECord:ENABle?      /*The query returns 1*/
```

:FUNCtion:WREPlay:FSTart

Syntax :FUNCtion:WREPlay:FSTart <sta>

:FUNCtion:WREPlay:FSTart?

Description Set or query the start frame of waveform playback.

Parameter

Name	Type	Range	Default
<sta>	Integer	1 to the maximum number of frames recorded	1

Explanation

- Use the :FUNCtion:WRECord:FEND command to set the maximum number of frames recorded.
- The start frame of waveform playback cannot be greater than the end frame of waveform playback (:FUNCtion:WREPlay:FEND).
- You can only set the start frame of waveform playback when a waveform is currently recorded.
- You cannot set the start frame of waveform playback during the waveform recording or playback process.

Return Format The query returns an integer.

Example

```
:FUNCtion:WREPlay:FSTart 5      /*Set the start frame of waveform playback to 5*/
:FUNCtion:WREPlay:FSTart?       /*The query returns 5*/
```

:FUNCtion:WREPlay:FEND

Syntax :FUNCtion:WREPlay:FEND <end>

:FUNCtion:WREPlay:FEND?

Description Set or query the end frame of waveform playback.

Parameter

Name	Type	Range	Default
<end>	Integer	1 to the maximum number of frames recorded	The maximum number of frames recorded

Explanation

- Use the :FUNCtion:WRECord:FEND command to set the maximum number of frames recorded.
- The end frame of waveform playback cannot be lower than the start frame of waveform playback (:FUNCtion:WREPlay:FSTart).
- You can only set the end frame of waveform playback when a waveform is currently recorded.
- You cannot set the end frame of waveform playback during the waveform recording or playback process.

Return Format The query returns an integer.

Example

```
:FUNCtion:WREPlay:FEND 4096      /*Set the end frame of waveform playback to 4096*/
:FUNCtion:WREPlay:FEND?          /*The query returns 4096*/
```

:FUNCtion:WREPlay:FMAX?

Syntax :FUNCtion:WREPlay:FMAX?

Description Query theb maximum number of frames can be played, namely the maximum number of frames recorded.

Explanation Use the :FUNCtion:WRECord:FEND command to set the maximum number of frames recorded.

Return Format The query returns an integer.

:FUNCtion:WREPlay:FINTerval

Syntax :FUNCtion:WREPlay:FINTerval <interval>

:FUNCtion:WREPlay:FINTerval?

Description Set or query the time interval between frames in waveform playback. The default unit is s.

Parameter

Name	Type	Range	Default
<interval>	Real	100ns to 10s	100ns

Explanation

- You can only set the time interval of waveform playback when a waveform is currently recorded.
- You cannot set the time interval of waveform playback during the waveform recording or playback process.

Return Format The query returns the current time interval in scientific notation.

Example

```
:FUNCtion:WREPlay:FINTerval 0.001      /*Set the time interval to 1ms*/
:FUNCtion:WREPlay:FINTerval?           /*The query returns 1.000000e-03*/
```

:FUNCtion:WREPlay:MODE

Syntax :FUNCtion:WREPlay:MODE <mode>

:FUNCtion:WREPlay:MODE?

Description Set or query the waveform playback mode.

Parameter

Name	Type	Range	Default
<mode>	Discrete	{REPeat|SINGle}	SINGle

Explanation

- REPeat: cycle playback. Play from the start frame to the end frame and then repeat until you stop it manually.
- SINGle: single playback. Play from the start frame to the end frame and then stop.
- You can only set the waveform playback mode when a waveform is currently recorded.
- You cannot set the waveform playback mode during the waveform recording or playback process.

Return Format The query returns REP or SING.

Example

```
:FUNCtion:WREPlay:MODE REPeat      /*Set the waveform playback mode to cycle*/
:FUNCtion:WREPlay:MODE?            /*The query returns REP*/
```

:FUNCtion:WREPlay:DIRection

Syntax :FUNCtion:WREPlay:DIRection <dir>

:FUNCtion:WREPlay:DIRection?

Description Set or query the waveform playback direction.

Parameter

Name	Type	Range	Default
<dir>	Discrete	{FORWard|BACKward}	FORWard

Explanation

- FORWard: positive direction. Play from the start frame to the end frame.
- BACKward: negative direction. Play from the end frame to the start frame.
- You can only set the waveform playback direction when a waveform is currently recorded.
- You cannot set the waveform playback direction during the waveform recording or playback process.

Return Format The query returns FORW or BACK.

Example

```
:FUNCtion:WREPlay:DIRection FORWard    /*Set the waveform playback direction to forward*/
:FUNCtion:WREPlay:DIRection?           /*The query returns FORW*/
```

:FUNCtion:WREPlay:OPERate

Syntax :FUNCtion:WREPlay:OPERate <opt>

:FUNCtion:WREPlay:OPERate?

Description Start, pause, or stop the waveform playback, or query the status of the waveform playback.

Parameter

Name	Type	Range	Default
<opt>	Discrete	{PLAY|PAUSe|STOP}	STOP

Explanation This command is only valid when waveform has already been recorded.

Return Format The query returns PLAY, PAUS, or STOP.

Example

```
:FUNCtion:WREPlay:OPERate PLAY    /*Start the waveform playback*/
:FUNCtion:WREPlay:OPERate?        /*The query returns PLAY if the playback is in progress and returns STOP if the playback finishes*/
```

Related Command :FUNCtion:WRECord:OPERate

:FUNCtion:WREPlay:FCURrent

Syntax :FUNCtion:WREPlay:FCURrent <cur>

:FUNCtion:WREPlay:FCURrent?

Description Set or query the current frame in waveform playback.

Parameter

Name	Type	Range	Default
<cur>	Integer	1 to the maximum number of frames recorded	The maximum number of frames recorded

Explanation

- Use the :FUNCtion:WRECord:FEND command to set the maximum number of frames recorded.
- You can only set the current frame of waveform playback when a waveform is currently recorded.
- You cannot set the current frame of waveform playback during the waveform recording or playback process.

Return Format The query returns an integer.

Example

```
:FUNCtion:WREPlay:FCURrent 300      /*Set the current frame in waveform playback to 300*/
:FUNCtion:WREPlay:FCURrent?         /*The query returns 300*/
```

IEEE488.2 Common Commands

The IEEE 488.2 standard defines some common commands used for querying the basic information of the instrument or executing the basic operations. These commands usually start with "*" and the keyword of the command is usually 3-character long.

Command List:

*CLS

Syntax *CLS

Description Clear all the event registers and clear the error queue.

*ESE

Syntax *ESE <value>

*ESE?

Description Set or query the enable register for the standard event status register set.

Parameter

Name	Type	Range	Default
<value>	Integer	Refer to **Explanation**	0

Explanation The bit 1 and bit 6 of the standard event status register are not used and are always treated as 0; therefore, the range of <value> are the decimal numbers corresponding to the binary numbers X0XXXX0X (X is 1 or 0).

Return Format The query returns an integer which equals the sum of the weights of all the bits that have already been set in the register.

Example

```
*ESE 16      /*Enable the bit 4 (16 in decimal) of the standard event status register*/
*ESE?        /*The query returns 16*/
```

*ESR?

Syntax *ESR?

Description Query and clear the event register for the standard event status register.

Explanation The bit 1 and bit 6 of the standard event status register are not used and are always treated as 0. The range of the return value are the decimal numbers corresponding to the binary numbers X0XXXX0X (X is 1 or 0).

Return Format The query returns an integer which equals the sum of the weights of all the bits in the register.

*IDN?

Syntax *IDN?

Description Query the ID string of the instrument.

Return Format The query returns RIGOL TECHNOLOGIES,<model>,<serial number>,<software version>.
Wherein,
<model>: the model number of the instrument.
<serial number>: the serial number of the instrument.
<software version>: the software version of the instrument.

*OPC

Syntax *OPC

*OPC?

Description The *OPC command is used to set the Operation Complete bit (bit 0) in the standard event status register to 1 after the current operation is finished. The *OPC? command is used to query whether the current operation is finished.

Return Format The query returns 1 if the current operation is finished; otherwise, returns 0.

*RST

Syntax *RST

Description Restore the instrument to the default state.

*SRE

Syntax *SRE <value>

*SRE?

Description Set or query the enable register for the status byte register set.

Parameter

Name	Type	Range	Default
<value>	Integer	0 to 255	0

Explanation The bit 0 and bit 1 of the status byte register are not used and are always treated as 0; therefore, the range of <value> are the decimal numbers corresponding to the binary numbers XXXXXX00 (X is 1 or 0).

Return Format The query returns an integer which equals the sum of the weights of all the bits that have already been set in the register.

Example
```
*SRE 16      /*Enable the bit 4 (16 in decimal) of the status byte register*/
*SRE?        /*The query returns 16*/
```

*STB?

Syntax *STB?

Description Query the event register for the status byte register. The value of the status byte register is set to 0 after this command is executed.

Explanation The bit 0 and bit 1 of the status byte register are not used and are always treated as 0. The query returns the decimal numbers corresponding to the binary numbers X0XXXX0X (X is 1 or 0).

Return Format The query returns an integer which equals the sum of the weights of all the bits in the register.

*TST?

Syntax *TST?

Description Perform a self-test and then return the seilf-test results.

Return Format The query returns a decimal integer.

*WAI

Syntax *WAI

Description Wait for the operation to finish.

Explanation The subsequent command can only be carried out after the current command has been executed.

:LAN Commands

Command List:

:LAN:DHCP

Syntax :LAN:DHCP <bool>

:LAN:DHCP?

Description Turns on or off the DHCP configuration mode; or queries the on/off status of the current DHCP configuration mode.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	1|ON

Explanation

- When the three IP configuration types (DHCP, Auto IP, and Static IP) are all turned on, the priority of the parameter configuration from high to low is "DHCP", "Auto IP", and "Static IP". The three IP configuration types cannot be all turned off at the same time.
- When DHPC is valid, the DHCP server in the current network will assign the network parameters (such as the IP address) for the oscilloscope.

Return Format The query returns 1 or 0.

Example

```
:LAN:DHCP ON          /*Enables DHCP configuration mode.*/
:LAN:DHCP?            /*The query returns 1.*/
```

:LAN:AUToip

Syntax :LAN:AUToip <bool>

:LAN:AUToip?

Description Turns on or off the Auto IP configuration mode; or queries the on/off status of the current Auto IP configuration mode.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	1|ON

Explanation

- When the three IP configuration types (DHCP, Auto IP, and Static IP) are all turned on, the priority of the parameter configuration from high to low is "DHCP", "Auto IP", and "Static IP". The three IP configuration types cannot be all turned off at the same time.
- When the auto IP mode is valid, disable DHCP manually. You can self-define the gateway and DNS address for the oscilloscope.
- For the gateway setting, refer to the :LAN:GATeway command. For the DNS setting, refer to the :LAN:DNS command.

Return Format The query returns 1 or 0.

Example

```
:LAN:AUToip ON          /*Disables the Auto IP configuration mode.*/
:LAN:AUToip?            /*The query returns 1.*/
```

:LAN:GATeway

Syntax :LAN:GATeway <string>

:LAN:GATeway?

Description Sets or queries the default gateway.

Parameter

Name	Type	Range	Default
<string>	ASCII String	Refer to **Explanation**	——

Explanation

- The format of <string> is nnn,nnn,nnn,nnn. The range of the first section of "nnn" is from 0 to 223 (except 127), and the ranges of the other three sections of "nnn" are from 0 to 255.
- When you use this command, the IP configuration mode should be Auto IP (refer to the :LAN:AUToip command) or Static IP mode (refer to the :LAN:MANual command).

Return Format The query returns the current gateway in strings.

Example

```
:LAN:GATeway 192.168.1.1        /*Sets the default gateway to 192.168.1.1.*/
:LAN:GATeway?                   /*The query returns the current gateway.*/
```

:LAN:DNS

Syntax :LAN:DNS <string>

:LAN:DNS?

Description Sets or queries the DNS address.

Parameter

Name	Type	Range	Default
<string>	ASCII String	Refer to **Explanation**	——

Explanation

- The format of <string> is nnn,nnn,nnn,nnn. The range of the first section of "nnn" is from 0 to 223 (except 127), and the ranges of the other three sections of "nnn" are from 0 to 255.
- When you use this command, the IP configuration mode should be Auto IP (refer to the :LAN:AUToip command) or Static IP mode (refer to the :LAN:MANual command).

Return Format The query returns the current DNS address in strings.

Example
```
:LAN:DNS 192.168.1.1        /*Sets the DNS address to 192.168.1.1.*/
:LAN:DNS?                   /*The query returns the current DNS address.*/
```

:LAN:MAC?

Syntax :LAN:MAC?

Description Queries the MAC address of the instrument. The address format displayed on the oscilloscope is as follows: 00-19-AF-30-00-00.

Return Format The query returns the MAC address in strings. For example, 0019af300000.

:LAN:MANual

Syntax :LAN:MANual <bool>

:LAN:MANual?

Description Turns on or off the static IP configuration mode; or queries the on/off status of the static IP configuration mode.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation

- When the three IP configuration types (DHCP, Auto IP, and Static IP) are all turned on, the priority of the parameter configuration from high to low is "DHCP", "Auto IP", and "Static IP". The three IP configuration types cannot be all turned off at the same time.
- When the static IP mode is valid, disable DHCP and Auto IP manually. You can self-define the network parameters of the oscilloscope, such as IP address, subnet mask, gateway, and DNS address. For the setting of the IP address, refer to the :LAN:IPADdress command. For the setting of the subnet mask, refer to the :LAN:SMASk command. For the setting of the gateway, refer to the :LAN:GATeway command. For the setting of DNS, refer to the :LAN:DNS command.

Return Format The query returns 1 or 0.

Example
```
:LAN:MANual ON              /*Enables the static IP configuration mode.*/
:LAN:MANual?                /*The query returns 1.*/
```

:LAN:INITiate

Syntax :LAN:INITiate

Description Initiates the network parameters. Before running the command, confirm that the oscilloscope has been connected to the network properly.

:LAN:IPADdress

Syntax :LAN:IPADdress <string>

:LAN:IPADdress?

Description Sets or queries the IP address of the instrument.

Parameter

Name	Type	Range	Default
<string>	ASCII String	Refer to **Explanation**	——

Explanation

- The format of <string> is nnn,nnn,nnn,nnn. The range of the first section of "nnn" is from 0 to 223 (except 127), and the ranges of the other three sections of "nnn" are from 0 to 255.
- When you use the command, the IP configuration mode should be static IP (refer to the :LAN:MANual command). Besides, the DHCP and auto IP should be disabled.

Return Format The query returns the current IP address in strings.

Example

```
:LAN:IPADdress 192.168.1.10        /*Sets the IP address to 192.168.1.10.*/
:LAN:IPADdress?                    /*The query returns the current IP address.*/
```

:LAN:SMASk

Syntax :LAN:SMASk <string>

:LAN:SMASk?

Description Sets or queries the subnet mask.

Parameter

Name	Type	Range	Default
<string>	ASCII String	Refer to **Explanation**	——

Explanation

- The format of <string> is nnn,nnn,nnn,nnn. The range of the section "nnn" is from 0 to 255.
- When you use the command, the IP configuration mode should be static IP (refer to the :LAN:MANual command). The DHCP and auto IP should be disabled.

Return Format The query returns the current subnet mask in strings.

Example

```
:LAN:SMASk 255.255.255.0           /*Sets the subnet mask to 255.255.255.0*/
:LAN:SMASk?                        /*The query returns the current subnet mask.*/
```

:LAN:STATus?

Syntax :LAN:STATus?

Description Queries the current network configuration status.

Return Format The query returns UNLINK, INIT, IPCONFLICT, CONFIGURED, or DHCPFAILED.
UNLINK: not connected.
INIT: the instrument is acquiring an IP address.
IPCONFLICT: there is an IP address conflict.
CONFIGURED: the network configuration has been successfully configured.
DHCPFAILED: the DHCP configuration has failed.

:LAN:VISA?

Syntax :LAN:VISA?

Description Queries the VISA address of the instrument.

Return Format The query returns the VISA address in strings. For example, TCPIP::172.16.3.119::INSTR.

:LAN:APPLy

Syntax :LAN:APPLy

Description Applies the network configuration.

:MATH Commands

The :MATH commands are used to set the operations between the waveforms of multiple channels.

Note:

✓ The operations include the following types:
Algebraic Operations: A+B, A-B, AxB, A/B
Spectrum Operation: FFT
Logic Operations: A&&B, A||B, A^B, !A
Functional Operations: Intg, Diff, Sqrt, Lg, Ln, Exp, Abs
Filter: Low Pass Filter, High Pass Filter, Band Pass Filter, Band Stop Filter
Compound Operations: Combination of two operations (inner and outer)

✓ For logic operations, the waveform data to be operated is compared with the preset threshold and is converted to 0 or 1. Thus, the result will also be 0 or 1.

✓ For a relatively complicated operation, you can split it into inner and outer layer operations (namely compound operation) according to your need. The inner layer operation (fx) can only be algebraic operation and the outer layer operation can only be algebraic operation or functional operation.

When the outer layer operation is algeriac operation, at least one of source A and source B of the outer layer operation should be set to FX.
When the outer layer operation is functional operation, the source of the outer layer operation can only be set to FX.

Note: For the inner layer operation (fx), you can send the :MATH:OPTion:FX:SOURce1, :MATH:OPTion:FX:SOURce2, and :MATH:OPTion:FX:OPERator commands to set the sources and operator of the inner layer operation. For the outer layer operation, you can send the :MATH:SOURce1, :MATH:SOURce2, and :MATH:OPERatorcommands to set the sources and operator of the outer layer operation.

Command List:

- :MATH:DISPlay
- :MATH:OPERator
- :MATH:SOURce1
- :MATH:SOURce2
- :MATH:LSOUrce1
- :MATH:LSOUrce2
- :MATH:SCALe
- :MATH:OFFSet
- :MATH:INVert
- :MATH:RESet
- :MATH:FFT:SOURce
- :MATH:FFT:WINDow
- :MATH:FFT:SPLit
- :MATH:FFT:UNIT
- :MATH:FFT:HSCale
- :MATH:FFT:HCENter
- :MATH:FFT:MODE

:MATH:DISPlay

Syntax :MATH:DISPlay <bool>

:MATH:DISPlay?

Description Enable or disable the math operation function or query the math operation status.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Return Format The query returns 1 or 0.

Example
```
:MATH:DISPlay ON      /*Enable the math operation function*/
:MATH:DISPlay?        /*The query returns 1*/
```

:MATH:OPERator

Syntax :MATH:OPERator <opt>

:MATH:OPERator?

Description Set or query the operator of the math operation.

Parameter

Name	Type	Range	Default
<opt>	Discrete	{ADD|SUBTract|MULTiply|DIVision|AND|OR|XOR|NOT|FFT|INTG|DIFF|SQRT|LOG|LN|EXP|ABS|FILTer}	ADD

Explanation When the parameter in :MATH:SOURce1 and/or :MATH:SOURce2 is FX, this command is used to set the operator of the outer layer operation of compound operation. The range of <opt> is {ADD|SUBTract|MULTiply|DIVision|INTG|DIFF|SQRT|LOG|LN|EXP|ABS}.

Return Format The query returns ADD, SUBT, MULT, DIV, AND, OR, XOR, NOT, FFT, INTG, DIFF, SQRT, LOG, LN, EXP, ABS, or FILT.

Example
```
:MATH:OPERator INTG     /*Set the operator of the math operation to integration*/
:MATH:OPERator?         /*The query returns INTG*/
```

:MATH:SOURce1

Syntax :MATH:SOURce1 <src>

:MATH:SOURce1?

Description Set or query the source or source A of algebraic operation/functional operation/the outer layer operation of compound operation.

Parameter

Name	Type	Range	Default
<src>	Discrete	{CHANnel1|CHANnel2|FX}	CHANnel1

Explanation

- For algebraic operations, this command is used to set source A.
- For functional operations, only this command is used to set the source.
- For compound operations, this command is used to set source A of the outer layer operation when the outer layer operation is algeriac operation and the range of <src> is {CHANnel1|CHANnel2|FX}; this command is used to set the source of the outer layer operation when the outer layer operation is functional operation and <src> can only be set to FX.

  Note: When the outer layer operation of compound operation is algebraic operation, at least one of source A and source B of the outer layer operation should be set to FX.
- When "FX" is selected, you can send the :MATH:OPTion:FX:SOURce1, :MATH:OPTion:FX:SOURce2, and :MATH:OPTion:FX:OPERator commands to set the sources and operator of the inner layer operation.

Return Format The query returns CHAN1, CHAN2, or FX.

Example

```
:MATH:SOURce1 CHANnel2      /*Set source A of algebraic operation to CH2*/
:MATH:SOURce1?              /*The query returns CHAN2*/
```

:MATH:SOURce2

Syntax :MATH:SOURce2 <src>

:MATH:SOURce2?

Description Set or query source B of algebraic operation/the outer layer operation of compound operation.

Parameter

Name	Type	Range	Default
<src>	Discrete	{CHANnel1|CHANnel2|FX}	CHANnel1

Explanation

- This command is only applicable to algebraic operations (requiring two sources) and compound operations whose outer layer operations are algebraic operations.
- When the outlayer operation of compound operation is algebraic operation, this command is used to set source B of the outer layer operation.

  Note: When the outer layer operation of compound operation is algebraic operation, at least one of source A and source B of the outer layer operation should be set to FX.
- When "FX" is selected, you can send the :MATH:OPTion:FX:SOURce1, :MATH:OPTion:FX:SOURce2, and :MATH:OPTion:FX:OPERator commands to set the sources and operator of the inner layer operation.

Return Format The query returns CHAN1, CHAN2, or FX.

Example
```
:MATH:SOURce2 CHANnel2      /*Set source B of algebraic operation to CH2*/
:MATH:SOURce2?              /*The query returns CHAN2*/
```

:MATH:LSOUrce1

Syntax :MATH:LSOUrce1 <src>

:MATH:LSOUrce1?

Description Set or query source A of logic operation.

Parameter

Name	Type	Range	Default
<src>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Explanation The logic operations include A&&B, A||B, A^B, and !A.

Return Format The query returns CHAN1 or CHAN2.

Example
```
:MATH:LSOUrce1 CHANnel1     /*Set source A of logic operation to CH1*/
:MATH:LSOUrce1?             /*The query returns CHAN1*/
```

:MATH:LSOUrce2

Syntax :MATH:LSOUrce2 <src>

:MATH:LSOUrce2?

Description Set or query source B of logic operation.

Parameter

Name	Type	Range	Default
<src>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Explanation

- The logic operations include A&&B, A||B, A^B, and !A.
- This command is only applicable to logic operations that require two signal sources and is used to set source B.

Return Format The query returns CHAN1 or CHAN2.

Example
```
:MATH:LSOUrce2 CHANnel1     /*Set source B of logic operation to CH1*/
:MATH:LSOUrce2?             /*The query returns CHAN1*/
```

:MATH:SCALe

Syntax :MATH:SCALe <scale>

:MATH:SCALe?

Description Set or query the vertical scale of the operation result. The unit depends on the operator currently selected and the unit of the source.

Parameter

Name	Type	Range	Default
<scale>	Real	The max range is from 1p to 5T (in 1-2-5 step)	1.00V

Explanation The range of the vertical scale is related to the operator currently selected and the vertical scale of the source channel. For the integration (intg) and differential (diff) operations, it is also related to the current horizontal timebase.

Return Format The query returns the vertical scale of the operation result in scientific notation.

Example
```
:MATH:SCALe 2      /*Set the vertical scale to 2V*/
:MATH:SCALe?       /*The query returns 2.000000e+00*/
```

Related Commands :CHANnel<n>:SCALe

:CHANnel<n>:UNITs

:MATH:OPERator

:TIMebase[:MAIN]:SCALe

:MATH:OFFSet

Syntax :MATH:OFFSet <offs>

:MATH:OFFSet?

Description Set or query the vertical offset of the operation result. The unit depends on the operator currently selected and the unit of the source.

Parameter

Name	Type	Range	Default
<offs>	Real	Related to the vertical scale of the operation result Range: (-1000 x MathVerticalScale) to (1000 x MathVerticalScale) Step: MathVerticalScale/50	0.00V

Explanation MathVerticalScale is the vertical scale of the operation result and can be set by the :MATH:SCALe command.

Return Format The query returns the vertical offset of the operation result in scientific notation.

Example
```
:MATH:OFFSet 8     /*Set the vertical offset to 8V*/
:MATH:OFFSet?      /*The query returns 8.000000e+00*/
```

Related Commands :MATH:OPERator

:CHANnel<n>:UNITs

:MATH:INVert

Syntax :MATH:INVert <bool>

:MATH:INVert?

Description Enable or disable the inverted display mode of the operation result, or query the inverted display mode status of the operation result.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation

- This command is invalid for the FFT operation.
- This command has the same function as the :MATH:OPTion:INVert command.

Return Format The query returns 1 or 0.

Example
```
:MATH:INVert ON      /*Enable the inverted display mode*/
:MATH:INVert?        /*The query returns 1*/
```

Related Command :MATH:OPERator

:MATH:RESet

Syntax :MATH:RESet

Description Sending this command, the instrument adjusts the vertical scale of the operation result to the most proper value according to the current operator and the horizontal timebase of the source.

Related Commands :MATH:OPERator

:MATH:SCALe

:TIMebase[:MAIN]:SCALe

:MATH:FFT:SOURce

Syntax :MATH:FFT:SOURce <src>

:MATH:FFT:SOURce?

Description Set or query the source of FFT operation/filter.

Parameter

Name	Type	Range	Default
<src>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example
```
:MATH:FFT:SOURce CHANnel1     /* Set the source of FFT operation to CH1*/
:MATH:FFT:SOURce?             /* The query returns CHAN1*/
```

Related Command :MATH:OPERator

:MATH:FFT:WINDow

Syntax :MATH:FFT:WINDow <wnd>

:MATH:FFT:WINDow?

Description Set or query the window function of the FFT operation.

Parameter

Name	Type	Range	Default
<wnd>	Discrete	{RECTangle|BLACkman|HANNing|HAMMing|FLATtop|TRIangle}	RECTangle

Explanation

- Spectral leakage can be considerably decreased when a window function is used.
- Different window functions are applicable to measure different waveforms. You need to select the window function according to waveform to be measured and its characteristics.

Return Format The query returns RECT, BLAC, HANN, HAMM, FLAT, or TRI.

Example
```
:MATH:FFT:WINDow BLACkman      /*Set the window function of the FFT operation to
                                blackman*/
:MATH:FFT:WINDow?              /* The query returns BLACK*/
```

Related Command :MATH:OPERator

:MATH:FFT:SPLit

Syntax
```
:MATH:FFT:SPLit <bool>
:MATH:FFT:SPLit?
```

Description Enable or disable the half-screen display mode of the FFT operation, or query the status of the half display mode of the FFT operation.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	1|ON

Explanation

- Enable the half-screen display mode: the source channel and the FFT operation results are displayed separately. The time domain and frequency domain signals are displayed clearly.
- Disable the half-screen display mode (full-screen display mode): the source channel and the FFT operation results are displayed in the same window to view the frequency spectrum more clearly and to perform more precise measurement.

Return Format The query returns 1 or 0.

Example
```
:MATH:FFT:SPLit OFF      /*Disable the half-screen display mode*/
:MATH:FFT:SPLit?         /*The query returns 0*/
```

Related Command :MATH:OPERator

:MATH:FFT:UNIT

Syntax
```
:MATH:FFT:UNIT <unit>
:MATH:FFT:UNIT?
```

Description Set or query the vertical unit of the FFT operation result.

Parameter

Name	Type	Range	Default
<unit>	Discrete	{VRMS|DB}	DB

Return Format The query returns VRMS or DB.

Example
```
:MATH:FFT:UNIT VRMS      /*Set the vertical unit of the FFT operation result to Vrms*/
:MATH:FFT:UNIT?          /*The query returns VRMS*/
```

Related Command :MATH:OPERator

:MATH:FFT:HSCale

Syntax :MATH:FFT:HSCale <hsc>

:MATH:FFT:HSCale?

Description Set or query the horizontal scale of the FFT operation result. The default unit is Hz.

Parameter

Name	Type	Range	Default
<hsc>	Real	Refer to **Explanation**	5MHz

Explanation

- <hsc> can be set to 1/1000, 1/400, 1/200, 1/100, 1/40, or 1/20 of the FFT sample rate.
- When the FFT mode is set to TRACe, FFT sample rate equals to screen sample rate, that is, 100/horizontal timebase.
  When the FFT mode is set to MEMory, FFT sample rate equals to memory sample rate (:ACQuire:SRATe?).
- You can view the detailed information of the frequency spectrum by reducing the horizontal scale.

Return Format The query returns the horizontal scale in scientific notation.

Example

```
:MATH:FFT:HSCale 500000     /*Set the horizontal scale of the FFT operation result to 500kHz*/
:MATH:FFT:HSCale?           /*The query returns 5.000000e+05*/
```

Related Commands :MATH:OPERator

:MATH:FFT:MODE

:TIMebase[:MAIN]:SCALe

:MATH:FFT:HCENter

Syntax :MATH:FFT:HCENter <cent>

:MATH:FFT:HCENter?

Description Set or query the center frequency of the FFT operation result, namely the frequency relative to the horizontal center of the screen. The default unit is Hz.

Parameter

Name	Type	Range	Default
<cent>	Real	Refer to **Explanation**	5MHz

Explanation

- When the FFT mode is set to TRACe, the range of <cent> is from 0 to (0.4 x FFT sample rate). Wherein, FFT sample rate equals to screen sample rate, that is, 100/horizontal timebase.
  When the FFT mode is set to MEMory, the range of <cent> is from 0 to (0.5 x FFT sample rate). Wherein, FFT sample rate equals to memory sample rate (:ACQuire:SRATe?).
- Step = Horizontal Scale of the FFT operation result/50.

Return Format The query returns the current center frequency in scientific notation.

Example

```
:MATH:FFT:HCENter 10000000     /*Set the center frequency of the FFT operation result to 10MHz*/
:MATH:FFT:HCENter?             /*The query returns 1.000000e+07*/
```

Related Commands :MATH:OPERator

:MATH:FFT:MODE

:TIMebase[:MAIN]:SCALe

:MATH:FFT:HSCale

:MATH:FFT:MODE

Syntax :MATH:FFT:MODE <mode>

:MATH:FFT:MODE?

Description Set or query the FFT mode.

Parameter

Name	Type	Range	Default
<mode>	Discrete	{TRACe|MEMory}	TRACe

Explanation

- TRACe: denotes that the data source of the FFT operation is the data of the waveform displayed on the screen.
- MEMory: denotes that the data source of the FFT operation is the data of the waveform in the memory.

Return Format The query returns TRAC or MEM.

Example

```
:MATH:FFT:MODE TRACe        /*Set the FFT mode to TRACe*/
:MATH:FFT:MODE?             /* The query returns TRAC*/
```

:MATH:FILTer:TYPE

Syntax :MATH:FILTer:TYPE <type>

:MATH:FILTer:TYPE?

Description Set or query the filter type.

Parameter

Name	Type	Range	Default
<type>	Discrete	{LPASs|HPASs|BPASs|BSTOP}	LPASs

Explanation

- DS1000Z-E provides 4 types of filters (Low Pass Filter, High Pass Filter, Band Pass Filter, and Band Stop Filter). Setting the bandwidth can filter the specified frequencies. You can send the :MATH:FFT:SOURce command to set or query the source of the filter.
- LPASs: low pass, which indicates that only the signals whose frequencies are lower than the current cutoff frequency (**ωc1**) can pass the filter.
- HPASs: high pass, which indicates that only the signals whose frequencies are greater than the current cutoff frequency (**ωc1**) can pass the filter.
- BPASs: band pass, which indicates that only the signals whose frequencies are greater than current cutoff frequency 1 (**ωc1**) and lower than current cutoff frequency 2 (**ωc2**) can pass the filter.
  Note: Cutoff frequency 1 (**ωc1**) must be lower than cutoff frequency 2 (**ωc2**).
- BSTOP: band stop, which indicates that only the signals whose frequencies are lower than current cutoff frequency 1 (**ωc1**) or greater than current cutoff frequency 2 (**ωc2**) can pass the filter.
  Note: Cutoff frequency 1 (**ωc1**) must be lower than cutoff frequency 2 (**ωc2**).

Return Format The query returns LPAS, HPAS, BPAS, or BSTOP.

Example
```
:MATH:FILTer:TYPE LPASs      /*Set the filter type to low pass*/
:MATH:FILTer:TYPE?           /*The query returns LPAS*/
```

Related Commands :MATH:FILTer:W1

:MATH:FILTer:W2

:MATH:FILTer:W1

Syntax :MATH:FILTer:W1 <freq1>

:MATH:FILTer:W1?

Description Set or query the cutoff frequency (**ωc1**) of the low pass/high pass filter or cutoff frequency 1 (**ωc1**) of the band pass/band stop filter. The default unit is Hz.

Parameter

Name	Type	Range	Default
<freq1>	Real	Refer to **Explanation**	Related to the filter type LPASs|BPASs|BSTOP: 0.005 x screen sample rate HPASs: 0.1 x screen sample rate

Explanation

- When the type of the filter is set to LPASs (low pass) or HPASs (high pass), you need to set one cutoff frequency (**ωc1**). At this time, the range of <freq1> is from (0.005 x screen sample rate) to (0.1 x screen sample rate), with a step of (0.005 x screen sample rate). Wherein, Screen Sample Rate = 100/Horizontal Timebase.
- When the type of the filter is set to BPASs (band pass) or BSTOP (band stop), you need to set two cutoff frequencies. Send this command to set cutoff frequency 1 (**ωc1**); and send the :MATH:FILTer:W2 command to set cutoff frequency 2 (**ωc2**). At this time, the range of <freq1> is from (0.005 x screen sample rate) to (0.095 x screen sample rate), with a step of (0.005 x screen sample rate). Wherein, Screen Sample Rate = 100/Horizontal Timebase.
  Note: Cutoff frequency 1 (**ωc1**) must be lower than cutoff frequency 2 (**ωc2**).

Return Format The query returns the current cutoff frequency or cutoff frequency 1 in scientific notation.

Example
```
:MATH:FILTer:W1 1000000      /*Set the cutoff frequency of the low pass filter to
                              1MHz*/
:MATH:FILTer:W1?             /*The query returns 1.000000e+06*/
```

Related Commands :MATH:FILTer:TYPE

:TIMebase[:MAIN]:SCALe

:MATH:FILTer:W2

Syntax :MATH:FILTer:W2 <freq2>

:MATH:FILTer:W2?

Description Set or query the cutoff frequency 2 (**ωc2**) of the band pass/band stop filter. The default unit is Hz.

Parameter

Name	Type	Range	Default
<freq2>	Real	Refer to **Explanation**	0.1 x screen sample rate

Explanation When the type of the filter is set to BPASs (band pass) or BSTOP (band stop), you need to set two cutoff frequencies. Send the :MATH:FILTer:W1 command to set cutoff frequency 1 (**ωc1**); and send this command to set cutoff frequency 2 (**ωc2**). At this time, the range of <freq2> is from (0.01 x screen sample rate) to (0.1 x screen sample rate), with a step of (0.005 x screen sample rate). Wherein, Screen Sample Rate = 100/Horizontal Timebase.
Note: Cutoff frequency 2 (**ωc2**) must be greater than cutoff frequency 1 (**ωc1**).

Return Format The query returns current cutoff frequency 2 in scientific notation.

Example
```
:MATH:FILTer:W2 1500000      /*Set cutoff frequency 2 of the band pass filter to 1.5MHz*/
:MATH:FILTer:W2?             /*The query returns 1.500000e+06*/
```

Related Commands :MATH:FILTer:TYPE
:TIMebase[:MAIN]:SCALe

:MATH:OPTion:STARt

Syntax :MATH:OPTion:STARt <sta>
:MATH:OPTion:STARt?

Description Set or query the start point of the waveform math operation.

Parameter

Name	Type	Range	Default
<sta>	Integer	0 to (End point currently set - 1)	0

Explanation
- This command is invalid for the FFT operation. Sending this command will modify the start points of all the operations (except FFT).
- The source selected is equally divided into 1200 parts horizontally, in which the leftmost is 0 and the rightmost is 1199.

Return Format The query returns an integer.

Example
```
:MATH:OPTion:STARt 50        /*Set the waveform start point to 50*/
:MATH:OPTion:STARt?          /*The query returns 50*/
```

Related Command :MATH:OPTion:END

:MATH:OPTion:END

Syntax :MATH:OPTion:END <end>
:MATH:OPTion:END?

Description Set or query the end point of the waveform math operation.

Parameter

Name	Type	Range	Default
<end>	Discrete	(Start point currently set+1) to 1199	1199

Explanation
- This command is invalid for the FFT operation. Sending this command will modify the end points of all the operations (except FFT).
- The source selected is equally divided into 1200 parts horizontally, in which the leftmost is 0 and the rightmost is 1199.

Return Format The query returns an integer.

Example
```
:MATH:OPTion:END 200      /*Set the waveform end point to 200*/
:MATH:OPTion:END?         /*The query returns 200*/
```

Related Command :MATH:OPTion:STARt

:MATH:OPTion:INVert

Syntax :MATH:OPTion:INVert <bool>

:MATH:OPTion:INVert?

Description Enable or disable the inverted display mode of the operation result, or query the inverted display mode status of the operation result.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation

- This command is invalid for the FFT operation. Sending this command will modify the inverted display mode status of all the operations (except FFT).
- This command has the same function as the :MATH:INVert command.

Return Format The query returns 1 or 0.

Example
```
:MATH:OPTion:INVert ON    /*Enable the inverted display mode*/
:MATH:OPTion:INVert?      /*The query returns 1*/
```

:MATH:OPTion:SENSitivity

Syntax :MATH:OPTion:SENSitivity <sens>

:MATH:OPTion:SENSitivity?

Description Set or query the sensitivity of the logic operation. The default unit is div (namely the current vertical scale).

Parameter

Name	Type	Range	Default
<sens>	Real	0 to 0.96, the step is 0.08	0

Explanation This command is only applicable to logic operations (A&&B, A||B, A^B, and !A).

Return Format The query returns the current sensitivity in scientific notation.

Example
```
:MATH:OPTion:SENSitivity 0.24   /*Set the sensitivity of the logic operation to 0.24div*/
:MATH:OPTion:SENSitivity?       /*The query returns 2.400000e-01*/
```

Related Commands :CHANnel<n>:SCALe

:MATH:OPERator

:MATH:OPTion:DIStance

Syntax :MATH:OPTion:DIStance <dist>

:MATH:OPTion:DIStance?

Description Set or query the smoothing window width of differential operation (diff).

Parameter

Name	Type	Range	Default
<dist>	Integer	3 to 201	3

Explanation This command is only applicable to differential operation (diff).

Return Format The query returns an integer between 3 and 201.

Example
```
:MATH:OPTion:DIStance 25      /*Set the smoothing window width to 25*/
:MATH:OPTion:DIStance?        /*The query returns 25*/
```

Related Command :MATH:OPERator

:MATH:OPTion:ASCale

Syntax :MATH:OPTion:ASCale <bool>

:MATH:OPTion:ASCale?

Description Enable or disable the auto scale setting of the operation result or query the status of the auto scale setting.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation

- When the auto scale is enabled, the instrument will automatically calculate the vertical scale range according to the current operator, the vertical scale and the horizontal timebase. If the current scale is out of the range, it will adjust the vertical scale to the best value automatically.
- Sending this command will modify the auto scale status of all the operation results.

Return Format The query returns 1 or 0.

Example
```
:MATH:OPTion:ASCale ON      /*Enable the auto scale setting */
:MATH:OPTion:ASCale?        /*The query returns 1*/
```

:MATH:OPTion:THReshold1

Syntax :MATH:OPTion:THReshold1 <thre>

:MATH:OPTion:THReshold1?

Description Set or query the threshold level of source A in logic operations. The default unit is V.

Parameter

Name	Type	Range	Default
<thre>	Real	(-4 x VerticalScale - VerticalOffset) to (4 x VerticalScale - VerticalOffset)	0.00V

Explanation

- This command is only applicable to A&&B, A||B, A^B, and !A logic operations of which source A is an analog channel.

- VerticalScale refers to the vertical scale of source A
  VeticalOffset refers to the vertical offset of source A
  The step is VerticalScale/50

Return Format The query returns the threshold level in scientific notation.

Example
```
:MATH:OPTion:THReshold1 0.8     /*Set the the threshold level of source A in logic
                                  operations to 800mV*/
:MATH:OPTion:THReshold1?        /*The query returns 8.000000e-01*/
```

Related Commands :CHANnel<n>:SCALe
:CHANnel<n>:OFFSet
:MATH:LSOUrce1

:MATH:OPTion:THReshold2

Syntax :MATH:OPTion:THReshold2 <thre>
:MATH:OPTion:THReshold2?

Description Set or query the threshold level of source B in logic operations. The default unit is V.

Parameter

Name	Type	Range	Default
<thre>	Real	(-4 x VerticalScale - VerticalOffset) to (4 x VerticalScale - VerticalOffset)	0.00V

Explanation

- This command is only applicable to A&&B, A||B, and A^B logic operations of which source B is an analog channel.
- VerticalScale refers to the vertical scale of source B
  VeticalOffset refers to the vertical offset of source B
  The step is VerticalScale/50

Return Format The query returns the threshold level in scientific notation.

Example
```
:MATH:OPTion:THReshold2 0.5     /*Set the threshold level of source B in logic
                                  operations to 0.5V*/
:MATH:OPTion:THReshold2?        /*The query returns 5.000000e-01*/
```

Related Commands :CHANnel<n>:SCALe
:CHANnel<n>:OFFSet
:MATH:LSOUrce2

:MATH:OPTion:FX:SOURce1

Syntax :MATH:OPTion:FX:SOURce1 <src>
:MATH:OPTion:FX:SOURce1?

Description Set or query source A of the inner layer operation of compound operation.

Parameter

Name	Type	Range	Default
<src>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example
```
:MATH:OPTion:FX:SOURce1 CHANnel1      /*Set source A of the inner layer operation of
                                        compound operation to CH1*/
:MATH:OPTion:FX:SOURce1?              /*The query returns CHAN1*/
```

:MATH:OPTion:FX:SOURce2

Syntax
```
:MATH:OPTion:FX:SOURce2 <src>
:MATH:OPTion:FX:SOURce2?
```

Description Set or query source B of the inner layer operation of compound operation.

Parameter

Name	Type	Range	Default
<src>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example
```
:MATH:OPTion:FX:SOURce2 CHANnel1      /*Set source B of the inner layer operation of
                                        compound operation to CH1*/
:MATH:OPTion:FX:SOURce2?              /*The query returns CHAN1*/
```

:MATH:OPTion:FX:OPERator

Syntax
```
:MATH:OPTion:FX:OPERator <op>
:MATH:OPTion:FX:OPERator?
```

Description Set or query the operator of the inner layer operation of compound operation.

Parameter

Name	Type	Range	Default
<op>	Discrete	{ADD|SUBTract|MULTiply|DIVision}	ADD

Return Format The query returns ADD, SUBT, MULT, or DIV.

Example
```
:MATH:OPTion:FX:OPERator ADD          /*Set the operator of the inner layer operation of
                                        compound operation to addition*/
:MATH:OPTion:FX:OPERator?             /*The query returns ADD*/
```

:MASK Commands

The :MASK commands are used to set and query the pass/fail test parameters.

Command List:

:MASK:ENABle

Syntax :MASK:ENABle <bool>

:MASK:ENABle?

Description Enable or disable the pass/fail test or query the status of the past/fail test.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation The pass/fail test is invalid in the following conditions: the horizontal timebase is XY or ROLL, in the slow sweep mode (namely when the horizontal timebase is YT, the horizontal timebase is set to 200ms/div or slower) as well as during waveform record.

Return Format The query returns 1 or 0.

Example

```
:MASK:ENABle ON      /*Enable the pass/fail test*/
:MASK:ENABle?        /*The query returns 1*/
```

:MASK:SOURce

Syntax :MASK:SOURce <source>

:MASK:SOURce?

Description Set or query the source of the pass/fail test.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Explanation Only channels enabled can be selected as the source. You can send the :CHANnel<n>:DISPlay command to enable the desired channel.

Return Format The query returns CHAN1 or CHAN2.

Example
```
:MASK:SOURce CHANnel2      /*Set the source of the pass/fail test to CH2*/
:MASK:SOURce?              /*The query returns CHAN2*/
```

:MASK:OPERate

Syntax :MASK:OPERate <oper>

:MASK:OPERate?

Description Run or stop the pass/fail test, or query the status of the pass/fail test.

Parameter

Name	Type	Range	Default
<oper>	Discrete	{RUN|STOP}	STOP

Explanation Before executing this command, you need to send the :MASK:ENABle command to enable the pass/fail test.

Return Format The query returns RUN or STOP.

Example
```
:MASK:OPERate RUN      /*Run the pass/fail test*/
:MASK:OPERate?         /*The query returns RUN*/
```

:MASK:MDISplay

Syntax :MASK:MDISplay <bool>

:MASK:MDISplay?

Description Enable or disable the statistic information when the pass/fail test is enabled, or query the status of the statistic information.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation

- Before executing this command, you need to send the :MASK:ENABle command to enable the pass/fail test.
- When the statistic information is on, the test results as shown in the figure below will be displayed at the upper right corner of the screen.

- You can send the :MASK:PASSed?, :MASK:FAILed?, and :MASK:TOTal? commands to query the test results.

Return Format The query returns 1 or 0.

Example
```
:MASK:MDISplay ON      /*Enable the statistic information*/
:MASK:MDISplay?        /*The query returns 1*/
```

:MASK:SOOutput

Syntax :MASK:SOOutput <bool>

:MASK:SOOutput?

Description Turn the "Stop on Fail" function on or off, or query the status of the "Stop on Fail" function.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation

- ON: when failed waveforms are detected, the oscilloscope will stop the test and enter the "STOP" state. At this point, the results of the test remain the same on the screen (if the display is turned on) and the **[Trigger Out]** connector (if enabled) on the rear panel outputs a single pulse.
- OFF: the oscilloscope will continue with the test even though failed waveforms are detected. The test results on the screen will update continuously and the **[Trigger Out]** connector on the rear panel outputs a pulse each time a failed waveform is detected.

Return Format The query returns 1 or 0.

Example

```
:MASK:SOOutput ON      /*Turn the "Stop on Fail" function on*/
:MASK:SOOutput?        /*The query returns 1*/
```

:MASK:OUTPut

Syntax :MASK:OUTPut <bool>

:MASK:OUTPut?

Description Enable or disable the sound prompt when failed waveforms are detected, or query the status of the sound prompt.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation

- OFF: when failed waveforms are detected, there are display and output but the beeper does not sound.
- ON: when failed waveforms are detected, there are display and output and the beeper sounds (not related to the on/off state of the sound).

Return Format The query returns 1 or 0.

Example

```
:MASK:OUTPut ON      /*Enable the sound prompt*/
:MASK:OUTPut?        /*The query returns 1*/
```

:MASK:X

Syntax :MASK:X <x>

:MASK:X?

Description Set or query the horizontal adjustment parameter in the pass/fail test mask. The default unit is div.

Parameter

Name	Type	Range	Default
<x>	Real	0.02 to 4, the step is 0.02 within the range	0.02

Return Format The query returns the horizontal adjustment parameter in scientific notation.

Example
```
:MASK:X 0.28     /*Set the horizontal adjustment parameter to 0.28div*/
:MASK:X?         /*The query returns 2.800000e-01*/
```

:MASK:Y

Syntax :MASK:Y <y>

:MASK:Y?

Description Set or query the vertical adjustment parameter in the pass/fail test mask. The default unit is div.

Parameter

Name	Type	Range	Default
<y>	Real	0.04 to 5.12, the step is 0.04 within the range	0.96

Return Format The query returns the vertical adjustment parameter in scientific notation.

Example
```
:MASK:Y 0.36     /*Set the vertical adjustment parameter to 0.36div*/
:MASK:Y?         /*The query returns 3.600000e-01*/
```

:MASK:CREate

Syntax :MASK:CREate

Description Create the pass/fail test mask using the current horizontal adjustment parameter and vertical adjustment parameter.

Explanation This command is valid only when the pass/fail test is enabled (:MASK:ENABle) and is not in the run state (:MASK:OPERate).

Related Commands :MASK:X

:MASK:Y

:MASK:PASSed?

Syntax :MASK:PASSed?

Description Query the number of passed frames in the pass/fail test.

Return Format The query returns an integer.

:MASK:FAILed?

Syntax :MASK:FAILed?

Description Query the number of failed frames in the pass/fail test.

Return Format The query returns an integer.

:MASK:TOTal?

Syntax :MASK:TOTal?

Description Query the total number of frames in the pass/fail test.

Return Format The query returns an integer.

:MASK:RESet

Syntax :MASK:RESet

Description Reset the numbers of passed frames and failed frames as well as the total number of frames in the pass/fail test to 0.

Related Commands :MASK:PASSed?

:MASK:FAILed?

:MASK:TOTal?

:MEASure Commands

DS1000Z-E supports the auto measurement of the following 37 waveform parameters and provides the statistic function for the measurement results. In additional, you can use the frequency counter to make more precise frequency measurement. The :MEASure commands are used to set and query the measurement parameters.

1. Time Parameters

Period: Defined as the time between the threshold middle points of two consecutive, like-polarity edges.

Frequency: Defined as the reciprocal of period.

Rise Time: The time for the signal amplitude to rise from the lower limit to the upper limit of the threshod.

Fall Time: The time for the signal amplitude to fall from the upper limit to the lower limit of the threshod.

+Width: The time difference between the threshold middle point of a rising edge to that of the next falling edge of the pulse.

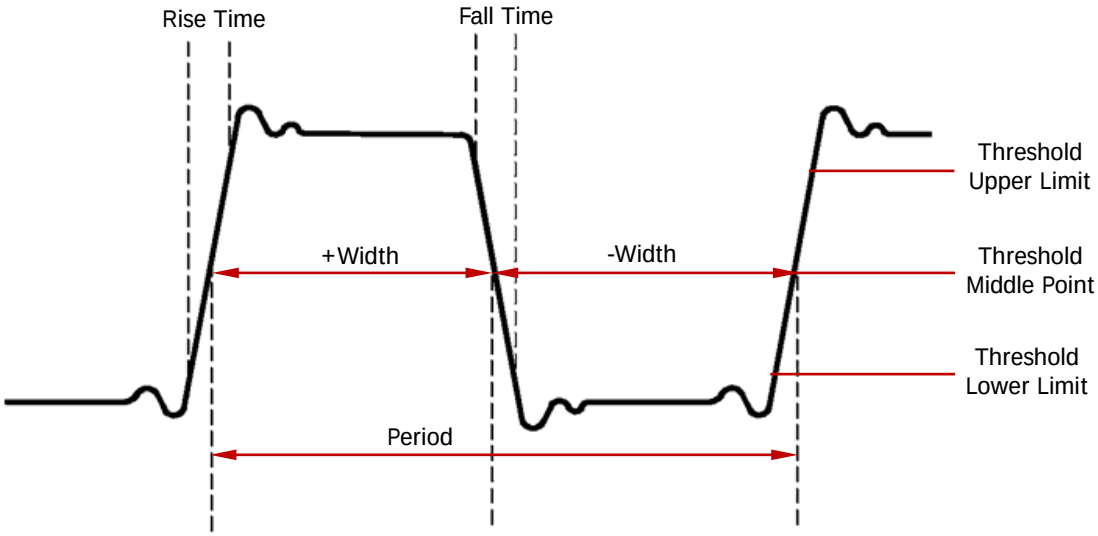
-Width: The time difference between the threshold middle point of a falling edge to that of the next rising edge of the pulse.

+Duty: The ratio of the positive pulse width to the period.

-Duty: The ratio of the negative pulse width to the period.

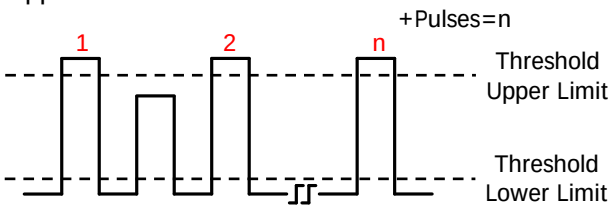
tVmax: The time corresponding to the waveform maximum value (Vmax).

tVmin: The time corresponding to the waveform minimum value (Vmin).

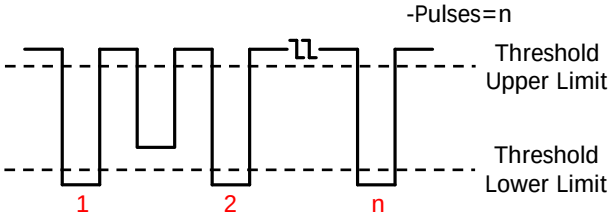


2. Count Values

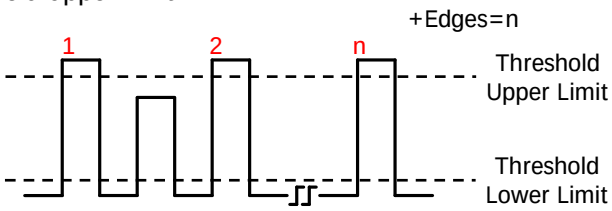
+Pulses: The number of positive pulses that rise from below the threshold lower limit to above the threshold upper limit.



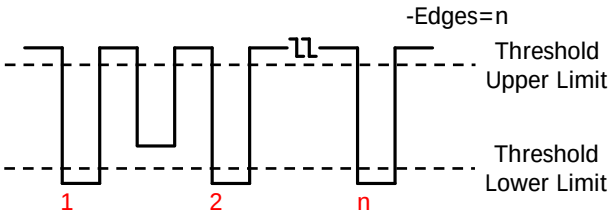
-Pulses: The number of negative pulses that fall from above the threshold upper limit to below the threshold lower limit.



+Edges: The number of rising edges that rise from below the threshold lower limit to above the threshold upper limit.



-Edges: The number of falling edges that fall from above the threshold upper limit to below the threshold lower limit.



3. Delay and Phase

Delay 1→2: The time difference between the rising edges of source 1 and source 2. Negative delay indicates that the selected rising edge of source 1 occurred after the selected rising edge of source 2.

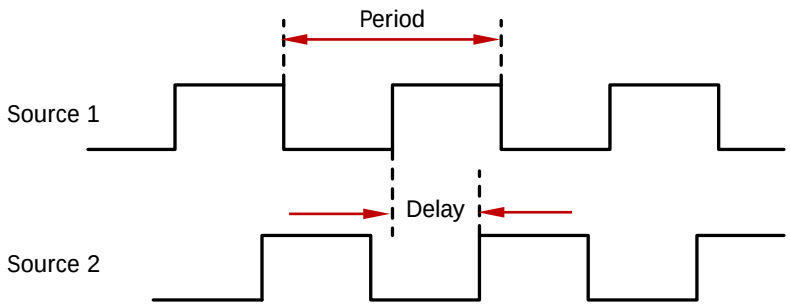
Delay 1→2: The time difference between the falling edges of source 1 and source 2. Negative delay indicates that the selected falling edge of source 1 occurred after the selected falling edge of source 2.

Phase 1→2: Phase difference calculated according to "**Delay 1→2**" and the period of source 1, expressed in degree. The calculation formula is as shown below.

Phase 1→2: Phase difference calculated according to "**Delay 1→2**" and the period of source 1, expressed in degree.The calculation formula is as shown below.

$$Phase = \frac{Delay}{Period1} \times 360°$$

Wherein, $Phase$ denotes "**Phase 1→2**" or "**Phase 1→2**", $Delay$ denotes "**Delay 1→2**" or "**Delay 1→2**" and $Period1$ denotes the period of source 1.



4. Voltage Parameters

Vmax: The voltage value from the highest point of the waveform to the GND.

Vmin: The voltage value from the lowest point of the waveform to the GND.

Vpp: The voltage value from the highest point to the lowest point of the waveform.

Vtop: The voltage value from the flat top of the waveform to the GND.

Vbase: The voltage value from the flat base of the waveform to the GND.

Vamp: The voltage value from the top of the waveform to the base of the waveform.

Vupper: The actual voltage value corresponding to the threshold maximum value.

Vmid: The actual voltage value corresponding to the threshold middle value.

Vlower: The actual voltage value corresponding to the threshold minimum value.

Vavg: The arithmetic average value on the whole waveform or on the gating area. The calculation formula is as follows:

$$Average = \frac{\sum x_i}{n}$$

Wherein, x_i is the measurement result of the ith point, n is the number of points being measured.

Vrms: The root mean square value on the whole waveform or the gating area. The calculation formula is as follows:

$$RMS = \sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}$$

Wherein, x_i is the measurement result of the ith point, n is the number of points being measured.

Overshoot: The ratio of the difference between the maximum value and top value of the waveform to the amplitude value.

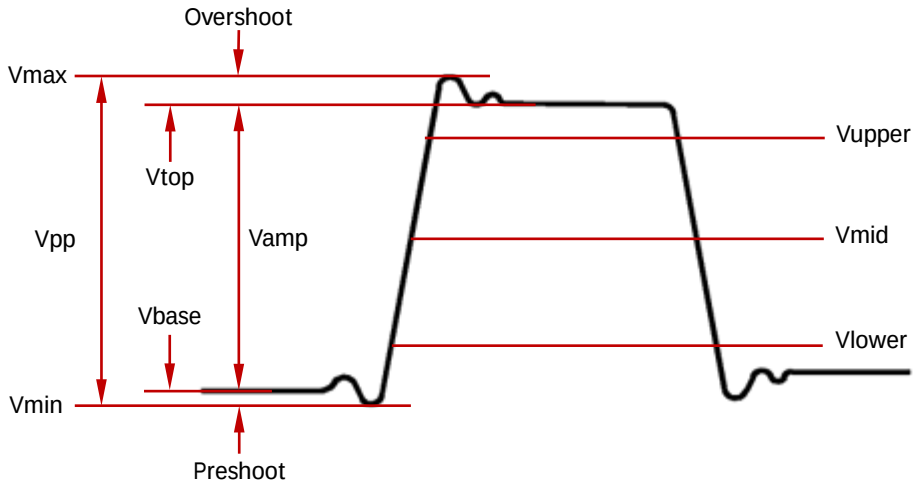
Preshoot: The ratio of the difference between the minimum value and base value of the waveform to the amplitude value.

Per.Vrms: The root mean square value within a period. For the calculation formula, please refer to "**Vrms**".

Variance: The average of the sum of the squares for the difference between the amplitude value of each waveform point and the waveform average value on the whole waveform or on the gating area. The variance reflects the fluctuation degree of the waveform. The calculation formula is as follows:

$$Variance = \frac{\sum_{i=1}^{n} (Vamp(i) - Average)^2}{n}$$

Wherein, $Vamp(i)$ is the amplitude of the ith point, $Average$ is the average value of the waveform, n is the number of points being measured.



5. Other Parameters

+Rate: Divide the difference of the upper value and lower value on the rising edge by the corresponding time.

-Rate: Divide the difference of the lower value and upper value on the falling edge by the corresponding time.

Area: The area of the whole waveform within the screen and the unit is voltage-second. The area meadured above the zero reference (namely the vertical offset) is positive and the area measured below the zero reference is negative. The area measured is the algebraic sum of the area of the whole waveform within the screen.

Period Area: The area of the first period of the waveform on the screen and the unit is voltage-second. The area above the zero reference (namely the vertical offset) is positive and the area below the zero reference is negative. The area measured is the algeraic sum of the area of the waveform within the whole period.

Command List:

- :MEASure:SOURce
- :MEASure:COUNter:SOURce
- :MEASure:COUNter:VALue?
- :MEASure:CLEar
- :MEASure:RECover
- :MEASure:ADISplay
- :MEASure:AMSource
- :MEASure:SETup:MAX
- :MEASure:SETup:MID
- :MEASure:SETup:MIN
- :MEASure:SETup:PSA
- :MEASure:SETup:PSB
- :MEASure:SETup:DSA
- :MEASure:SETup:DSB
- :MEASure:STATistic:DISPlay
- :MEASure:STATistic:MODE
- :MEASure:STATistic:RESet
- :MEASure:STATistic:ITEM
- :MEASure:ITEM

:MEASure:SOURce

Syntax :MEASure:SOURce <sour>

:MEASure:SOURce?

Description Set or query the source of the current measurement parameter.

Parameter

Name	Type	Range	Default
<sour>	Discrete	{CHANnel1|CHANnel2|MATH}	CHANnel1

Explanation

- Analog channels (CH1 and CH2) can be selected no matter wether they are currently turned on.
- This command is used to set the source for the measurement parameters (except delay and phase). The delay and phase measurements requires two sources which can be set by the :MEASure:SETup:DSA and :MEASure:SETup:DSB, :MEASure:SETup:PSA and :MEASure:SETup:PSB commands respectively.

Return Format The query returns CHAN1, CHAN2, or MATH.

Example

```
:MEASure:SOURce CHANnel2      /*Set the parameter measurement source to CH2*/
:MEASure:SOURce?              /*The query returns CHAN2*/
```

Related Command :MEASure:ITEM

:MEASure:COUNter:SOURce

Syntax :MEASure:COUNter:SOURce <sour>

:MEASure:COUNter:SOURce?

Description Set or query the source of the frequency counter, or disable the frequency counter.

Parameter

Name	Type	Range	Default
<sour>	Discrete	{CHANnel1|CHANnel2|OFF}	OFF

Explanation Analog channels (CH1 and CH2) can be selected no matter wether they are currently turned on.

Return Format The query returns CHAN1, CHAN2, or OFF.

Example

```
:MEASure:COUNter:SOURce CHANnel2      /*Set the source of the frequency counter to
                                       CH2*/
:MEASure:COUNter:SOURce?              /*The query returns CHAN2*/
```

Related Command :MEASure:COUNter:VALue?

:MEASure:COUNter:VALue?

Syntax :MEASure:COUNter:VALue?

Description Query the measurement result of the frequency counter. The default unit is Hz.

Return Format The query returns the measurement result in scientific notation. If the frequency counter is disabled, 0.0000000e+00 will be returned.

Example

```
:MEASure:COUNter:VALue?     /*The query returns 1.000004e+03*/
```

Related Command :MEASure:COUNter:SOURce

:MEASure:CLEar

Syntax :MEASure:CLEar <item>

Description Clear one or all of the last five measurement items enabled.

Parameter

Name	Type	Range	Default
<item>	Discrete	{ITEM1|ITEM2|ITEM3|ITEM4|ITEM5|ALL}	--

Explanation

- You can use the :MEASure:ITEM command to enable the desired parameters of the 37 waveform parameters. Note that the last five parameters are determined according to the order in which you turned them on and they will not change as you delete one or more measurement items.
- You can send the :MEASure:RECover command to recover the item(s) when a certain item or all the items are cleared.

Example :MEASure:CLEar ITEM1 /*Clear ITEM1*/

:MEASure:RECover

Syntax :MEASure:RECover <item>

Description Recover the measurement item which has been cleared.

Parameter

Name	Type	Range	Default
<item>	Discrete	{ITEM1|ITEM2|ITEM3|ITEM4|ITEM5|ALL}	--

Explanation

- You can use the :MEASure:ITEM command to enable the desired parameters of the 37 waveform parameters. Note that the last five parameters are determined according to the order in which you turned them on and they will not change as you delete one or more measurement items.
- You can send the :MEASure:CLEar command to clear the items again after one or all of the measurement items are recovered.

:MEASure:ADISplay

Syntax :MEASure:ADISplay <bool>

:MEASure:ADISplay?

Description Enable or disable the all measurement function, or query the status of the all measurement function.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation

- The all measurement function can measure the following 29 parameters of the source at the same time:
  Voltage Parameters: Vmax, Vmin, Vpp, Vtop, Vbase, Vamp, Vupper, Vmid, Vlower, Vavg, Vrms, Overshoot, Preshoot, Period Vrms, and Variance
  Time Parameters: Period, Frequency, Rise Time, Fall Time, +Width, -Width, +Duty, -Duty, tVmax, and tVmin
  Other Parameters: +Rate, -Rate, Area, and Period Area.
- The all measurement function can measure CH1, CH2, and the MATH channel at the same time. You can send the :MEASure:AMSource command to set the source of the all measurement function.

Return The query returns 1 or 0.

Format

Example
```
:MEASure:ADISplay ON      /*Enable the all measurement function*/
:MEASure:ADISplay?        /*The query returns 1*/
```

:MEASure:AMSource

Syntax
```
:MEASure:AMSource <src>[,<src>[,<src>[,<src>]]]
:MEASure:AMSource?
```

Description Set or query the source(s) of the all measurement function.

Parameter

Name	Type	Range	Default
<src>	Discrete	{CHANnel1|CHANnel2|MATH}	CHANnel1

Return Format The query returns CHAN1, CHAN2, or MATH. Multiple sources are separated by commas.

Example
```
:MEASure:AMSource CHANnel1,CHANnel2      /*Set the sources of the all measurement
                                           function to CH1 and CH2*/
:MEASure:AMSource?                       /*The query returns CHAN1,CHAN2*/
```

:MEASure:SETup:MAX

Syntax
```
:MEASure:SETup:MAX <value>
:MEASure:SETup:MAX?
```

Description Set or query the upper limit of the threshold (expressed in the percentage of amplitude) in time, delay, and phase measurements.

Parameter

Name	Type	Range	Default
<value>	Integer	7 to 95	90

Explanation

- The upper limit, middle point, and lower limit of the threshold (expressed in the percentage of amplitude) are used to define the time, delay, and phase parameters. Setting these values will affect the measurement results of the time, delay, and phase parameters.
- Setting the upper limit lower than the middle point will automatically reduce the middle point to keep it lower than the upper limit.

Return Format The query returns an integer between 7 and 95.

Example
```
:MEASure:SETup:MAX 95     /*Set the upper limit of the threshold to 95%*/
:MEASure:SETup:MAX?       /*The query returns 95*/
```

Related Commands

:MEASure:SETup:MID

:MEASure:SETup:MIN

:MEASure:ITEM

:MEASure:SETup:MID

Syntax :MEASure:SETup:MID <value>

:MEASure:SETup:MID?

Description Set or query the middle point of the threshold (expressed in the percentage of amplitude) in time, delay, and phase measurements.

Parameter

Name	Type	Range	Default
<value>	Integer	6 to 94	50

Explanation

- The upper limit, middle point, and lower limit of the threshold (expressed in the percentage of amplitude) are used to define the time, delay, and phase parameters. Setting these values will affect the measurement results of the time, delay, and phase parameters.
- The middle point must be lower than the upper limit and greater than the lower limit.

Return Format The query returns an integer between 6 and 94.

Example

```
:MEASure:SETup:MID 89      /*Set the middle point of the threshold to 89%*/
:MEASure:SETup:MID?        /*The query returns 89*/
```

Related Commands :MEASure:SETup:MAX

:MEASure:SETup:MIN

:MEASure:ITEM

:MEASure:SETup:MIN

Syntax :MEASure:SETup:MIN <value>

:MEASure:SETup:MIN?

Description Set or query the lower limit of the threshold (expressed in the percentage of amplitude) in time, delay, and phase measurements.

Parameter

Name	Type	Range	Default
<value>	Integer	5 to 93	10

Explanation

- The upper limit, middle point, and lower limit of the threshold (expressed in the percentage of amplitude) are used to define the time, delay, and phase parameters. Setting these values will affect the measurement results of the time, delay, and phase parameters.
- Setting the lower limit greater than the middle point will automatically increase the middle point to keep it greater than the lower limit.

Return Format The query returns an integer between 5 and 93.

Example

```
:MEASure:SETup:MIN 53      /*Set the lower limit of the threshold to 53%*/
:MEASure:SETup:MIN?        /*The query returns 53*/
```

Related Commands :MEASure:SETup:MAX

:MEASure:SETup:MID

:MEASure:ITEM

:MEASure:SETup:PSA

Syntax :MEASure:SETup:PSA <source>

:MEASure:SETup:PSA?

Description Set or query source A of Phase 1→2 and Phase 1→2 measurements.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example
```
:MEASure:SETup:PSA CHANnel1     /*Set source A of phase measurement to CH1*/
:MEASure:SETup:PSA?             /*The query returns CHAN1*/
```

Related Command :MEASure:ITEM

:MEASure:SETup:PSB

Syntax :MEASure:SETup:PSB <source>

:MEASure:SETup:PSB?

Description Set or query source B of Phase 1→2 and Phase 1→2 measurements.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel2

Return Format The query returns CHAN1 or CHAN2.

Example
```
:MEASure:SETup:PSB CHANnel2     /*Set source B of phase measurement to CH2*/
:MEASure:SETup:PSB?             /*The query returns CHAN2*/
```

Related Command :MEASure:ITEM

:MEASure:SETup:DSA

Syntax :MEASure:SETup:DSA <source>

:MEASure:SETup:DSA?

Description Set or query source A of Delay 1→2 and Delay 1→2 measurements.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example
```
:MEASure:SETup:DSA CHANnel1     /*Set source A of delay measurement to CH1*/
:MEASure:SETup:DSA?             /*The query returns CHAN1*/
```

Related Command :MEASure:ITEM

:MEASure:SETup:DSB

Syntax :MEASure:SETup:DSB <source>

:MEASure:SETup:DSB?

Description Set or query source B of Delay 1→2 and Delay 1→2 measurements.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel2

Return Format The query returns CHAN1 or CHAN2.

Example
```
:MEASure:SETup:DSB CHANnel2      /*Set source B of delay measurement to CH2*/
:MEASure:SETup:DSB?              /*The query returns CHAN2*/
```

Related Command :MEASure:ITEM

:MEASure:STATistic:DISPlay

Syntax :MEASure:STATistic:DISPlay <bool>

:MEASure:STATistic:DISPlay?

Description Enable or disable the statistic function, or query the status of the statistic function.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation When the statistic function is enabled, the oscilloscope makes statistic and displays the statistic results of at most 5 measurement items that are turned on last.

Return Format The query returns 1 or 0.

Example
```
:MEASure:STATistic:DISPlay ON    /*Enable the statistic function*/
:MEASure:STATistic:DISPlay?      /*The query returns 1*/
```

Related Commands

:MEASure:STATistic:MODE

:MEASure:STATistic:RESet

:MEASure:STATistic:ITEM

:MEASure:ITEM

:MEASure:STATistic:MODE

Syntax :MEASure:STATistic:MODE <mode>

:MEASure:STATistic:MODE?

Description Set or query the statistic mode.

Parameter

Name	Type	Range	Default
<mode>	Discrete	{DIFFerence|EXTRemum}	EXTRemum

Explanation

- DIFFerence: select the difference mode. The statistic results contain the current value, average value, standard deviation, and counts.
- EXTRemum: select the extremum mode. The statistic results contain the current value, average value, minimum, and maximum.

- Sending the :MEASure:STATistic:DISPlay command can enable the statistic function. When the statistic function is enabled, the oscilloscope makes statistic and displays the statistic results of at most 5 measurement items that are turned on last.

Return Format The query returns DIFF or EXTR.

Example
```
:MEASure:STATistic:MODE DIFFerence      /*Select the difference mode*/
:MEASure:STATistic:MODE?                /*The query returns DIFF*/
```

Related Commands :MEASure:STATistic:RESet

:MEASure:STATistic:ITEM

:MEASure:STATistic:RESet

Syntax :MEASure:STATistic:RESet

Description Clear the history data and make statistic again.

Related Commands :MEASure:STATistic:DISPlay

:MEASure:STATistic:MODE

:MEASure:STATistic:ITEM

:MEASure:STATistic:ITEM

Syntax :MEASure:STATistic:ITEM <item>[,<src>[,<src>]]

:MEASure:STATistic:ITEM? <type>,<item>[,<src>[,<src>]]

Description Enable the statistic function of any waveform parameter of the specified source, or query the statistic result of any waveform parameter of the specified source.

Parameter

Name	Type	Range	Default
<item>	Discrete	{VMAX|VMIN|VPP|VTOP|VBASe|VAMP|VAVG|VRMS|OVERshoot|PREShoot|MARea|MPARea|PERiod|FREQuency|RTIMe|FTIMe|PWIDth|NWIDth|PDUTy|NDUTy|RDELay|FDELay|RPHase|FPHase|TVMAX|TVMIN|PSLEWrate|NSLEWrate|VUPper|VMID|VLOWer|VARIance|PVRMS|PPULses|NPULses|PEDGes|NEDGes}	--
<src>	Discrete	{CHANnel1|CHANnel2|MATH}	
<type>	Discrete	{MAXimum|MINimum|CURRent|AVERages|DEViation}	--

Explanation

- [,<src>[,<src>]] sets the source of the parameter to be measured.
- If the parameter to be measured only needs a single source (VMAX, VMIN, VPP, VTOP, VBASe, VAMP, VAVG, VRMS, OVERshoot, MARea, MPARea, PREShoot, PERiod, FREQuency, RTIMe, FTIMe, PWIDth, NWIDth, PDUTy, NDUTy, TVMAX, TVMIN, PSLEWrate, NSLEWrate, VUPper, VMID, VLOWer, VARIance, PVRMS, PPULses, NPULses, PEDGes, and NEDGes), you only need to set a single source. If [,<src>[,<src>]] is omitted, the source is the one selected by the :MEASure:SOURce command by default.
- If the parameter to be measured needs two sources (RDELay, FDELay, RPHase, and FPHase), the command needs to include two sources; otherwise, the command is invalid. If [,<src>[,<src>]] is omitted, the sources are the ones selected by the :MEASure:SETup:DSA and :MEASure:SETup:DSB or :MEASure:SETup:PSA

and :MEASure:SETup:PSB commands by default.

Return Format The query returns the statistic result in scientific notation.

Example
```
:MEASure:STATistic:ITEM VPP,CHANnel2     /*Enable the Vpp statistic function of CH2*/
:MEASure:STATistic:ITEM? MAXimum,VPP     /*The query returns 9.120000e-01*/
```

:MEASure:ITEM

Syntax
```
:MEASure:ITEM <item>[,<src>[,<src>]]
:MEASure:ITEM? <item>[,<src>[,<src>]]
```

Description Measure any waveform parameter of the specified source, or query the measurement result of any waveform parameter of the specified source.

Parameter

Name	Type	Range	Default
<item>	Discrete	{VMAX|VMIN|VPP|VTOP|VBASe|VAMP|VAVG|VRMS|OVERshoot|PREShoot|MARea|MPARea|PERiod|FREQuency|RTIMe|FTIMe|PWIDth|NWIDth|PDUTy|NDUTy|RDELay|FDELay|RPHase|FPHase|TVMAX|TVMIN|PSLEWrate|NSLEWrate|VUPper|VMID|VLOWer|VARIance|PVRMS|PPULses|NPULses|PEDGes|NEDGes}	--
<src>	Discrete	{CHANnel1|CHANnel2|MATH}	

Explanation

- [,<src>[,<src>]] sets the source of the parameter to be measured.
- If the parameter to be measured only needs a single source (VMAX, VMIN, VPP, VTOP, VBASe, VAMP, VAVG, VRMS, OVERshoot, MARea, MPARea, PREShoot, PERiod, FREQuency, RTIMe, FTIMe, PWIDth, NWIDth, PDUTy, NDUTy, TVMAX, TVMIN, PSLEWrate, NSLEWrate, VUPper, VMID, VLOWer, VARIance, PVRMS, PPULses, NPULses, PEDGes, and NEDGes), you only need to set a single source. If [,<src>[,<src>]] is omitted, the source is the one selected by the :MEASure:SOURce command by default.
- If the parameter to be measured needs two sources (RDELay, FDELay, RPHase, and FPHase), the command needs to include two sources; otherwise, the command is invalid. If [,<src>[,<src>]] is omitted, the sources are the ones selected by the :MEASure:SETup:DSA and :MEASure:SETup:DSB or :MEASure:SETup:PSA and :MEASure:SETup:PSB commands by default.

Return Format The query returns the measurement result in scientific notation.

Example
```
:MEASure:ITEM OVERshoot,CHANnel2     /*Enable the overshoot measurement of CH2*/
:MEASure:ITEM? OVERshoot,CHANnel2    /*The query returns 8.888889e-03*/
```

:REFerence Commands

The :REFerence commands are used to set the reference waveform parameters.

Command List:

:REFerence:DISPlay

Syntax :REFerence:DISPlay <bool>

:REFerence:DISPlay?

Description Enable or disable the REF function, or query the status of the REF function.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Return Format The query returns 1 or 0.

Example
```
:REFerence:DISPlay ON      /*Enable the REF function*/
:REFerence:DISPlay?        /*The query returns 1*/
```

:REFerence<n>:ENABle

Syntax :REFerence<n>:ENABle <bool>

:REFerence<n>:ENABle?

Description Enable or disable the specified reference channel, or query the status of the specified reference channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2|3|4|5|6|7|8|9|10}	1
<bool>	Bool	{{1|ON}|{0|OFF}}	Ref1: 1|ON Others: 0|OFF

Return Format The query returns 1 or 0.

Example
```
:REFerence1:ENABle ON      /*Enable reference channel 1*/
:REFerence1:ENABle?        /*The query returns 1*/
```

:REFerence<n>:SOURce

Syntax :REFerence<n>:SOURce <source>

:REFerence<n>:SOURce?

Description Set or query the source of the current reference channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2|3|4|5|6|7|8|9|10}	--
<source>	Discrete	{CHANnel1|CHANnel2|MATH}	CHANnel1

Explanation Only channels currently enabled can be selected as the source of the current reference channel.

Return Format The query returns CHAN1, CHAN2, or MATH.

Example

```
:REFerence1:SOURce CHANnel1     /*Set the source of reference channel 1 to CH1*/
:REFerence1:SOURce?             /*The query returns CHAN1*/
```

:REFerence<n>:VSCale

Syntax :REFerence<n>:VSCale <scale>

:REFerence<n>:VSCale?

Description Set or query the vertical scale of the specified reference channel. The unit is the same as the unit of the source.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2|3|4|5|6|7|8|9|10}	--
<scale>	Real	Related to the probe ratio When probe ratio is 1X: 1mV to 10V When probe ratio is 10X: 10mV to 100V	100mV

Explanation This command is valid only when the specified reference channel has stored reference waveform. Otherwise, "No reference data!" will be displayed when sending this command.

Return Format The query returns the vertical scale in scientific notation.

Example

```
:REFerence1:VSCale 2      /*Set the vertical scale of reference channel 1 to 2V*/
:REFerence1:VSCale?       /*The query returns 2.000000e+00*/
```

Related Commands :REFerence<n>:SOURce

:CHANnel<n>:PROBe

:REFerence<n>:VOFFset

Syntax :REFerence<n>:VOFFset <offset>

:REFerence<n>:VOFFset?

Description Set or query the vertical offset of the specified reference channel. The unit is the same as the unit of the source.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2|3|4|5|6|7|8|9|10}	--
<offset>	Real	(-10 x RefVerticalScale) to (10 x RefVerticalScale)	0

Explanation RefVerticalScale refers to the vertical scale of the reference channel.

Return Format The query returns the vertical offset in scientific notation.

Example

```
:REFerence1:VOFFset 0.5      /*Set the vertical offset of reference channel 1 to 500mV*/
:REFerence1:VOFFset?         /*The query returns 5.000000e-01*/
```

Related Commands :REFerence<n>:SOURce

:REFerence<n>:VSCale

:REFerence<n>:RESet

Syntax :REFerence<n>:RESet

Description Reset the vertical scale and vertical offset of the specified reference channel to their default values.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2|3|4|5|6|7|8|9|10}	--

:REFerence<n>:CURRent

Syntax :REFerence<n>:CURRent

Description Select the current reference channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2|3|4|5|6|7|8|9|10}	1

Explanation Only reference channels currently turned on can be selected as the current reference channel.

Related Command :REFerence<n>:ENABle

:REFerence<n>:SAVe

Syntax :REFerence<n>:SAVe

Description Store the waveform of the current reference channel to the internal memory as reference waveform.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2|3|4|5|6|7|8|9|10}	--

Related Command :REFerence<n>:CURRent

:REFerence<n>:COLor

Syntax :REFerence<n>:COLor <color>

:REFerence<n>:COLor?

Description Set or query the display color of the current reference channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2|3|4|5|6|7|8|9|10}	--
<color>	Discrete	{GRAY|GREEn|LBLue|MAGenta|ORANge}	GRAY

Return Format The query returns GRAY, GREE, LBL, MAG, or ORAN.

Example

```
:REFerence1:COLor GREEn     /*Set the display color of reference channel 1 (the
                              current reference channel) to GREEn*/
:REFerence1:COLor?          /*The query returns GREE*/
```

Related Command :REFerence<n>:CURRent

:STORage Commands

The :STORage commands are used to set the related parameters when storing images.

Command List:

:STORage:IMAGe:TYPE

Syntax :STORage:IMAGe:TYPE <type>

:STORage:IMAGe:TYPE?

Description Set or query the image type when storing images.

Parameter

Name	Type	Range	Default
<type>	Discrete	{PNG|BMP8|BMP24|JPEG|TIFF}	PNG

Return Format The query returns PNG, BMP8, BMP24, JPEG, or TIFF.

Example

```
:STORage:IMAGe:TYPE PNG      /*Set the image type when storing images to PNG*/
:STORage:IMAGe:TYPE?         /*The query returns PNG*/
```

:STORage:IMAGe:INVERT

Syntax :STORage:IMAGe:INVERT <bool>

:STORage:IMAGe:INVERT?

Description Turn on or off the invert function when storing images; or query the status of the invert function.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Return Format The query returns ON or OFF.

Example

```
:STORage:IMAGe:INVERT ON     /*Turn on the invert function when storing images*/
:STORage:IMAGe:INVERT?       /*The query returns ON*/
```

:STORage:IMAGe:COLor

Syntax :STORage:IMAGe:COLor <bool>

:STORage:IMAGe:COLor?

Description Set the image color when storing images to color (ON) or intensity graded color (OFF); or query the image color when storing images.

Parameter

Name	Type	Range	Default
<bool>	Bool	{ON|OFF}	ON

Return Format The query returns ON or OFF.

Example

```
:STORage:IMAGe:COLor ON      /*Set the image color when storing images to color*/
:STORage:IMAGe:COLor?        /*The query returns ON*/
```

:SYSTem Commands

The :SYSTem commands are used to set the system-related parameters, such as the sound and language.

Command List:

:SYSTem:AUToscale

Syntax :SYSTem:AUToscale <bool>

:SYSTem:AUToscale?

Description Enable or disable the **AUTO** key on the front panel, or query the status of the **AUTO** key.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	1|ON

Explanation

- Users can disable the **AUTO** key by sending this command or pressing **Utility** → **Auto Options** → **Lock**. The key can only be enabled by sending this command.
- After the **AUTO** key is disabled, the Auto Scale operation is invalid. The :AUToscale command will be invalid.

Return Format The query returns 1 or 0.

Example

```
:SYSTem:AUToscale ON      /*Enable the AUTO key on the front panel*/
:SYSTem:AUToscale?        /*The query returns 1*/
```

:SYSTem:BEEPer

Syntax :SYSTem:BEEPer <bool>

:SYSTem:BEEPer?

Description Enable or disable the beeper, or query the status of the beeper.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	--

Return Format The query returns 1 or 0.

Example
```
:SYSTem:BEEPer ON      /*Enable the beeper*/
:SYSTem:BEEPer?        /*The query returns 1*/
```

:SYSTem:ERRor[:NEXT]?

Syntax :SYSTem:ERRor[:NEXT]?

Description Query and delete the last system error message.

Return Format The query returns the error message in "<message number>,<message content>" format. Wherein, <message number> is an integer and <message content> is a ASCII string. For example, -113,"Undefined header; command cannot be found".

:SYSTem:GAM?

Syntax :SYSTem:GAM?

Description Query the number of grids in the horizontal direction of the instrument screen.

Return Format The query always returns 12.

:SYSTem:LANGuage

Syntax :SYSTem:LANGuage <lang>

:SYSTem:LANGuage?

Description Set or query the system language.

Parameter

Name	Type	Range	Default
<lang>	Discrete	{SCHinese|TCHinese|ENGLish|PORTuguese|GERMan|POLish|KOREan|JAPAnese|FRENch|RUSSian}	--

Explanation The system language is not affected by sending *RST.

Return Format The query returns SCH, TCH, ENGL, PORT, GERM, POL, KOR, JAPA, FREN, or RUSS.

Example
```
:SYSTem:LANGuage SCHinese     /*Set the system language to simplified Chinese*/
:SYSTem:LANGuage?             /*The query returns SCH*/
```

:SYSTem:LOCKed

Syntax :SYSTem:LOCKed <bool>

:SYSTem:LOCKed?

Description Enable or disable the keyboard lock function, or query the status of the keyboard lock function.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Return Format The query returns 1 or 0.

:SYSTem:PON

Syntax :SYSTem:PON <pon>

:SYSTem:PON?

Description Set or query the system configuration to be recalled when the oscilloscope is powered on again after power-off.

Parameter

Name	Type	Range	Default
<pon>	Discrete	{LATest|DEFault}	LATest

Return Format The query returns LAT or DEF.

Example

```
:SYSTem:PON LATest      /*Set the system configuration to be recalled when the
                          oscilloscope is powered on again after power-off to last*/
:SYSTem:PON?            /*The query returns LAT*/
```

:SYSTem:OPTion:INSTall

Syntax :SYSTem:OPTion:INSTall <license>

Description Install a option.

Parameter

Name	Type	Range	Default
<license>	ASCII String	Refer to **Explanation**	--

Explanation

- To install an option, you need to order the option to obtain the corresponding key. Then, use the key to acquire the option license by following the steps below.
  - Log in **RIGOL** official website (www.rigol.com); click **License Activation** to enter the "Registered product license code" interface.
  - Input the correct key, serial number (press **Utility** → **System** → **System Info** to get the instrument serial number) and identifying code in the product license register interface; click **Generate** to acquire the option license.
- <license> is the option license (the hyphens should be omitted). It is a 28-byte string and can only include uppercase English characters and numbers.

Example :SYSTem:OPTion:INSTall PDUY9N9QTS9PQSWPLAETRD3UJHYA

:SYSTem:OPTion:UNINSTall

Syntax :SYSTem:OPTion:UNINSTall

Description Uninstall the options installed.

:SYSTem:RAM?

Syntax :SYSTem:RAM?

Description Query the number of analog channels of the instrument.

Return Format The query always returns 2.

:SYSTem:SETup

Syntax :SYSTem:SETup <setup_stream>

:SYSTem:SETup?

Description Import the setting parameters of the oscilloscope to restore the oscilloscope to the specified setting.

Query the setting of the oscilloscope.

Parameter

Name	Type	Range	Default
<setup_stream>	Refer to **Explanation**		

Explanation

- Here, the setting of the oscilloscope corresponds to the "Setups" in **Storage** → **Storage**.
- <setup_stream> is the specified setting parameters of the oscilloscope and is the return value of the :SYSTem:SETup? command; you cannot set the parameters manually. For the specific format, please refer to the **Return Format** below.
- This command is mainly used to help users to easily export/import the oscilloscope settings remotely. To export the setting parameters of the specified setting of the oscilloscope, send the :SYSTem:SETup? command and save the return value to a file. After that, if users want to restore the oscilloscope to the specified setting, import the setting parameters of the oscilloscope using the :SYSTem:SETup <setup_stream> command (note that <setup_stream> must be the return value formerly acquired from the query).

Return Format The return value consists of two parts, including the TMC data description header and the setting data. The format of the TMC data description header is #NXXXXXX; wherein, **#** is the denoter, **N** is lower than or equal to 9 and the N data following it denotes the length (the number of bytes) of the data stream. For example, #9000002077; wherein, **N** is 9 and **000002077** denotes that there are 2077 byte effective data. The setting data is in binary format.

:TIMebase Commands

The :TIMebase commands are used to set the horizontal parameters, such as enabling the delayed sweep and setting the horizontal timebase mode.

Command List:

:TIMebase:DELay:ENABle

Syntax :TIMebase:DELay:ENABle <bool>

:TIMebase:DELay:ENABle?

Description Enable or disable the delayed sweep, or query the status of the delayed sweep.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation Delayed sweep can be used to enlarge a length of waveform horizontally to view waveform details.

Return Format The query returns 1 or 0.

Example
```
:TIMebase:DELay:ENABle ON      /*Enable the delayed sweep*/
:TIMebase:DELay:ENABle?        /*The query returns 1*/
```

:TIMebase:DELay:OFFSet

Syntax :TIMebase:DELay:OFFSet <offset>

:TIMebase:DELay:OFFSet?

Description Set or query the delayed timebase offset. The default unit is s.

Parameter

Name	Type	Range	Default
<offset>	Real	-(LeftTime - DelayRange/2) to (RightTime - DelayRange/2)	0

Explanation LeftTime = 6 x MainScale - MainOffset
RightTime = 6 x MainScale + MainOffset
DelayRange = 12 x DelayScale
Wherein, MainScale is the current main timebase scale of the oscilloscope, MainOffset is the current main timebase offset of the oscilloscope, and DelayScale is the current delayed timebase scale of the oscilloscope.

Return Format The query returns the delayed timebase offset in scientific notation.

Example
```
:TIMebase:DELay:OFFSet 0.000002    /*Set the delayed timebase offset to 2μs*/
:TIMebase:DELay:OFFSet?            /*The query returns 2.0000000e-06*/
```

Related Commands

:TIMebase[:MAIN]:SCALe

:TIMebase[:MAIN]:OFFSet

:TIMebase:DELay:SCALe

:TIMebase:DELay:SCALe

Syntax

:TIMebase:DELay:SCALe <scale>

:TIMebase:DELay:SCALe?

Description

Set or query the delayed timebase scale. The default unit is s/div.

Parameter

Name	Type	Range	Default
<scale>	Real	Refer to **Explanation**	500ns/div

Explanation

- The maximum value of <scale> is the main timebase scale currently set, and the minimum value is expressed as: 50/(current sample rate x amplification factor).
  Wherein, the amplification factor is related to the sum number (the number of enabled analog channels plus the number of analog channels that are set as trigger sources). When the sum number counts as 1, the amplification factor is 10; when the sum number counts as 2, the amplification factor is 20.
  Note: If an analog channel is both enabled and set as the trigger source, the sum number will only be counted once.
  For example,
  - If only CH1 is enabled currently and there is only one trigger source (CH1), then, the sum number counts as 1, and the amplification factor is 10.
  - If only CH1 is enabled currently and there is only one trigger source (CH2), then, the sum number counts as 2, and the amplification factor is 20.
  - If CH1 and CH2 are enabled currently; and there are two trigger sources (CH1 and CH2), then, the sum number counts as 2, and the amplification factor is 20.
- The delayed timebase scale can only be the maximum value or the value decreased from the maximum value in 1-2-5 step. If the minimum value calculated by the expression mentioned above is not a settable value, the larger settable value that is nearest to the calculated value will be used as the minimum value.
- For example, if the following conditions are met:
  The current main timebase scale is set to 50ms/div; the sample rate is 10MSa/s; only CH1 and CH2 are enabled; and there is only one trigger source (CH2).
  It can be concluded that the amplification factor is 20. The maximum value of <scale> is 50ms/div. The minimum value is calculated using the above expression: 50/(10M x 20) = 2.5e-7, namely 250ns/div. As 250ns/div is not a settable value, take the larger settable value that is nearest to the calculated value, namely the minimum value is 500ns/div.

Return Format

The query returns the delayed timebase scale in scientific notation.

Example

```
:TIMebase:DELay:SCALe 0.00000005 /*Set the delayed timebase scale to
                                  50ns/div*/
:TIMebase:DELay:SCALe?           /*The query returns 5.0000000e-08*/
```

Related Commands
:TIMebase[:MAIN]:SCALe
:ACQuire:SRATe?
:TRIGger:MODE

:TIMebase[:MAIN]:OFFSet

Syntax
:TIMebase[:MAIN]:OFFSet <offset>
:TIMebase[:MAIN]:OFFSet?

Description
Set or query the main timebase offset. The default unit is s.

Parameter

Name	Type	Range	Default
<offset>	Real	Refer to **Explanation**	0

Explanation
- The range of <offset> is related to the current mode of the horizontal timebase (refer to :TIMebase:MODE) and run state of the oscilloscope.
  - YT mode
    RUN: (-0.5 x MemDepth/SampleRate) to 1s (when the horizontal timebase is less than 200ms/div)
    (-0.5 x MemDepth/SampleRate) to (10 x MainScale) (when the horizontal timebase is greater than or equal to 200ms/div, namely the "Slow Sweep" mode)
    STOP: (-MemDepth/SampleRate) to (1s + 0.5 x MemDepth/SampleRate)
  - Roll mode
    RUN: This command is invalid.
    STOP: (-12 x MainScale) to 0

  Wherein, MemDepth is the current memory depth of the oscilloscope, SampleRate is the current sample rate of the oscilloscope, and MainScale is the current main timebase scale of the oscilloscope.
- When the horizontal timebase mode is YT and the horizontal timebase is 200ms/div or larger (namely the "Slow Sweep" mode), this command is invalid when the oscilloscope is in the transition to the "Stop" state.

Return Format
The query returns the main timebase offset in scientific notation.

Example
```
:TIMebase:MAIN:OFFSet 0.0002     /*Set the main timebase offset to 20ms*/
:TIMebase:MAIN:OFFSet?           /*The query returns 2.0000000e-04*/
```

Related Commands
:RUN
:STOP
:ACQuire:MDEPth
:ACQuire:SRATe?
:TIMebase[:MAIN]:SCALe

:TIMebase[:MAIN]:SCALe

Syntax :TIMebase[:MAIN]:SCALe <scale>

:TIMebase[:MAIN]:SCALe?

Description Set or query the main timebase scale. The default unit is s/div.

Parameter

Name	Type	Range	Default
<scale>	Real	YT mode: 5ns/div to 50s/div in 1-2-5 step Roll mode: 200ms/div to 50s/div in 1-2-5 step	1μs/div

Explanation When the horizontal timebase mode is YT and the horizontal timebase is 200ms/div or larger (namely the "Slow Sweep" mode), this command is invalid when the oscilloscope is in the transition to the "Stop" state.

Return Format The query returns the main timebase scale in scientific notation.

Example
```
:TIMebase:MAIN:SCALe 0.0002    /*Set the main timebase scale to 200μs/div*/
:TIMebase:MAIN:SCALe?          /*The query returns 2.0000000e-04*/
```

Related Command :TIMebase:MODE

:TIMebase:MODE

Syntax :TIMebase:MODE <mode>

:TIMebase:MODE?

Description Set or query the mode of the horizontal timebase.

Parameter

Name	Type	Range	Default
<mode>	Discrete	{MAIN|XY|ROLL}	MAIN

Explanation

- MAIN: YT mode
- XY: XY mode
- ROLL: Roll mode

Return Format The query returns MAIN, XY, or ROLL.

Example
```
:TIMebase:MODE XY      /*Set the horizontal timebase mode to XY*/
:TIMebase:MODE?        /*The query returns XY*/
```

Related Commands

:TIMebase:DELay:OFFSet

:TIMebase:DELay:SCALe

:TIMebase[:MAIN]:OFFSet

:TIMebase[:MAIN]:SCALe

:TRIGger Commands

The :TRIGger commands are used to set the trigger system of the oscilloscope.

Command List:

:TRIGger:MODE

Syntax :TRIGger:MODE <mode>

:TRIGger:MODE?

Description Select or query the trigger type.

Parameter

Name	Type	Range	Default
<mode>	Discrete	{EDGE|PULSe|RUNT|WIND|NEDG|SLOPe|VIDeo|PATTern|DELay|TIMeout|DURation|SHOLd|RS232|IIC|SPI}	EDGE

Return Format The query returns EDGE, PULS, RUNT, WIND, NEDG, SLOP, VID, PATT, DEL, TIM, DUR, SHOL, RS232, IIC, or SPI.

Example

```
:TRIGger:MODE SLOPe      /*Select slope trigger*/
:TRIGger:MODE?           /*The query returns SLOP*/
```

:TRIGger:COUPling

Syntax :TRIGger:COUPling <couple>

:TRIGger:COUPling?

Description Select or query the trigger coupling type.

Parameter

Name	Type	Range	Default
<couple>	Discrete	{AC|DC|LFReject|HFReject}	DC

Explanation

- This command is only applicable to edge trigger of which the signal source is an analog channel.
- AC: block all the DC components and attenuate signals lower than 75 kHz.
- DC: allow DC and AC components into the trigger path.
- LFReject: block the DC components and reject the low frequency components (lower than 75 kHz).
- HFReject: reject the high frequency components (higher than 75 kHz).

Return Format The query returns AC, DC, LFR, or HFR.

Example

```
:TRIGger:COUPling LFReject      /*Set the trigger coupling type to low-frequency
                                 rejection*/
:TRIGger:COUPling?              /*The query returns LFR*/
```

:TRIGger:STATus?

Syntax :TRIGger:STATus?

Description Query the current trigger status.

Return Format The query returns TD, WAIT, RUN, AUTO, or STOP.

:TRIGger:SWEep

Syntax :TRIGger:SWEep <sweep>

:TRIGger:SWEep?

Description Set or query the trigger mode.

Parameter

Name	Type	Range	Default
<sweep>	Discrete	{AUTO|NORMal|SINGle}	AUTO

Explanation

- AUTO: auto trigger. No matter whether the trigger condition is met, there is always waveform display.
- NORMal: normal trigger. Display waveform when the trigger condition is met; otherwise, the oscilloscope holds the original waveform and waits for the next trigger.
- SINGle: single trigger. The oscilloscope waits for a trigger and displays the waveform when the trigger condition is met and then stops.

Return Format The query returns AUTO, NORM, or SING.

Example

```
:TRIGger:SWEep SINGle       /*Select single trigger mode */
:TRIGger:SWEep?             /*The query returns SING*/
```

:TRIGger:HOLDoff

Syntax :TRIGger:HOLDoff <value>

:TRIGger:HOLDoff?

Description Set or query the trigger holdoff time. The default unit is s.

Parameter

Name	Type	Range	Default
<value>	Real	16ns to 10s	16ns

Explanation

- Trigger holdoff can be used to stably trigger the complex waveforms (such as pulse series). Holdoff time is the time that the oscilloscope waits before re-arming the trigger circuitry. The oscilloscope will not trigger until the holdoff time expires.
- When the trigger type is video, timeout, setup/hold, Nth edge, RS232, I2C, or SPI, this setting will be not available.

Return Format The query returns the trigger holdoff time in scientific notation.

Example

```
:TRIGger:HOLDoff 0.0000002     /*Set the trigger holdoff time to 200ns*/
:TRIGger:HOLDoff?              /*The query returns 2.000000e-07*/
```

:TRIGger:NREJect

Syntax :TRIGger:NREJect <bool>

:TRIGger:NREJect?

Description Enable or disable noise rejection, or query the status of noise rejection.

Parameter

Name	Type	Range	Default
<bool>	Bool	{{1|ON}|{0|OFF}}	0|OFF

Explanation Noise rejection reduces the possibility of noise trigger.

Return Format The query returns 1 or 0.

Example

```
:TRIGger:NREJect ON      /*Enable noise rejection*/
:TRIGger:NREJect?        /*The query returns 1*/
```

:TRIGger:POSition?

Syntax :TRIGger:POSition?

Description Query the position in the internal memory that corresponds to the waveform trigger position.

Return Format The query returns an integer.

- -2 denotes that the instrument is not triggered and there is no trigger position.
- -1 denotes the instrument is triggered outside the internal memory; namely, at this point, users cannot set the instrument to read the data in the internal memory starting from the trigger position.
- An integer that is greater than 0 denotes that the return value is the position in the internal memory that corresponds to the trigger position.

Example

```
:TRIGger:POSition?      /*The query returns 100*/
```

:TRIGger:EDGe

Command List:

- :TRIGger:EDGe:SOURce
- :TRIGger:EDGe:SLOPe
- :TRIGger:EDGe:LEVel

:TRIGger:EDGe:SOURce

Syntax :TRIGger:EDGe:SOURce <source>

:TRIGger:EDGe:SOURce?

Description Set or query the trigger source in edge trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2|AC}	CHANnel1

Return Format The query returns CHAN1, CHAN2, or AC.

Example
```
:TRIGger:EDGe:SOURce CHANnel1      /*Set the trigger source to CH1*/
:TRIGger:EDGe:SOURce?              /*The query returns CHAN1*/
```

:TRIGger:EDGe:SLOPe

Syntax :TRIGger:EDGe:SLOPe <slope>

:TRIGger:EDGe:SLOPe?

Description Set or query the edge type in edge trigger.

Parameter

Name	Type	Range	Default
<slope>	Discrete	{POSitive|NEGative|RFALl}	POSitive

Explanation
POSitive: rising edge
NEGative: falling edge
RFALl: rising/falling edge

Return Format The query returns POS, NEG, or RFAL.

Example
```
:TRIGger:EDGe:SLOPe NEGative       /*Set the edge type to falling edge*/
:TRIGger:EDGe:SLOPe?               /*The query returns NEG*/
```

Related Command :TRIGger:MODE

:TRIGger:EDGe:LEVel

Syntax :TRIGger:EDGe:LEVel <level>

:TRIGger:EDGe:LEVel?

Description Set or query the trigger level in edge trigger. The unit is the same as the current amplitude unit of the signal source selected.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to the :CHANnel<n>:SCALe command. For OFFSet, refer to the :CHANnel<n>:OFFSet command.

Return Format The query returns the trigger level in scientific notation.

Example

```
:TRIGger:EDGe:LEVel 0.16      /*Set the trigger level to 160mV*/
:TRIGger:EDGe:LEVel?          /*The query returns 1.600000e-01*/
```

:TRIGger:PULSe

Command List:

:TRIGger:PULSe:SOURce

Syntax :TRIGger:PULSe:SOURce <source>

:TRIGger:PULSe:SOURce?

Description Set or query the trigger source in pulse width trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example
```
:TRIGger:PULSe:SOURce CHANnel1      /*Set the trigger source to CH1*/
:TRIGger:PULSe:SOURce?              /*The query returns CHAN1*/
```

:TRIGger:PULSe:WHEN

Syntax :TRIGger:PULSe:WHEN <when>

:TRIGger:PULSe:WHEN?

Description Set or query the trigger condition in pulse width trigger.

Parameter

Name	Type	Range	Default
<when>	Discrete	{PGReater|PLESs|NGReater|NLESs|PGLess|NGLess}	PGReater

Explanation

- PGReater/PLESs: you need to specify a pulse width (refer to :TRIGger:PULSe:WIDTh). The oscilloscope triggers when the positive pulse width of the input signal is greater/lower than the specified Pulse Width.
- NGReater/NLESs: you need to specify a pulse width (refer to :TRIGger:PULSe:WIDTh). The oscilloscope triggers when the negative pulse width of the input signal is greater/lower than the specified Pulse Width.
- PGLess/NGLess: you need to specify an upper (refer to :TRIGger:PULSe:UWIDth) and a lower (refer to :TRIGger:PULSe:LWIDth) pulse width. The oscilloscope triggers when the positive/negative pulse width of the input signal is greater than the specified lower pulse width and lower than the upper pulse width.

Return Format The query returns PGR, PLES, NGR, NLES, PGL, or NGL.

Example
```
:TRIGger:PULSe:WHEN NLESs       /*Set the trigger condition to NLESs*/
:TRIGger:PULSe:WHEN?            /*The query returns NELS*/
```

:TRIGger:PULSe:WIDTh

Syntax :TRIGger:PULSe:WIDTh <width>

:TRIGger:PULSe:WIDTh?

Description Set or query the pulse width in pulse width trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<width>	Real	8ns to 10s	PGReater, NGReater: 1μs PLESs, NLESs: 2μs

Explanation This command is available when the trigger condition (refer to :TRIGger:PULSe:WHEN) is PGReater, PLESs, NGReater, and NLESs.

Return Format The query returns the pulse width in scientific notation.

Example
```
:TRIGger:PULSe:WIDTh 0.000003    /*Set the pulse width to 3μs*/
:TRIGger:PULSe:WIDTh?            /*The query returns3.000000e-06*/
```

:TRIGger:PULSe:UWIDth

Syntax :TRIGger:PULSe:UWIDth <width>

:TRIGger:PULSe:UWIDth?

Description Set or query the upper pulse width in pulse width trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<width>	Real	16ns to 10s	2μs

Explanation This command is available when the trigger condition (refer to :TRIGger:PULSe:WHEN) is PGLess and NGLess.

Return Format The query returns the upper pulse width in scientific notation.

Example
```
:TRIGger:PULSe:UWIDth 0.000003    /*Set the upper pulse width to 3μs*/
:TRIGger:PULSe:UWIDth?            /*The query returns3.000000e-06*/
```

Related Command :TRIGger:PULSe:LWIDth

:TRIGger:PULSe:LWIDth

Syntax :TRIGger:PULSe:LWIDth <width>

:TRIGger:PULSe:LWIDth?

Description Set or query the lower pulse width in pulse width trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<width>	Real	8ns to 9.99s	1μs

Explanation This command is available when the trigger condition (refer to :TRIGger:PULSe:WHEN) is PGLess and NGLess.

Return Format The query returns the lower pulse width in scientific notation.

Example
```
:TRIGger:PULSe:LWIDth 0.000003    /*Set the lower pulse width to 3μs*/
:TRIGger:PULSe:LWIDth?            /*The query returns 3.000000e-06*/
```

Related Command :TRIGger:PULSe:UWIDth

:TRIGger:PULSe:LEVel

Syntax :TRIGger:PULSe:LEVel <level>

:TRIGger:PULSe:LEVel?

Description Set or query the trigger level in pulse width trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation

- For VerticalScale, refer to the :CHANnel<n>:SCALe command. For OFFSet, refer to the :CHANnel<n>:OFFSet command.
- This command is only available when the signal source is an analog channel.

Return Format The query returns the trigger level in scientific notation.

Example

```
:TRIGger:PULSe:LEVel 0.16     /*Set the trigger level to 160mV*/
:TRIGger:PULSe:LEVel?         /*The query returns 1.600000e-01*/
```

:TRIGger:SLOPe

Command List:

- :TRIGger:SLOPe:SOURce
- :TRIGger:SLOPe:WHEN
- :TRIGger:SLOPe:TIME
- :TRIGger:SLOPe:TUPPer
- :TRIGger:SLOPe:TLOWer
- :TRIGger:SLOPe:WINDow
- :TRIGger:SLOPe:ALEVel
- :TRIGger:SLOPe:BLEVel

:TRIGger:SLOPe:SOURce

Syntax :TRIGger:SLOPe:SOURce <source>

:TRIGger:SLOPe:SOURce?

Description Set or query the trigger source in slope trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example
```
:TRIGger:SLOPe:SOURce CHANnel2      /*Set the trigger source to CH2*/
:TRIGger:SLOPe:SOURce?              /*The query returns CHAN2*/
```

:TRIGger:SLOPe:WHEN

Syntax :TRIGger:SLOPe:WHEN <when>

:TRIGger:SLOPe:WHEN?

Description Set or query the trigger condition in slope trigger.

Parameter

Name	Type	Range	Default
<when>	Discrete	{PGReater|PLESs|NGReater|NLESs|PGLess|NGLess}	PGReater

Explanation

- PGReater/PLESs: you need to specify a time value (refer to :TRIGger:SLOPe:TIME). The oscilloscope triggers when the positive slope time of the input signal is greater/lower than the specified time.
- NGReater/NLESs: you need to specify a time value (refer to :TRIGger:SLOPe:TIME). The oscilloscope triggers when the negative slope time of the input signal is greater/lower than the specified time.
- PGLess/NGLess: you need to specify an upper limit (refer to :TRIGger:SLOPe:TUPPer) and a lower limit (refer to :TRIGger:SLOPe:TLOWer) of the time. The oscilloscope triggers when the positive/negative slope time of the input signal is greater than the specified lower limit and lower than the specified upper limit.

Return The query returns PGR, PLES, NGR, NLES, PGL, or NGL.

Format

Example
```
:TRIGger:SLOPe:WHEN NLESs      /*Set the trigger condition to NLESs*/
:TRIGger:SLOPe:WHEN?           /*The query returns NLES*/
```

:TRIGger:SLOPe:TIME

Syntax :TRIGger:SLOPe:TIME <time>

:TRIGger:SLOPe:TIME?

Description Set or query the time value in slope trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<time>	Real	8ns to 10s	PGReater, NGReater: 1μs PLESs, NLESs: 2μs

Explanation This command is available when the trigger condition (refer to :TRIGger:SLOPe:WHEN) is PGReater, PLESs, NGReater, and NLESs.

Return Format The query returns the time value in scientific notation.

Example
```
:TRIGger:SLOPe:TIME 0.000003    /*Set the time value to 3μs*/
:TRIGger:SLOPe:TIME?            /*The query returns 3.000000e-06*/
```

:TRIGger:SLOPe:TUPPer

Syntax :TRIGger:SLOPe:TUPPer <time>

:TRIGger:SLOPe:TUPPer?

Description Set or query the upper limit of the time in slope trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<time>	Real	16ns to 10s	2μs

Explanation This command is available when the trigger condition (refer to :TRIGger:SLOPe:WHEN) is PGLess and NGLess.

Return Format The query returns the upper limit of the time in scientific notation.

Example
```
:TRIGger:SLOPe:TUPPer 0.000003    /*Set the upper limit of the time to 3μs*/
:TRIGger:SLOPe:TUPPer?            /*The query returns 3.000000e-06*/
```

Related Command :TRIGger:SLOPe:TLOWer

:TRIGger:SLOPe:TLOWer

Syntax :TRIGger:SLOPe:TLOWer <time>

:TRIGger:SLOPe:TLOWer?

Description Set or query the lower limit of the time in slope trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<time>	Real	8ns to 9.99s	1μs

Explanation This command is available when the trigger condition (refer to :TRIGger:SLOPe:WHEN) is PGLess and NGLess.

Return Format The query returns the lower limit of the time in scientific notation.

Example
```
:TRIGger:SLOPe:TLOWer 0.000000020      /*Set the lower limit of the time to 20ns*/
:TRIGger:SLOPe:TLOWer?                 /*The query returns 2.000000e-08*/
```

Related Command :TRIGger:SLOPe:TUPPer

:TRIGger:SLOPe:WINDow

Syntax
```
:TRIGger:SLOPe:WINDow <window>
:TRIGger:SLOPe:WINDow?
```

Description Set or query the vertical window type in slope trigger.

Parameter

Name	Type	Range	Default
<window>	Discrete	{TA|TB|TAB}	TA

Explanation

- TA: only adjust the upper limit of the trigger level, refer to :TRIGger:SLOPe:ALEVel.
- TB: only adjust the lower limit of the trigger level, refer to :TRIGger:SLOPe:BLEVel.
- TAB: adjust the upper and lower limits of the trigger level at the same time, refer to :TRIGger:SLOPe:ALEVel and :TRIGger:SLOPe:BLEVel.

Return Format The query returns TA, TB, or TAB.

Example
```
:TRIGger:SLOPe:WINDow TB       /*Set the vertical window type to TB*/
:TRIGger:SLOPe:WINDow?         /*The query returns TB*/
```

:TRIGger:SLOPe:ALEVel

Syntax
```
:TRIGger:SLOPe:ALEVel <level>
:TRIGger:SLOPe:ALEVel?
```

Description Set or query the upper limit of the trigger level in slope trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	2V

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the upper limit of the trigger level in scientific notation.

Example
```
:TRIGger:SLOPe:ALEVel 0.16     /*Set the upper limit of the trigger level to 160mV*/
:TRIGger:SLOPe:ALEVel?         /*The query returns 1.600000e-01*/
```

Related Command :TRIGger:SLOPe:BLEVel

:TRIGger:SLOPe:BLEVel

Syntax :TRIGger:SLOPe:BLEVel <level>

:TRIGger:SLOPe:BLEVel?

Description Set or query the lower limit of the trigger level in slope trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the lower limit of the trigger level in scientific notation.

Example

```
:TRIGger:SLOPe:BLEVel 0.16     /*Set the lower limit of the trigger level to 160mV*/
:TRIGger:SLOPe:BLEVel?         /*The query returns 1.600000e-01*/
```

Related Command :TRIGger:SLOPe:ALEVel

:TRIGger:VIDeo

Command List:

:TRIGger:VIDeo:SOURce

Syntax :TRIGger:VIDeo:SOURce <source>

:TRIGger:VIDeo:SOURce?

Description Select or query the trigger source in video trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example

```
:TRIGger:VIDeo:SOURce CHANnel2      /*Set the trigger source to CH2*/
:TRIGger:VIDeo:SOURce?              /*The query returns CHAN2*/
```

:TRIGger:VIDeo:POLarity

Syntax :TRIGger:VIDeo:POLarity <polarity>

:TRIGger:VIDeo:POLarity?

Description Select or query the video polarity in video trigger.

Parameter

Name	Type	Range	Default
<polarity>	Discrete	{POSitive|NEGative}	POSitive

Return Format The query returns POS or NEG.

Example

```
:TRIGger:VIDeo:POLarity POSitive    /*Set or query the video polarity to positive*/
:TRIGger:VIDeo:POLarity?            /*The query returns POS*/
```

:TRIGger:VIDeo:MODE

Syntax :TRIGger:VIDeo:MODE <mode>

:TRIGger:VIDeo:MODE?

Description Set or query the sync type in video trigger.

Parameter

Name	Type	Range	Default
<mode>	Discrete	{ODDField|EVENfield|LINE|ALINes}	ALINes

Explanation

- ODDField: trigger on the rising edge of the first ramp waveform pulse in the odd field. This type is available when the video standard is NTSC or PAL/SECAM.

- EVENfield: trigger on the rising edge of the first ramp waveform pulse in the even field. This type is available when the video standard is NTSC or PAL/SECAM.
- LINE: for the NTSC and PAL/SECAM video standards, trigger on the specified line in the odd or even field; for the 480P and 576P video standards, trigger on the specified line.
- ALINes: trigger on all the horizontal sync pulses.

Return Format The query returns ODDF, EVEN, LINE, or ALIN.

Example
```
:TRIGger:VIDeo:MODE ODDField    /*Set the sync type to odd field*/
:TRIGger:VIDeo:MODE?            /*The query returns ODDF*/
```

Related Commands :TRIGger:VIDeo:LINE
:TRIGger:VIDeo:STANdard

:TRIGger:VIDeo:LINE

Syntax :TRIGger:VIDeo:LINE <line>

:TRIGger:VIDeo:LINE?

Description Set or query the line number when the sync type in video trigger is LINE.

Parameter

Name	Type	Range	Default
<line>	Integer	Refer to **Explanation**	1

Explanation
- NTSC: 1 to 525
- PAL/SECAM: 1 to 625
- 480P: 1 to 525
- 576P: 1 to 625

Return Format The query returns an integer.

Example
```
:TRIGger:VIDeo:LINE 100     /*Set the line number to 100*/
:TRIGger:VIDeo:LINE?        /*The query returns 100*/
```

Related Command :TRIGger:VIDeo:MODE

:TRIGger:VIDeo:STANdard

Syntax :TRIGger:VIDeo:STANdard <standard>

:TRIGger:VIDeo:STANdard?

Description Set or query the video standard in video trigger.

Parameter

Name	Type	Range	Default
<standard>	Discrete	{PALSecam|NTSC|480P|576P}	NTSC

Explanation
- PALSecam:
  PAL: the frame frequency is 25 frames per second. The TV sweep line is 625 with the odd field goes first and the even field follows behind.
  SECAM: the frame frequency is 25 frames per second. The TV sweep line is 625 with interlacing sweep.
- NTSC: the field frequency is 60 fields per second and the frame frequency is 30

frames per second. The TV sweep line is 525 with the even field goes first and the odd field follows behind.

- 480P: the frame frequency is 60 frames per second; the TV sweep line is 525; line-by-line sweep; the line frequency is 31.5 kHz.
- 576P: the frame frequency is 60 frames per second; the TV sweep line is 625; line-by-line sweep.

Return Format The query returns PALS, NTSC, 480P, or 576P.

Example
```
:TRIGger:VIDeo:STANdard NTSC      /*Select NTSC video standard*/
:TRIGger:VIDeo:STANdard?          /*The query returns NTSC*/
```

Related Commands :TRIGger:VIDeo:LINE

:TRIGger:VIDeo:MODE

:TRIGger:VIDeo:LEVel

Syntax :TRIGger:VIDeo:LEVel <level>

:TRIGger:VIDeo:LEVel?

Description Set or query the trigger level in video trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the trigger level in scientific notation.

Example
```
:TRIGger:VIDeo:LEVel 0.16     /*Set the trigger level to 160mV*/
:TRIGger:VIDeo:LEVel?         /*The query returns 1.600000e-01*/
```

:TRIGger:PATTern

Command List:

- :TRIGger:PATTern:PATTern
- :TRIGger:PATTern:LEVel

:TRIGger:PATTern:PATTern

Syntax :TRIGger:PATTern:PATTern <pa_ch1>[,<pa_ch2>]

:TRIGger:PATTern:PATTern?

Description Set or query the pattern of each channel in pattern trigger.

Parameter

Name	Type	Range	Default
<pa_ch1>	Discrete	{H|L|X|R|F}	X
<pa_ch2>	Discrete	{H|L|X|R|F}	X

Explanation

- <pa_ch1> and <pa_ch2> set the patterns of analog channels CH1 and CH2.
- Users can send 2 parameters to set the pataterns of two channels. Users can also omit the parameter <pa_ch2> to set the pattern of CH1 (the pattern state of the CH2 remains unchanged).
- In the range of the parameter, H represents high level (higher than the threshold level of the channel). L represents low level (lower than the threshold level of the channel). X denotes that this channel is ignored (this channel is not used as a part of the pattern. When all the channels in the pattern are set to X, the oscilloscope will not trigger). R represents the rising edge and F represents the falling edge (in the pattern, only a single rising edge or falling edge can be defined. If an edge item is currently defined and then another edge item is defined in another channel in the pattern, the former edge item defined will be replaced by X).
- Please distinguish "omit parameter" from "ignore channel". The former means that the parameters corresponding to some channels are not sent when sending the command and the pattern states of these channels remain unchanged. The latter means that the channel is set to X when sending the command and the state of this channel is ignored when the instrument judges the patterns.

Return Format The query returns the patterns of all the channels (2). The patterns of multiple channels are separated by commas.

Example

```
:TRIGger:PATTern:PATTern H,R,L,X /*Set the pattern of CH1 to H. The pattern of CH2
                                  remains unchanged*/
:TRIGger:PATTern:PATTern?        /*The query returns H,X*/
```

:TRIGger:PATTern:LEVel

Syntax :TRIGger:PATTern:LEVel <chan>,<level>

:TRIGger:PATTern:LEVel? <chan>

Description Set or query the trigger level of the specified channel in pattern trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<chan>	Discrete	{CHANnel1|CHANnel2}	CHANnel1
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the trigger level in scientific notation.

Example

```
:TRIGger:PATTern:LEVel CHANnel2,0.16   /*Set the trigger level of CH2 to 160mV*/
:TRIGger:PATTern:LEVel? CHANnel2       /*The query returns 1.600000e-01*/
```

:TRIGger:DURATion

Command List:

:TRIGger:DURATion:SOURce

Syntax :TRIGger:DURATion:SOURce <source>

:TRIGger:DURATion:SOURce?

Description Set or query the trigger source in duration trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example

```
:TRIGger:DURATion:SOURce CHANnel2      /*Set the trigger source to CH2*/
:TRIGger:DURATion:SOURce?              /*The query returns CHAN2*/
```

:TRIGger:DURATion:TYPe

Syntax :TRIGger:DURATion:TYPe <type_ch1>[,<type_ch2>]

:TRIGger:DURATion:TYPe?

Description Set or query the pattern of each channel in duration trigger.

Parameter

Name	Type	Range	Default
<type_ch1>	Discrete	{H|L|X}	X
<type_ch2>	Discrete	{H|L|X}	X

Explanation

- <type_ch1> and <type_ch2> set the patterns of analog channels CH1 and CH2.
- Users can send 2 parameters to set the patateras of two channels. Users can also omit <type_ch2> to set the pattern of CH1 (the pattern state of CH2 remains unchanged).
- In the range of the parameter, H represents high level (higher than the threshold level of the channel). L represents low level (lower than the threshold level of the channel). X denotes that this channel is ignored (this channel is not used as a part of the pattern. When all the channels in the pattern are set to X, the oscilloscope will not trigger).
- Please distinguish "omit parameter" from "ignore channel". The former means that the parameters corresponding to some channels are not sent when sending the command and the pattern states of these channels remain unchanged. The latter means that the channel is set to X when sending the command and the state of this channel is ignored when the instrument judges the patterns.

Return Format The query returns the patterns of all the channels (2). The patterns of multiple channels are separated by commas.

Example
```
:TRIGger:DURATion:TYPe L,X,H,L   /*Set the pattern of CH1 to L. The pattern of CH2 remains unchanged*/
:TRIGger:DURATion:TYPe?          /*The query returns L,X */
```

:TRIGger:DURATion:WHEN

Syntax
```
:TRIGger:DURATion:WHEN <when>
:TRIGger:DURATion:WHEN?
```

Description Set or query the trigger condition in duration trigger.

Parameter

Name	Type	Range	Default
<when>	Discrete	{GREater|LESS|GLESs}	GREater

Explanation

- GREater: you need to specify a time (refer to :TRIGger:DURATion:TLOWer). The oscilloscope triggers when the duration of the pattern is greater than the preset time.
- LESS: you need to specify a time (refer to :TRIGger:DURATion:TUPPer). The oscilloscope triggers when the duration of the pattern is lower than the preset time.
- GLESs: you need to specify an upper limit of the time (refer to :TRIGger:DURATion:TUPPer) and lower limit of the time (refer to :TRIGger:DURATion:TLOWer). The oscilloscope triggers when the duration of the pattern is lower than the preset upper limit of the time and greater than the preset lower limit of the time.

Return Format The query returns GRE, LESS, or GLES.

Example
```
:TRIGger:DURATion:WHEN LESS   /*Set the trigger condition to <*/
:TRIGger:DURATion:WHEN?       /*The query returns LESS*/
```

:TRIGger:DURATion:TUPPer

Syntax
```
:TRIGger:DURATion:TUPPer <NR3>
:TRIGger:DURATion:TUPPer?
```

Description Set or query the duration time upper limit in duration trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<NR3>	Real	Related to the trigger condition LESS: 8ns to 10s GLESs: 16ns to 10s	2μs

Explanation This command is available when the trigger condition (:TRIGger:DURATion:WHEN) is LESS or GLESs.

Return Format The query returns the duration time upper limit in scientific notation.

Example
```
:TRIGger:DURATion:TUPPer 0.000003   /*Set the duration time upper limit to 3μs*/
:TRIGger:DURATion:TUPPer?           /*The query returns 3.000000e-06*/
```

Related Command :TRIGger:DURATion:TLOWer

:TRIGger:DURATion:TLOWer

Syntax :TRIGger:DURATion:TLOWer <NR3>

:TRIGger:DURATion:TLOWer?

Description Set or query the duration time lower limit in duration trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<NR3>	Real	8ns to 9.99s	1μs

Explanation This command is available when the trigger condition (:TRIGger:DURATion:WHEN) is GREater or GLESs.

Return Format The query returns the duration time lower limit in scientific notation.

Example

```
:TRIGger:DURATion:TLOWer 0.000003    /*Set the duration time lower limit to 3μs*/
:TRIGger:DURATion:TLOWer?            /*The query returns 3.000000e-06*/
```

Related Command :TRIGger:DURATion:TUPPer

:TRIGger:TIMeout

Command List:

- :TRIGger:TIMeout:SOURce
- :TRIGger:TIMeout:SLOPe
- :TRIGger:TIMeout:TIMe

:TRIGger:TIMeout:SOURce

Syntax :TRIGger:TIMeout:SOURce <source>

:TRIGger:TIMeout:SOURce?

Description Set or query the trigger source in timeout trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example
```
:TRIGger:TIMeout:SOURce CHANnel2      /*Set the trigger source to CH2*/
:TRIGger:TIMeout:SOURce?              /*The query returns CHAN2*/
```

:TRIGger:TIMeout:SLOPe

Syntax :TRIGger:TIMeout:SLOPe <slope>

:TRIGger:TIMeout:SLOPe?

Description Set or query the edge type in timeout trigger.

Parameter

Name	Type	Range	Default
<slope>	Discrete	{POSitive|NEGative|RFALl}	POSitive

Explanation

- POSitive: start timing when the rising edge of the input signal passes through the trigger level.
- NEGative: start timing when the falling edge of the input signal passes through the trigger level.
- RFALl: start timing when any edge of the input signal passes through the trigger level.

Return Format The query returns POS, NEG, or RFAL.

Example
```
:TRIGger:TIMeout:SLOPe NEGative      /*Set the edge type to falling edge*/
:TRIGger:TIMeout:SLOPe?              /*The query returns NEG*/
```

:TRIGger:TIMeout:TIMe

Syntax :TRIGger:TIMeout:TIMe <NR3>

:TRIGger:TIMeout:TIMe?

Description Set or query the timeout time in timeout trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<NR3>	Real	16ns to 10s	16ns

Return Format The query returns the timeout time in scientific notation.

Example
```
:TRIGger:TIMeout:TIMe 0.002     /*Set the timeout time to 2ms*/
:TRIGger:TIMeout:TIMe?          /*The query returns 2.000000e-03*/
```

:TRIGger:RUNT

Command List:

:TRIGger:RUNT:SOURce

Syntax :TRIGger:RUNT:SOURce <source>

:TRIGger:RUNT:SOURce?

Description Set or query the trigger source in runt trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example

```
:TRIGger:RUNT:SOURce CHANnel2      /*Set the trigger source to CH2*/
:TRIGger:RUNT:SOURce?              /*The query returns CHAN2*/
```

:TRIGger:RUNT:POLarity

Syntax :TRIGger:RUNT:POLarity <polarity>

:TRIGger:RUNT:POLarity?

Description Set or query the pulse polarity in runt trigger.

Parameter

Name	Type	Range	Default
<polarity>	Discrete	{POSitive|NEGative}	POSitive

Explanation

- POSitive: the instrument triggers on the positive runt pulse.
- NEGative: the instrument triggers on the negative runt pulse.

Return Format The query returns POS or NEG.

Example

```
:TRIGger:RUNT:POLarity NEGative    /*Set the pulse polarity to negative*/
:TRIGger:RUNT:POLarity?            /*The query returns NEG*/
```

:TRIGger:RUNT:WHEN

Syntax :TRIGger:RUNT:WHEN <when>

:TRIGger:RUNT:WHEN?

Description Set or query the qualifier in runt trigger.

Parameter

Name	Type	Range	Default
<when>	Discrete	{NONE|GREater|LESS|GLESs}	NONE

Explanation

- NONE: do not set the trigger condition in runt trigger.
- GREater: trigger when the runt pulse width is greater than the lower limit of the pulse width (refer to :TRIGger:RUNT:WLOWer).
- LESS: trigger when the runt pulse width is lower than the upper limit of the pulse width (refer to :TRIGger:RUNT:WUPPer).
- GLESs: trigger when the runt pulse width is greater than the lower limit (refer to :TRIGger:RUNT:WLOWer) and lower than the upper limit of the pulse width (refer to :TRIGger:RUNT:WUPPer). Note: the lower limit of the pulse width must be lower than the upper limit.

Return Format The query returns NONE, GRE, LESS, or GLES.

Example

```
:TRIGger:RUNT:WHEN LESS      /*Set the qualifier to <*/
:TRIGger:RUNT:WHEN?          /*The query returns LESS*/
```

:TRIGger:RUNT:WUPPer

Syntax :TRIGger:RUNT:WUPPer <NR3>

:TRIGger:RUNT:WUPPer?

Description Set or query the pulse width upper limit in runt trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<NR3>	Real	When the qualifier is LESS, the range is from 8ns to 10s; when the qualifier is GLESs, the range is from 16ns to 10s.	2μs

Explanation

- This command is only available when the qualifier (refer to :TRIGger:RUNT:WHEN) is LESS or GLESs.
- When the qualifier is GLESs, the upper limit of the pulse width must be greater than the lower limit of the pulse width (:TRIGger:RUNT:WLOWer).

Return Format The query returns the pulse width upper limit in scientific notation.

Example

```
:TRIGger:RUNT:WUPPer 0.02    /*Set the pulse width upper limit to 20ms*/
:TRIGger:RUNT:WUPPer?        /*The query returns 2.000000e-02*/
```

:TRIGger:RUNT:WLOWer

Syntax :TRIGger:RUNT:WLOWer <NR3>

:TRIGger:RUNT:WLOWer?

Description Set or query the pulse width lower limit in runt trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<NR3>	Real	When the qualifier is GREater, the range is from 8ns to 10s; when the qualifier is GLESs, the range is from 8ns to 9.99s.	1μs

Explanation

- This command is only available when the qualifier (refer to :TRIGger:RUNT:WHEN) is GREater or GLESs.
- When the qualifier is GLESs, the lower limit of the pulse width must be smaller than the upper limit of the pulse width (:TRIGger:RUNT:WUPPer).

Return Format The query returns the pulse width lower limit in scientific notation.

Example

```
:TRIGger:RUNT:WLOWer 0.01     /*Set the pulse width lower limit to 1ms*/
:TRIGger:RUNT:WLOWer?         /*The query returns 1.000000e-03*/
```

:TRIGger:RUNT:ALEVel

Syntax :TRIGger:RUNT:ALEVel <level>

:TRIGger:RUNT:ALEVel?

Description Set or query the trigger level upper limit in runt trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	2V

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the trigger level upper limit in scientific notation.

Example

```
:TRIGger:RUNT:ALEVel 0.16     /*Set the trigger level upper limit to 160mV*/
:TRIGger:RUNT:ALEVel?         /*The query returns 1.600000e-01*/
```

:TRIGger:RUNT:BLEVel

Syntax :TRIGger:RUNT:BLEVel <level>

:TRIGger:RUNT:BLEVel?

Description Set or query the trigger level lower limit in runt trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the trigger level lower limit in scientific notation.

Example

```
:TRIGger:RUNT:BLEVel 0.16     /*Set the trigger level lower limit to 160mV*/
:TRIGger:RUNT:BLEVel?         /*The query returns 1.600000e-01*/
```

:TRIGger:WINDows

Command List:

:TRIGger:WINDows:SOURce

Syntax :TRIGger:WINDows:SOURce <source>

:TRIGger:WINDows:SOURce?

Description Set or query the trigger source in windows trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example

```
:TRIGger:WINDows:SOURce CHANnel2      /*Set the trigger source to CH2*/
:TRIGger:WINDows:SOURce?              /*The query returns CHAN2*/
```

:TRIGger:WINDows:SLOPe

Syntax :TRIGger:WINDows:SLOPe <type>

:TRIGger:WINDows:SLOPe?

Description Set or query the windows type in windows trigger.

Parameter

Name	Type	Range	Default
<type>	Discrete	{POSitive|NEGative|RFALl}	POSitive

Explanation

- POSitive: trigger on the rising edge of the input signal when the voltage level is greater than the preset high trigger level.
- NEGative: trigger on the falling edge of the input signal when the voltage level is lower than the preset low trigger level.
- RFALl: trigger on any edge of the input signal when the voltage level meets the preset trigger level.

Return Format The query returns POS, NEG, or RFAL.

Example

```
:TRIGger:WINDows:SLOPe NEGative      /*Set the windows type to NEGative*/
:TRIGger:WINDows:SLOPe?              /*The query returns NEG*/
```

:TRIGger:WINDows:POSition

Syntax :TRIGger:WINDows:POSition <pos>

:TRIGger:WINDows:POSition?

Description Set or query the trigger position in windows trigger.

Parameter

Name	Type	Range	Default
<type>	Discrete	{EXIT|ENTER|TIMe}	ENTER

Explanation

- EXIT: trigger when the input signal exits the specified trigger level range.
- ENTER: trigger when the trigger signal enters the specified trigger level range.
- TIMe: used to specify the hold time of the input signal after it enters the specified trigger level range. The instrument triggers when the accumulated hold time equals the windows time.

Return Format The query returns EXIT, ENTER, or TIM.

Example

```
:TRIGger:WINDows:POSition ENTER      /*Set the trigger position to enter*/
:TRIGger:WINDows:POSition?           /*The query returns ENTER*/
```

:TRIGger:WINDows:TIMe

Syntax :TRIGger:WINDows:TIMe <NR3>

:TRIGger:WINDows:TIMe?

Description Set or query the hold time in windows trigger.

Parameter

Name	Type	Range	Default
<NR3>	Real	8ns to 10s	1μs

Return Format The query returns the hold time in scientific notation.

Example

```
:TRIGger:WINDows:TIMe 0.002     /*Set the hold time to 2ms*/
:TRIGger:WINDows:TIMe?          /*The query returns 2.000000e-03*/
```

Related Command :TRIGger:WINDows:POSition

:TRIGger:WINDows:ALEVel

Syntax :TRIGger:WINDows:ALEVel <level>

:TRIGger:WINDows:ALEVel?

Description Set or query the trigger level upper limit in windows trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	2V

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the trigger level upper limit in scientific notation.

Example
```
:TRIGger:WINDows:ALEVel 0.16    /*Set the trigger level upper limit to 160mV*/
:TRIGger:WINDows:ALEVel?        /*The query returns 1.600000e-01*/
```

:TRIGger:WINDows:BLEVel

Syntax :TRIGger:WINDows:BLEVel <level>

:TRIGger:WINDows:BLEVel?

Description Set or query the tigger level lower limit in windows trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the tigger level lower limit in scientific notation.

Example
```
:TRIGger:WINDows:BLEVel 0.05    /*Set the tigger level lower limit to 50mV*/
:TRIGger:WINDows:BLEVel?        /*The query returns 5.000000e-02*/
```

:TRIGger:DELay

Command List:

:TRIGger:DELay:SA

Syntax :TRIGger:DELay:SA <Source>

:TRIGger:DELay:SA?

Description Set or query the trigger source A in delay trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example
```
:TRIGger:DELay:SA CHANnel2      /*Set the trigger source A to CH2*/
:TRIGger:DELay:SA?              /*The query returns CHAN2*/
```

:TRIGger:DELay:SLOPA

Syntax :TRIGger:DELay:SLOPA <slope>

:TRIGger:DELay:SLOPA?

Description Set or query the edge type of edge A in delay trigger.

Parameter

Name	Type	Range	Default
<slope>	Discrete	{POSitive|NEGative}	POSitive

Return Format The query returns POS or NEG.

Example
```
:TRIGger:DELay:SLOPA NEGative     /*Set the edge type of edge A to falling edge*/
:TRIGger:DELay:SLOPA?             /*The query returns NEG*/
```

:TRIGger:DELay:SB

Syntax :TRIGger:DELay:SB <source>

:TRIGger:DELay:SB?

Description Set or query the trigger source B in delay trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel2

Return Format The query returns CHAN1 or CHAN2.

Example
```
:TRIGger:DELay:SB CHANnel2      /*Set the trigger source B to CH2*/
:TRIGger:DELay:SB?              /*The query returns CHAN2*/
```

:TRIGger:DELay:SLOPB

Syntax :TRIGger:DELay:SLOPB <slope>

:TRIGger:DELay:SLOPB?

Description Set or query the edge type of edge B in delay trigger.

Parameter

Name	Type	Range	Default
<slope>	Discrete	{POSitive|NEGative}	POSitive

Return Format The query returns POS or NEG.

Example
```
:TRIGger:DELay:SLOPB NEGative     /*Set the edge type of edge B to falling edge*/
:TRIGger:DELay:SLOPB?             /*The query returns NEG*/
```

:TRIGger:DELay:TYPe

Syntax :TRIGger:DELay:TYPe <type>

:TRIGger:DELay:TYPe?

Description Set or query the delay type in delay trigger.

Parameter

Name	Type	Range	Default
<type>	Discrete	{GREater|LESS|GLESs |GOUT}	GREater

Explanation

- GREater: trigger when the time difference (ΔT) between the specified edges of tigger source A and tigger source B is greater than the preset time limit.
- LESS: trigger when the time difference (ΔT) between the specified edges of tigger source A and tigger source B is lower than the preset time limit.
- GLESs: trigger when the time difference (ΔT) between the specified edges of tigger source A and tigger source B is greater than the lower limit of the preset time and lower than the upper limit of the preset time. Note that the time lower limit must be lower than the time upper limit.
- GOUT: trigger when the time difference (ΔT) between the specified edges of tigger source A and tigger source B is lower than the lower limit of the preset time or greater than the upper limit of the preset time. Note that the time lower limit must be lower than the time upper limit.

Return Format The query returns GOUT, GRE, LESS, or GLES.

Example

```
:TRIGger:DELay:TYPe GOUT      /*Set the delay type to ><*/
:TRIGger:DELay:TYPe?          /*The query returns GOUT*/
```

Related Commands :TRIGger:DELay:TUPPer

:TRIGger:DELay:TLOWer

:TRIGger:DELay:TUPPer

Syntax :TRIGger:DELay:TUPPer <NR3>

:TRIGger:DELay:TUPPer?

Description Set or query the upper limit of the delay time in delay trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<NR3>	Real	16ns to 10s	2μs

Explanation This command is only available when the delay type (refer to :TRIGger:DELay:TYPe) is LESS, GOUT, or GLESs.

Return Format The query returns the upper limit of the delay time in scientific notation.

Example

```
:TRIGger:DELay:TUPPer 0.002   /*Set the upper limit of the delay time to 2ms*/
:TRIGger:DELay:TUPPer?        /*The query returns 2.000000e-03*/
```

Related Command :TRIGger:DELay:TLOWer

:TRIGger:DELay:TLOWer

Syntax :TRIGger:DELay:TLOWer <NR3>

:TRIGger:DELay:TLOWer?

Description Set or query the lower limit of the delay time in delay trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<NR3>	Real	Related to the delay type GREater: 8ns to 10s GOUT or GLESs: 8ns to 9.99s	1μs

Explanation This command is only available when the delay type (refer to :TRIGger:DELay:TYPe) is GREater, GOUT, or GLESs.

Return Format The query returns the lower limit of the delay time in scientific notation.

Example
```
:TRIGger:DELay:TLOWer 0.002    /*Set the lower limit of the delay time to 2ms*/
:TRIGger:DELay:TLOWer?         /*The query returns 2.000000e-03*/
```

Related Command :TRIGger:DELay:TUPPer

:TRIGger:SHOLd

Command List:

:TRIGger:SHOLd:DSrc

Syntax :TRIGger:SHOLd:DSrc <source>

:TRIGger:SHOLd:DSrc?

Description Set or query the data source in setup/hold trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example
```
:TRIGger:SHOLd:DSrc CHANnel1      /*Set the data source to CH1*/
:TRIGger:SHOLd:DSrc?              /*The query returns CHAN1*/
```

:TRIGger:SHOLd:CSrc

Syntax :TRIGger:SHOLd:CSrc <source>

:TRIGger:SHOLd:CSrc?

Description Set or query the clock source in setup/hold trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel2

Return Format The query returns CHAN1 or CHAN2.

Example
```
:TRIGger:SHOLd:CSrc CHANnel2      /*Set the clock source to CH2*/
:TRIGger:SHOLd:CSrc?              /*The query returns CHAN2*/
```

:TRIGger:SHOLd:SLOPe

Syntax :TRIGger:SHOLd:SLOPe <slope>

:TRIGger:SHOLd:SLOPe?

Description Set or query the edge type in setup/hold trigger.

Parameter

Name	Type	Range	Default
<slope>	Discrete	{POSitive|NEGative}	POSitive

Return Format The query returns POS or NEG.

Example
```
:TRIGger:SHOLd:SLOPe NEGative      /*Set the edge type to falling edge*/
:TRIGger:SHOLd:SLOPe?              /*The query returns NEG*/
```

:TRIGger:SHOLd:PATTern

Syntax :TRIGger:SHOLd:PATTern <pattern>

:TRIGger:SHOLd:PATTern?

Description Set or query the data type in setup/hold trigger.

Parameter

Name	Type	Range	Default
<pattern>	Discrete	{H|L}	H

Explanation
H: high level
L: low level

Return Format The query returns H or L.

Example
```
:TRIGger:SHOLd:PATTern L      /*Set the data type to L*/
:TRIGger:SHOLd:PATTern?       /*The query returns L*/
```

:TRIGger:SHOLd:TYPe

Syntax :TRIGger:SHOLd:TYPe <type>

:TRIGger:SHOLd:TYPe?

Description Set or query the setup type in setup/hold trigger.

Parameter

Name	Type	Range	Default
<type>	Discrete	{SETup|HOLd|SETHOLd}	SETup

Explanation

- SETup: the oscilloscope triggers when the setup time is less than the setting value (:TRIGger:SHOLd:STIMe).
- HOLd: the oscilloscope triggers when the hold time is less than the setting value (:TRIGger:SHOLd:HTIMe).
- SETHOLd: the oscilloscope triggers when the setup time or hold time is less than the corresponding setting value (:TRIGger:SHOLd:STIMe or :TRIGger:SHOLd:HTIMe).

Return Format The query returns SET, HOL, or SETHOL.

Example
```
:TRIGger:SHOLd:TYPe SETHOLd      /*Set the setup type to setup&hold*/
:TRIGger:SHOLd:TYPe?             /*The query returns SETHOL*/
```

:TRIGger:SHOLd:STIMe

Syntax :TRIGger:SHOLd:STIMe <NR3>

:TRIGger:SHOLd:STIMe?

Description Set or query the setup time in setup/hold trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<NR3>	Real	8ns to 1s	1μs

Explanation

- Setup time refers to the period of time that the data stays stable and constant before the clock signal of the trigger appears.
- This command is only available when the setup type (refer to :TRIGger:SHOLd:TYPe) is SETup or SETHOLd.

Return Format The query returns the setup time in scientific notation.

Example

```
:TRIGger:SHOLd:STIMe 0.002     /*Set the setup time to 2ms*/
:TRIGger:SHOLd:STIMe?          /*The query returns 2.000000e-03*/
```

Related Command :TRIGger:SHOLd:HTIMe

:TRIGger:SHOLd:HTIMe

Syntax :TRIGger:SHOLd:HTIMe <NR3>

:TRIGger:SHOLd:HTIMe?

Description Set or query the hold time in setup/hold trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<NR3>	Real	8ns to 1s	1μs

Explanation

- Hold time refers to the period of time that the data stays stable and constant after the clock signal of the trigger appears.
- This command is only available when the setup type (refer to :TRIGger:SHOLd:TYPe) is HOLd or SETHOLd.

Return Format The query returns the hold time in scientific notation.

Example

```
:TRIGger:SHOLd:HTIMe 0.002     /*Set the hold time to 2ms*/
:TRIGger:SHOLd:HTIMe?          /*The query returns 2.000000e-03*/
```

Related Command :TRIGger:SHOLd:STIMe

:TRIGger:NEDGe

Command List:

:TRIGger:NEDGe:SOURce

Syntax :TRIGger:NEDGe:SOURce <source>

:TRIGger:NEDGe:SOURce?

Description Set or query the trigger source in Nth edge trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example
```
:TRIGger:NEDGe:SOURce CHANnel2      /*Set the trigger source to CH2*/
:TRIGger:NEDGe:SOURce?              /*The query returns CHAN2*/
```

:TRIGger:NEDGe:SLOPe

Syntax :TRIGger:NEDGe:SLOPe <slope>

:TRIGger:NEDGe:SLOPe?

Description Set or query the edge type in Nth edge trigger.

Parameter

Name	Type	Range	Default
<slope>	Discrete	{POSitive|NEGative}	POSitive

Explanation

- POSitive: trigger on the rising edge of the input signal when the voltage level meets the specified trigger level.
- NEGative: trigger on the falling edge of the input signal when the voltage level meets the specified trigger level.

Return Format The query returns POS or NEG.

Example
```
:TRIGger:NEDGe:SLOPe NEGative       /*Set the edge type to falling edge*/
:TRIGger:NEDGe:SLOPe?               /*The query returns NEG*/
```

:TRIGger:NEDGe:IDLE

Syntax :TRIGger:NEDGe:IDLE <NR3>

:TRIGger:NEDGe:IDLE?

Description Set or query the idle time in Nth edge trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<NR3>	Real	16ns to 10s	1μs

Return Format The query returns the idle time in scientific notation.

Example
```
:TRIGger:NEDGe:IDLE 0.002     /*Set the idle time to 2ms*/
:TRIGger:NEDGe:IDLE?          /*The query returns 2.000000e-03*/
```

:TRIGger:NEDGe:EDGE

Syntax :TRIGger:NEDGe:EDGE <NR1>

:TRIGger:NEDGe:EDGE?

Description Set or query the number of edges in Nth edge trigger.

Parameter

Name	Type	Range	Default
<NR1>	Integer	1 to 65535	2

Return Format The query returns an integer between 1 and 65535.

Example
```
:TRIGger:NEDGe:EDGE 20      /*Set the number of edges to 20*/
:TRIGger:NEDGe:EDGE?        /*The query returns 20*/
```

:TRIGger:NEDGe:LEVel

Syntax :TRIGger:NEDGe:LEVel <level>

:TRIGger:NEDGe:LEVel?

Description Set or query the trigger level in Nth edge trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the trigger level in scientific notation.

Example
```
:TRIGger:NEDGe:LEVel 0.16     /*Set the trigger level to 160mV*/
:TRIGger:NEDGe:LEVel?         /*The query returns 1.600000e-01*/
```

:TRIGger:RS232

Command List:

:TRIGger:RS232:SOURce

Syntax :TRIGger:RS232:SOURce <source>

:TRIGger:RS232:SOURce?

Description Set or query the trigger source in RS232 trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example

```
:TRIGger:RS232:SOURce CHANnel2      /*Set the trigger source to CH2*/
:TRIGger:RS232:SOURce?              /*The query returns CHAN2*/
```

:TRIGger:RS232:WHEN

Syntax :TRIGger:RS232:WHEN <when>

:TRIGger:RS232:WHEN?

Description Set or query the trigger condition in RS232 trigger.

Parameter

Name	Type	Range	Default
<when>	Discrete	{STARt|ERRor|PARity|DATA}	STARt

Explanation

- STARt: trigger on the start frame position.
- ERRor: trigger when error frame is detected.
- PARity: trigger when check error is detected.
- DATA: trigger on the last bit of the preset data bits and even-odd check bits.

Return Format The query returns STAR, ERR, PAR, or DATA.

Example

```
:TRIGger:RS232:WHEN ERRor       /*Set the trigger condition to error frame*/
:TRIGger:RS232:WHEN?            /*The query returns ERR*/
```

Related Command :TRIGger:RS232:DATA

:TRIGger:RS232:PARity

Syntax :TRIGger:RS232:PARity <parity>

:TRIGger:RS232:PARity?

Description Set or query the parity type when the trigger condition is ERRor or PARity in RS232 trigger.

Parameter

Name	Type	Range	Default
<parity>	Discrete	{EVEN|ODD|NONE}	Error Frame: NONE Check Error: ODD

Explanation When the trigger condition is PARity, the parity type cannot be set to NONE. At this point, the default parity type is ODD.

Return Format The query returns EVEN, ODD, or NONE.

Example
```
:TRIGger:RS232:PARity EVEN      /*Set the parity type to even*/
:TRIGger:RS232:PARity?          /*The query returns EVEN*/
```

Related Command :TRIGger:RS232:WHEN

:TRIGger:RS232:STOP

Syntax :TRIGger:RS232:STOP <bit>

:TRIGger:RS232:STOP?

Description Set or query the stop bit when the trigger condition is ERRor in RS232 trigger.

Parameter

Name	Type	Range	Default
<bit>	Discrete	{1|2}	1

Return Format The query returns 1 or 2.

Example
```
:TRIGger:RS232:STOP 2      /*Set the stop bit to 2*/
:TRIGger:RS232:STOP?       /*The query returns 2*/
```

Related Command :TRIGger:RS232:WHEN

:TRIGger:RS232:DATA

Syntax :TRIGger:RS232:DATA <data>

:TRIGger:RS232:DATA?

Description Set or query the data when the trigger condition is DATA in RS232 trigger.

Parameter

Name	Type	Range	Default
<data>	Integer	0 to 2^n-1	90

Explanation In the expression 2^n-1, n is the current data bits and can be 5, 6, 7, or 8.

Return Format The query returns an integer.

Example
```
:TRIGger:RS232:DATA 10      /*Set the data to 10*/
:TRIGger:RS232:DATA?        /*The query returns 10*/
```

Related Commands :TRIGger:RS232:WIDTh

:TRIGger:RS232:WHEN

:TRIGger:RS232:WIDTh

Syntax :TRIGger:RS232:WIDTh <width>

:TRIGger:RS232:WIDTh?

Description Set or query the data bits when the trigger condition is DATA in RS232 trigger.

Parameter

Name	Type	Range	Default
<width>	Discrete	{5|6|7|8}	8

Return Format The query returns 5, 6, 7, or 8.

Example
```
:TRIGger:RS232:WIDTh 6      /*Set the data bits to 6*/
:TRIGger:RS232:WIDTh?       /*The query returns 6*/
```

Related Command :TRIGger:RS232:WHEN

:TRIGger:RS232:BAUD

Syntax :TRIGger:RS232:BAUD <baud_rate>

:TRIGger:RS232:BAUD?

Description Set or query the baud rate in RS232 trigger. The default unit is bps.

Parameter

Name	Type	Range	Default
<baud_rate>	Discrete	{2400|4800|9600|19200|38400|57600|115200|230400|460800|921600|1000000|USER}	9600

Return Format The query returns an integer or USER.

Example
```
:TRIGger:RS232:BAUD 4800      /*Set the baud rate to 4800bps*/
:TRIGger:RS232:BAUD?          /*The query returns 4800*/
```

Related Command :TRIGger:RS232:BUSer

:TRIGger:RS232:BUSer

Syntax :TRIGger:RS232:BUSer <user baud>

:TRIGger:RS232:BUSer?

Description Set or query the user-defined baud rate in RS232 trigger. The default unit is bps.

Parameter

Name	Type	Range	Default
<user baud>	Integer	110 to 20000000	9600

Return Format The query returns an integer.

Example
```
:TRIGger:RS232:BUSer 50000      /*Set the user-defined baud rate to 50000bps*/
:TRIGger:RS232:BUSer?           /*The query returns 50000*/
```

Related Command :TRIGger:RS232:BAUD

:TRIGger:RS232:LEVel

Syntax :TRIGger:RS232:LEVel <level>

:TRIGger:RS232:LEVel?

Description Set or query the trigger level in RS232 trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the trigger level in scientific notation.

Example
```
:TRIGger:RS232:LEVel 0.16     /*Set the trigger level to 160mV*/
:TRIGger:RS232:LEVel?         /*The query returns 1.600000e-01*/
```

:TRIGger:IIC

Command List:

:TRIGger:IIC:SCL

Syntax :TRIGger:IIC:SCL <source>

:TRIGger:IIC:SCL?

Description Set or query the channel source of SCL in I2C trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example

```
:TRIGger:IIC:SCL CHANnel2      /*Set the SCL source to CH2*/
:TRIGger:IIC:SCL?              /*The query returns CHAN2*/
```

:TRIGger:IIC:SDA

Syntax :TRIGger:IIC:SDA <source>

:TRIGger:IIC:SDA?

Description Set or query the channel source of SDA in I2C trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel2

Return Format The query returns CHAN1 or CHAN2.

Example

```
:TRIGger:IIC:SDA CHANnel2      /*Set the SDA source to CH2*/
:TRIGger:IIC:SDA?              /*The query returns CHAN2*/
```

:TRIGger:IIC:WHEN

Syntax :TRIGger:IIC:WHEN <trig_type>

:TRIGger:IIC:WHEN?

Description Set or query the trigger condition in I2C trigger.

Parameter

Name	Type	Range	Default
<trig_type>	Discrete	{STARt|RESTart|STOP|NACKnowledge|ADDRess|DATA|ADATa}	STARt

Explanation

- STARt: trigger when the SDA data transitions from high to low while the SCL is high.
- RESTart: trigger when another start condition occurs before a stop condition.
- STOP: trigger when the SDA data transitions from low to high while the SCL is high.
- NACKnowledge: trigger when the SDA data is high during any acknowledgement of the SCL clock position.
- ADDRess: search for the specified address value and trigger on the read/write bit.
- DATA: search for the specified data value on the data line (SDA) and trigger on the jump edge of the clock line (SCL) corresponding to the last bit of the data.
- ADATa:search for the specified address value and data value at the same time and trigger when the "Address" and "Data" conditions are met at the same time.

Return Format The query returns STAR, STOP, NACK, REST, ADDR, DATA, or ADAT.

Example

```
:TRIGger:IIC:WHEN RESTart      /*Set the trigger condition to restart*/
:TRIGger:IIC:WHEN?             /*The query returns REST*/
```

:TRIGger:IIC:AWIDth

Syntax :TRIGger:IIC:AWIDth <bits>

:TRIGger:IIC:AWIDth?

Description Set or query the address bits when the trigger condition is ADDRess or ADATa in I2C trigger.

Parameter

Name	Type	Range	Default
<bits>	Discrete	{7|8|10}	7

Return Format The query returns 7, 8, or 10.

Example

```
:TRIGger:IIC:AWIDth 10      /*Set the address bits to 10*/
:TRIGger:IIC:AWIDth?        /*The query returns 10*/
```

Related Commands :TRIGger:IIC:DIRection

:TRIGger:IIC:ADDRess

:TRIGger:IIC:ADDRess

Syntax :TRIGger:IIC:ADDRess <adr>

:TRIGger:IIC:ADDRess?

Description Set or query the address when the trigger condition is ADDRess or ADATa in I2C trigger.

Parameter

Name	Type	Range	Default
<adr>	Integer	0 to 2^n-1: 0 to 127, 0 to 255, or 0 to 1023	1

Explanation In the expression 2^n-1, n is the current address bits.

Return Format The query returns an integer.

Example
```
:TRIGger:IIC:ADDRess 100      /*Set the address to 100*/
:TRIGger:IIC:ADDRess?         /*The query returns 100*/
```

Related Command :TRIGger:IIC:AWIDth

:TRIGger:IIC:DIRection

Syntax :TRIGger:IIC:DIRection <dir>

:TRIGger:IIC:DIRection?

Description Set or query the data direction when the trigger condition is ADDRess or ADATa in I2C trigger.

Parameter

Name	Type	Range	Default
<dir>	Discrete	{READ|WRITe|RWRite}	READ

Explanation This command is invalid when the address bits is 8.

Return Format The query returns READ, WRIT, or RWR.

Example
```
:TRIGger:IIC:DIRection RWRite     /*Set the data direction to read/write*/
:TRIGger:IIC:DIRection?           /*The query returns RWR*/
```

Related Command :TRIGger:IIC:AWIDth

:TRIGger:IIC:DATA

Syntax :TRIGger:IIC:DATA <data>

:TRIGger:IIC:DATA?

Description Set or query the data when the trigger condition is DATA or ADATa in I2C trigger.

Parameter

Name	Type	Range	Default
<data>	Integer	0 to 2^{40}-1	82

Explanation The range of <data> is related to the byte length. The maximum byte length is 5, namely 40 bit binary data. Thus, the range of <data> is from 0 to 2^{40}-1.

Return Format The query returns an integer.

Example
```
:TRIGger:IIC:DATA 64      /*Set the data to 64*/
:TRIGger:IIC:DATA?        /*The query returns 64*/
```

Related Command :TRIGger:IIC:WHEN

:TRIGger:IIC:CLEVel

Syntax :TRIGger:IIC:CLEVel <level>

:TRIGger:IIC:CLEVel?

Description Set or query the trigger level of SCL in I2C trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the trigger level of SCL in scientific notation.

Example
```
:TRIGger:IIC:CLEVel 0.16    /*Set the trigger level of SCL to 160mV*/
:TRIGger:IIC:CLEVel?        /*The query returns 1.600000e-01*/
```

:TRIGger:IIC:DLEVel

Syntax :TRIGger:IIC:DLEVel <level>

:TRIGger:IIC:DLEVel?

Description Set or query the trigger level of SDA in I2C trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the trigger level of SDA in scientific notation.

Example
```
:TRIGger:IIC:DLEVel 0.16    /*Set the trigger level of SDA to 160mV*/
:TRIGger:IIC:DLEVel?        /*The query returns 1.600000e-01*/
```

:TRIGger:SPI

Command List:

:TRIGger:SPI:SCL

Syntax :TRIGger:SPI:SCL <source>

:TRIGger:SPI:SCL?

Description Set or query the channel source of SCL in SPI trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example
```
:TRIGger:SPI:SCL CHANnel1     /*Set the channel source of SCL to CH1*/
:TRIGger:SPI:SCL?             /*The query returns CHAN1*/
```

:TRIGger:SPI:SDA

Syntax :TRIGger:SPI:SDA <source>

:TRIGger:SPI:SDA?

Description Set or query the channel source of SDA in SPI trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel2

Return Format The query returns CHAN1 or CHAN2.

Example
```
:TRIGger:SPI:SDA CHANnel2     /*Set the channel source of SDA to CH2*/
:TRIGger:SPI:SDA?             /*The query returns CHAN2*/
```

:TRIGger:SPI:WHEN

Syntax :TRIGger:SPI:WHEN <trig_type>

:TRIGger:SPI:WHEN?

Description Set or query the trigger condition in SPI trigger.

Parameter

Name	Type	Range	Default
<trig_type>	Discrete	{CS|TIMeout}	CS

Explanation When the trigger condition is set to TIMeout, you can use the :TRIGger:SPI:TIMeout command to set the timeout value.

Return Format The query returns CS or TIM.

Example
```
:TRIGger:SPI:WHEN TIMeout      /*Set the trigger condition to TIMeout*/
:TRIGger:SPI:WHEN?             /*The query returns TIM*/
```

:TRIGger:SPI:WIDTh

Syntax :TRIGger:SPI:WIDTh <width>

:TRIGger:SPI:WIDTh?

Description Set or query the data bits of the SDA channel in SPI trigger.

Parameter

Name	Type	Range	Default
<width>	Integer	4 to 32	8

Return Format The query returns an integer.

Example
```
:TRIGger:SPI:WIDTh 10   /*Set the data bits of the SDA channel to 10*/
:TRIGger:SPI:WIDTh?     /*The query returns 10*/
```

Related Command :TRIGger:SPI:DATA

:TRIGger:SPI:DATA

Syntax :TRIGger:SPI:DATA <data>

:TRIGger:SPI:DATA?

Description Set or query the data in SPI trigger.

Parameter

Name	Type	Range	Default
<data>	Integer	0 to 2^{32}-1	82

Explanation The range of <data> is related to the data bits. The maximum data bits is 32. Thus, the range of <data> is from 0 to 2^{32}-1.

Return Format The query returns an integer.

Example
```
:TRIGger:SPI:DATA 5     /*Set the data to 5*/
:TRIGger:SPI:DATA?      /*The query returns 5*/
```

Related Command :TRIGger:SPI:WIDTh

:TRIGger:SPI:TIMeout

Syntax :TRIGger:SPI:TIMeout <time_value>

:TRIGger:SPI:TIMeout?

Description Set or query the timeout value when the trigger condition is TIMeout in SPI trigger. The default unit is s.

Parameter

Name	Type	Range	Default
<time_value>	Real	100ns to 1s	1μs

Return Format The query returns the timeout value in scientific notation.

Example
```
:TRIGger:SPI:TIMeout 0.001    /*Set the timeout value to 1ms*/
:TRIGger:SPI:TIMeout?         /*The query returns 1.000000e-03*/
```

Related Command :TRIGger:SPI:WHEN

:TRIGger:SPI:SLOPe

Syntax :TRIGger:SPI:SLOPe <slope>

:TRIGger:SPI:SLOPe?

Description Set or query the clock edge in SPI trigger.

Parameter

Name	Type	Range	Default
<slope>	Discrete	{POSitive|NEGative}	POSitive

Explanation POSitive: sample the SDA data on the rising edge of the clock.
NEGative: sample the SDA data on the falling edge of the clock.

Return Format The query returns POS or NEG.

Example
```
:TRIGger:SPI:SLOPe POSitive    /*Set the clock edge to POSitive*/
:TRIGger:SPI:SLOPe?            /*The query returns POS*/
```

:TRIGger:SPI:CLEVel

Syntax :TRIGger:SPI:CLEVel <level>

:TRIGger:SPI:CLEVel?

Description Set or query the trigger level of the SCL channel in SPI trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the trigger level of the SCL channel in scientific notation.

Example
```
:TRIGger:SPI:CLEVel 0.16    /*Set the trigger level of the SCL channel to 160mV*/
:TRIGger:SPI:CLEVel?        /*The query returns 1.600000e-01*/
```

:TRIGger:SPI:DLEVel

Syntax :TRIGger:SPI:DLEVel <level>

:TRIGger:SPI:DLEVel?

Description Set or query the trigger level of the SDA channel in SPI trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the trigger level of the SDA channel in scientific notation.

Example
```
:TRIGger:SPI:DLEVel 0.16     /*Set the trigger level of the SDA channel to 160mV*/
:TRIGger:SPI:DLEVel?     /*The query returns 1.600000e-01*/
```

:TRIGger:SPI:SLEVel

Syntax :TRIGger:SPI:SLEVel <level>

:TRIGger:SPI:SLEVel?

Description Set or query the trigger level of the CS channel in SPI trigger. The unit is the same as the current amplitude unit.

Parameter

Name	Type	Range	Default
<level>	Real	(-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)	0

Explanation For VerticalScale, refer to :CHANnel<n>:SCALe. For OFFSet, refer to :CHANnel<n>:OFFSet.

Return Format The query returns the trigger level of the CS channel in scientific notation.

Example
```
:TRIGger:SPI:SLEVel 0.16     /*Set the trigger level of the CS channel to 160mV*/
:TRIGger:SPI:SLEVel?         /*The query returns 1.600000e-01*/
```

:TRIGger:SPI:MODE

Syntax :TRIGger:SPI:MODE <mode>

:TRIGger:SPI:MODE?

Description Set or query the CS mode when the trigger condition is CS in SPI trigger.

Parameter

Name	Type	Range	Default
<mode>	Discrete	{HIGH|LOW}	LOW

Return Format The query returns HIGH or LOW.

Example
```
:TRIGger:SPI:MODE LOW     /*Set the CS mode to low*/
:TRIGger:SPI:MODE?        /*The query returns LOW*/
```

Related Command :TRIGger:SPI:WHEN

:TRIGger:SPI:CS

Syntax :TRIGger:SPI:CS <source>

:TRIGger:SPI:CS?

Description Set or query the data source of the CS signal in SPI trigger.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2}	CHANnel1

Return Format The query returns CHAN1 or CHAN2.

Example

```
:TRIGger:SPI:CS CHANnel2     /*Set the data source of the CS signal to CH2*/
:TRIGger:SPI:CS?             /*The query returns CHAN2*/
```

:WAVeform Commands

The :WAVeform commands are used to read the waveform data and its related settings. :WAVeform:MODE is used to set the reading mode of the waveform data. In different mode, the definition of each parameter is different, as shown in Figure 2-1 and Figure 2-2.

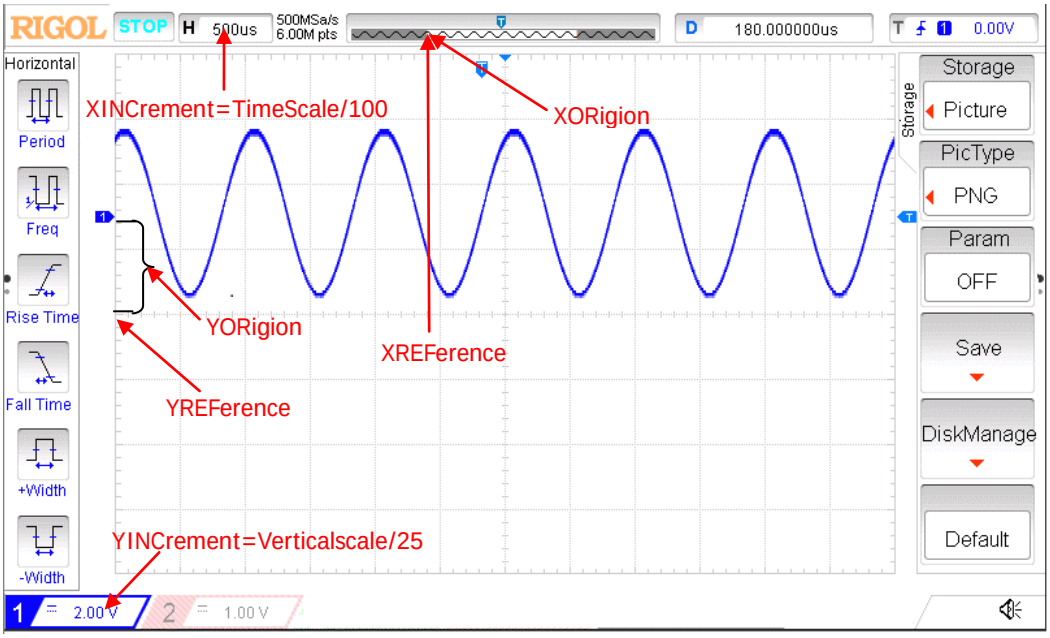


Figure 2-1 Parameter Definitions in NORMAL Mode

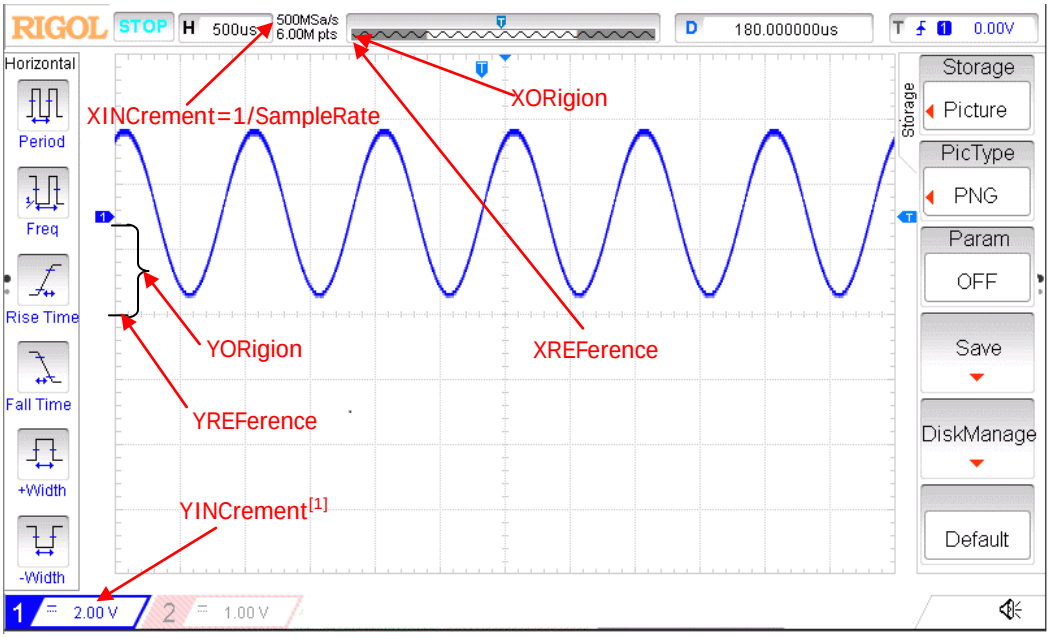


Figure 2-2 Paramter Definitions in RAW Mode

Note[1]: In RAW mode, YINCrement is related to the Verticalscale of the internal waveform and the Verticalscale currently selected.

Command List:

:WAVeform:SOURce

Syntax :WAVeform:SOURce <source>

:WAVeform:SOURce?

Description Set or query the channel of which the waveform data will be read.

Parameter

Name	Type	Range	Default
<source>	Discrete	{CHANnel1|CHANnel2|MATH}	CHANnel1

Explanation If the MATH channel is selected, only NORMal can be selected in :WAVeform:MODE.

Return Format The query returns CHAN1, CHAN2, or MATH.

Example

```
:WAVeform:SOURce CHANnel2      /*Set the channel to CH2*/
:WAVeform:SOURce?              /*The query returns CHAN2*/
```

Related Command :WAVeform:FORMat

:WAVeform:MODE

Syntax :WAVeform:MODE <mode>

:WAVeform:MODE?

Description Set or query the reading mode used by :WAVeform:DATA?.

Parameter

Name	Type	Range	Default
<mode>	Discrete	{NORMal|MAXimum|RAW}	NORMal

Explanation

- NORMal: read the waveform data displayed on the screen.
- MAXimum: read the waveform data displayed on the screen when the instrument is in the run state and the waveform data in the internal memory in the stop state.

- RAW: read the waveform data in the internal memory. Note that the waveform data in the internal memory can only be read when the oscilloscope is in the stop state and the oscilloscope cannot be operated during the reading process.
- If the MATH channel is selected, only the NORMal mode is valid.

Return Format The query returns NORM, MAX, or RAW.

Example
```
:WAVeform:MODE RAW    /*Set the reading mode to RAW*/
:WAVeform:MODE?       /*The query returns RAW*/
```

:WAVeform:FORMat

Syntax :WAVeform:FORMat <format>

:WAVeform:FORMat?

Description Set or query the return format of the waveform data.

Parameter

Name	Type	Range	Default
<format>	Discrete	{WORD|BYTE|ASCii}	BYTE

Explanation
- WORD: a waveform point occupies two bytes (namely 16 bits) in which the lower 8 bits are valid and the higher 8 bits are 0.
- BYTE: a waveform point occupies one byte (namely 8 bits).
- ASCii: return the actual voltage value of each waveform point in scientific notation. The voltage values are separated by commas.

Return Format The query returns WORD, BYTE, or ASC.

Example
```
:WAVeform:FORMat WORD    /*Set the return format to WORD*/
:WAVeform:FORMat?        /*The query returns WORD*/
```

:WAVeform:DATA?

Syntax :WAVeform:DATA?

Description Read the waveform data.

Explanation
- Reading procedures of the screen waveform data:

S1.	:WAV:SOUR CHAN1	Set the channel source to CH1
S2.	:WAV:MODE NORM	Set the waveform reading mode to NORMal
S3.	:WAV:FORM BYTE	Set the return format of the waveform data to BYTE
S4.	:WAV:DATA?	Read the screen waveform data

- Reading procedures of the internal memory waveform data:

Tip:
When reading the waveform data in the internal memory, the maximum number of waveform points can be read each time the :WAV:DATA? command is sent is related to the return format of the waveform data currently selected, as shown in the table below.

Return Format of the Waveform Data	Maximum Number of Waveform Points can be Read Each Time
BYTE	250000
WORD	125000
ASCii	15625

Before reading the waveform data in the internal memory, you need to judge whether the waveform data can all be read at one time according to the memory depth of the oscilloscope and the maximum number of waveform points that can be read each time (refer to the table above).

1) When the memory depth of the oscilloscope is lower than or equal to the maximum number of waveform points that can be read each time, the waveform data in the internal memory can all be read at one time by specifying the start point and stop point.
2) When the memory depth of the oscilloscope is greater than the maximum number of waveform points that can be read each time, the waveform data in the internal memory need to be read in several batches by specifying the start point and stop point. Each time, only the waveform data in one area of the internal memory is read (the waveform data of two neighbouring areas are continuous); then, you need to combine the waveform data that are read separately in sequence.

Example 1 (read the internal memory waveform data at one time):
Assume that the memory depth of the oscilloscope is 120kpts and the return format of the waveform data is BYTE; at this point, the memory depth of the oscilloscope is lower than the maximum number of waveform points that can be read each time (250000) and you can read all the waveform data in the internal memory at one time.

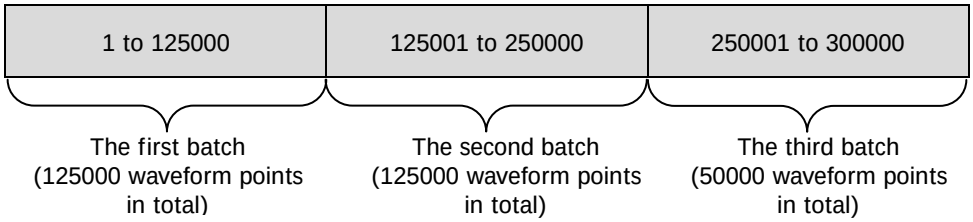
The reading procedures are as follows.

S1. :STOP	Set the instrument to STOP state (you can only read the waveform data in the internal memory when the oscilloscope is in STOP state)
S2. :WAV:SOUR CHAN1	Set the channel source to CH1
S3. :WAV:MODE RAW	Set the waveform reading mode to RAW
S4. :WAV:FORM BYTE	Set the return format of the waveform data to BYTE
S5. :WAV:STAR 1	Set the start point of waveform data reading to the first waveform point
S6. :WAV:STOP 120000	Set the stop point of waveform data reading to the 120000th waveform point (the last point)
S7. :WAV:DATA?	Read the waveform data in the internal memory (all the points)

Example 2 (read the internal memory waveform data in several batches):
Assume that the memory depth of the oscilloscope is 300kpts and the return format of the waveform data is WORD; at this point, the memory depth of the oscilloscope is greater than the maximum number of waveform points that can be read each time (125000) and you need to read all the waveform data in the

internal memory in at least 3 batches (300000/125000=2.4).

When reading the waveform data in batches, note that the number of waveform points in the internal memory that is read each time (namely, stop point - start point + 1) cannot be greater than the maximum number of points that can be read each time (in this example, it is 125000); then, you need to combine the waveform data that are read separately in sequence. The figure below shows an example of the start point and stop point settings when reading the above waveform data in the internal memory in 3 batches.



The corresponding reading procedures are as follows.

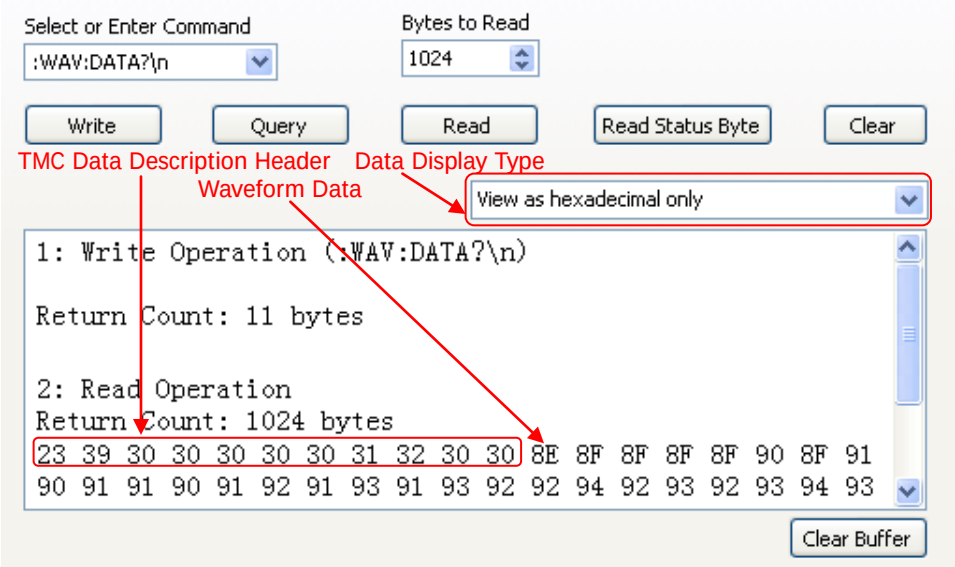
S1. :STOP	Set the instrument to STOP state (you can only read the waveform data in the internal memory when the oscilloscope is in STOP state)
S2. :WAV:SOUR CHAN1	Set the channel source to CH1
S3. :WAV:MODE RAW	Set the waveform reading mode to RAW
S4. :WAV:FORM WORD	Set the return format of the waveform data to WORD
Perform the first reading operation	
S5. :WAV:STAR 1	Set the start point of the first reading operation to the first waveform point
S6. :WAV:STOP 125000	Set the stop point of the first reading operation to the 125000th waveform point
S7. :WAV:DATA?	Read the data from the first waveform point to the 125000th waveform point
Perform the second reading operation	
S8. :WAV:STAR 125001	Set the start point of the second reading operation to the 125001th waveform point
S9. :WAV:STOP 250000	Set the stop point of the second reading operation to the 250000th waveform point
S10. :WAV:DATA?	Read the data from the 125001th waveform point to the 250000th waveform point
Perform the third reading operation	
S11. :WAV:STAR 250001	Set the start point of the third reading operation to the 250001th waveform point
S12. :WAV:STOP 300000	Set the stop point of the third reading operation to the 300000th waveform point (the last point)
S13. :WAV:DATA?	Read the data from the 250001th waveform point to the 300000th waveform point (the last point)
Finally, combine the data read in the 3 reading operations in sequence.	

Return Format

- The return format is related to the return format of the waveform data currently selected (:WAVeform:FORMat).
  WORD or BYTE format: the data returned consists of 2 parts, including the TMC data description header and the waveform data. The format of the TMC data description header is #NXXXXXXXXX; wherein, # is the denoter, N is 9 and the 9 data following it denote the number of bytes of the waveform data. The TMC data description header is followed by the waveform data.
  ASCii format: the query returns the actual voltage value of each waveform point in scientific notation; the voltage values are separated by ",".
- When reading the waveform data in the internal memory in several batches, the

data returned each time is the data in one area in the internal memory. Each of the data returned contains the TMC data description header (WORD or BYTE format) and the waveform data from two adjacent areas are continuous.

- The figure below shows the waveform data read (in BYTE format). First, select "View as hexadecimal only" in the dropdown list at the right side. At this point, the waveform data read is displayed in hexadecimal form with the first 11 bytes being the TMC data description header and the others starting from the 12th byte (namely 8E) being the waveform data. Users can convert the waveform data read to the voltage value of each waveform point using **(0x8E - YORigin - YREFerence) x YINCrement**.



Related Commands
:WAVeform:MODE
:WAVeform:YINCrement?
:WAVeform:YREFerence?
:WAVeform:YORigin?

:WAVeform:XINCrement?

Syntax :WAVeform:XINCrement?

Description Query the time difference between two neighboring points of the specified channel source in the X direction.

Explanation
- The returned value is related to the current data reading mode:
  In the NORMal mode, XINCrement = TimeScale/100.
  In the RAW mode, XINCrement = 1/SampleRate.
  In MAX mode, XINCrement = TimeScale/100 when the instrument is in running status; XINCrement = 1/SampleRate when the instrument is in stop status.
- The unit is related to the current channel source:
  When the channel source is CHANnel1 or CHANnel2, the unit is s. When the channel source is MATH and the operation type is FFT, the unit is Hz.

Return Format The query returns the XINCrement in scientific notation.

Example :WAVeform:XINCrement? /*The query returns 1.000000e-08*/

Related Command :WAVeform:SOURce

:WAVeform:XORigin?

Syntax :WAVeform:XORigin?

Description Query the start time of the waveform data of the channel source currently selected in the X direction.

Explanation

- The return value is related to the current data reading mode:
  In NORMal mode, the query returns the start time of the waveform data displayed on the screen.
  In RAW mode, the query returns the start time of the waveform data in the internal memory.
  In MAX mode, the query returns the start time of the waveform data displayed on the screen when the instrument is in running status; the query returns the start time of the waveform data in the internal memory when the instrument is in stop status.
- The unit is related to the current channel source:
  When the channel source is CHANnel1 or CHANnel2, the unit is s.
  When the channel source is MATH and the operation type is FFT, the unit is Hz.

Return Format The query returns the XORigin in scientific notation.

Example :WAVeform:XORigin? /*The query returns -6.000000e-06*/

Related Commands :WAVeform:SOURce
:WAVeform:XREFerence?

:WAVeform:XREFerence?

Syntax :WAVeform:XREFerence?

Description Query the reference time of the specified channel source in the X direction.

Return Format The query returns 0 (namely the first point on the screen or in the internal memory).

Related Command :WAVeform:SOURce

:WAVeform:YINCrement?

Syntax :WAVeform:YINCrement?

Description Query the waveform increment of the specified channel source in the Y direction. The unit is the same as the current amplitude unit.

Explanation The return value is related to the current data reading mode:
In NORMal mode, YINCrement = VerticalScale/25.
In RAW mode, YINCrement is related to the Verticalscale of the internal waveform and the Verticalscale currently selected.
In MAX mode, YINCrement = VerticalScale/25 when the instrument is in running status; YINCrement is related to the Verticalscale of the internal waveform and the Verticalscale currently selected when the instrument is in stop status.

Return Format The query returns the YINCrement in scientific notation.

Example :WAVeform:YINCrement? /*The query returns 8.000000e-03*/

Related Command :WAVeform:SOURce

:WAVeform:YORigin?

Syntax :WAVeform:YORigin?

Description Query the vertical offset relative to the vertical reference position of the specified channel source in the Y direction.

Explanation The return value is related to the current data reading mode:
In NORMal mode, YORigin = VerticalOffset/YINCrement.
In RAW mode, YORigin is related to the Verticalscale of the internal waveform and the Verticalscale currently selected.
In MAX mode, YORigin = VerticalOffset/YINCrement when the instrument is in running status; YORigin is related to the Verticalscale of the internal waveform and the Verticalscale currently selected when the instrument is in stop status.

Return Format The query returns an integer.

Example
```
:WAVeform:YORigin?     /*The query returns 25*/
```

Related Commands :WAVeform:SOURce
:WAVeform:YREFerence?

:WAVeform:YREFerence?

Syntax :WAVeform:YREFerence?

Description Query the vertical reference position of the specified channel source in the Y direction.

Explanation YREFerence is always 127 (the screen bottom is 0 and the screen top is 255).

Return Format The query returns 127.

Related Command :WAVeform:SOURce

:WAVeform:STARt

Syntax
```
:WAVeform:STARt <sta>
:WAVeform:STARt?
```

Description Set or query the start point of waveform data reading.

Parameter

Name	Type	Range	Default
<sta>	Integer	NORMal: 1 to 1200 MAX: 1 to the number of effective points currently on the screen RAW: 1 to the current maximum memory depth	1

Explanation When reading the waveform data in the internal memory, the actual settable ranges of the start point and stop point of a reading operation are related to the memory depth of the oscilloscope and the return format of the waveform data currently selected; for the details, refer to the explanation in the :WAVeform:DATA? command.

Return Format The query returns an integer.

Example
```
:WAVeform:STARt 100     /*Set the start point to 100*/
:WAVeform:STARt?        /*The query returns 100*/
```

Related :ACQuire:MDEPth

Commands

:WAVeform:MODE

:WAVeform:FORMat

:WAVeform:STOP

:WAVeform:STOP

Syntax

:WAVeform:STOP <stop>

:WAVeform:STOP?

Description

Set or query the stop point of waveform data reading.

Parameter

Name	Type	Range	Default
<stop>	Integer	NORMal: 1 to 1200 MAX: 1 to the number of effective point currently on the screen RAW: 1 to the current maximum memory depth	1200

Explanation

When reading the waveform data in the internal memory, the actual settable ranges of the start point and stop point of a reading operation are related to the memory depth of the oscilloscope and the return format of the waveform data currently selected; for the details, refer to the explanation in the :WAVeform:DATA? command.

Return Format

The query returns an integer.

Example

```
:WAVeform:STOP 500     /*Set the stop point to 500*/
:WAVeform:STOP?        /*The query returns 500*/
```

Related Commands

:ACQuire:MDEPth

:WAVeform:MODE

:WAVeform:FORMat

:WAVeform:STARt

:WAVeform:PREamble?

Syntax

:WAVeform:PREamble?

Description

Query and return all the waveform parameters.

Return Format

The query returns 10 waveform parameters separated by ",":

<format>,<type>,<points>,<count>,<xincrement>,<xorigin>,<xreference>,<yincrement>,<yorigin>,<yreference>

Wherein,
<format>: 0 (BYTE), 1 (WORD) or 2 (ASC).
<type>: 0 (NORMal), 1 (MAXimum) or 2 (RAW).
<points>: an integer between 1 and 24000000.
<count>: the number of averages in the average sample mode and 1 in other modes.
<xincrement>: the time difference between two neighboring points in the X direction.
<xorigin>: the start time of the waveform data in the X direction.
<xreference>: the reference time of the data point in the X direction.
<yincrement>: the waveform increment in the Y direction.
<yorigin>: the vertical offset relative to the "Vertical Reference Position" in the Y direction.

<yreference>: the vertical reference position in the Y direction.

Example :WAVeform:PREamble? /*The query returns 0,2,6000000,1,1.000000e-09,-3.000000e-03,0,4.132813e-01,0,122*/

Related Commands

:WAVeform:FORMat

:WAVeform:MODE

:ACQuire:AVERages

:WAVeform:XINCrement?

:WAVeform:XREFerence?

:WAVeform:YINCrement?

:WAVeform:YORigin?

:WAVeform:YREFerence?

Chapter 3 Programming Demos

This chapter lists some programming demos to illustrate how to use commands to realize the common functions of the oscilloscope in the development environments of Excel, Matlab, LabVIEW, Visual Basic 6.0 and Visual C++ 6.0. All the demos are based on VISA (Virtual Instrument Software Architecture).

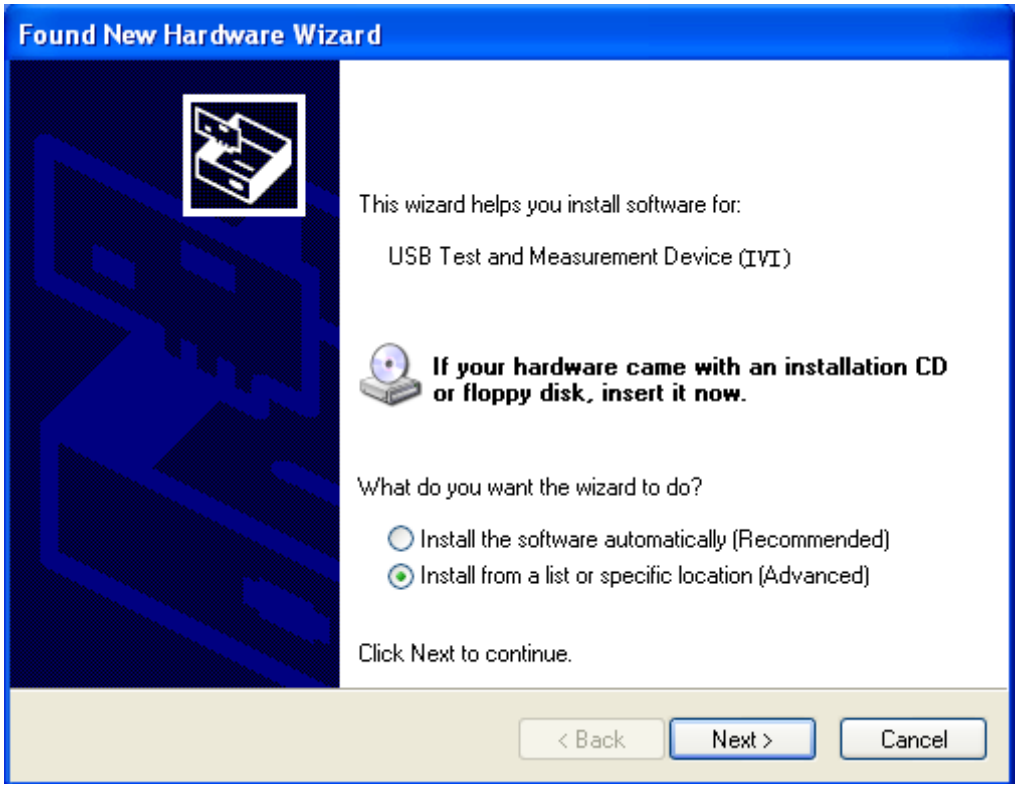
The main topics of this chapter:

Programming Preparations

Before programming, you need to make the following preparations:

First, install the Ultra Sigma common PC software. You can log in **RIGOL** official website (www.rigol.com) to download the software; then, install the software according to the instructions. When Ultra Sigma is installed, the NI-VISA library is installed automatically. Here, the default installation path is C:\Program Files\IVI Foundation\VISA.

Here, the USB interface of the oscilloscope is used to communicate with the PC and please use a USB cable to connect the USB Device interface on the rear panel of the oscilloscope to the PC. After successful connection, power on and start the instrument (make sure that **Utility** → **IO Setting** → **USB Device** is currently set to "Computer"). At this point, the "**Found New Hardware Wizard**" dialog box appears on the PC at the first connection. Please follow the instructions to install the "USB Test and Measurement Device (IVI)".



By now, the programming preparations are finished. In the following part, detailed introductions are given about the programming demos in the Excel, Matlab, Labview, Visual Basic 6.0, and Visual C++ 6.0 development environments.

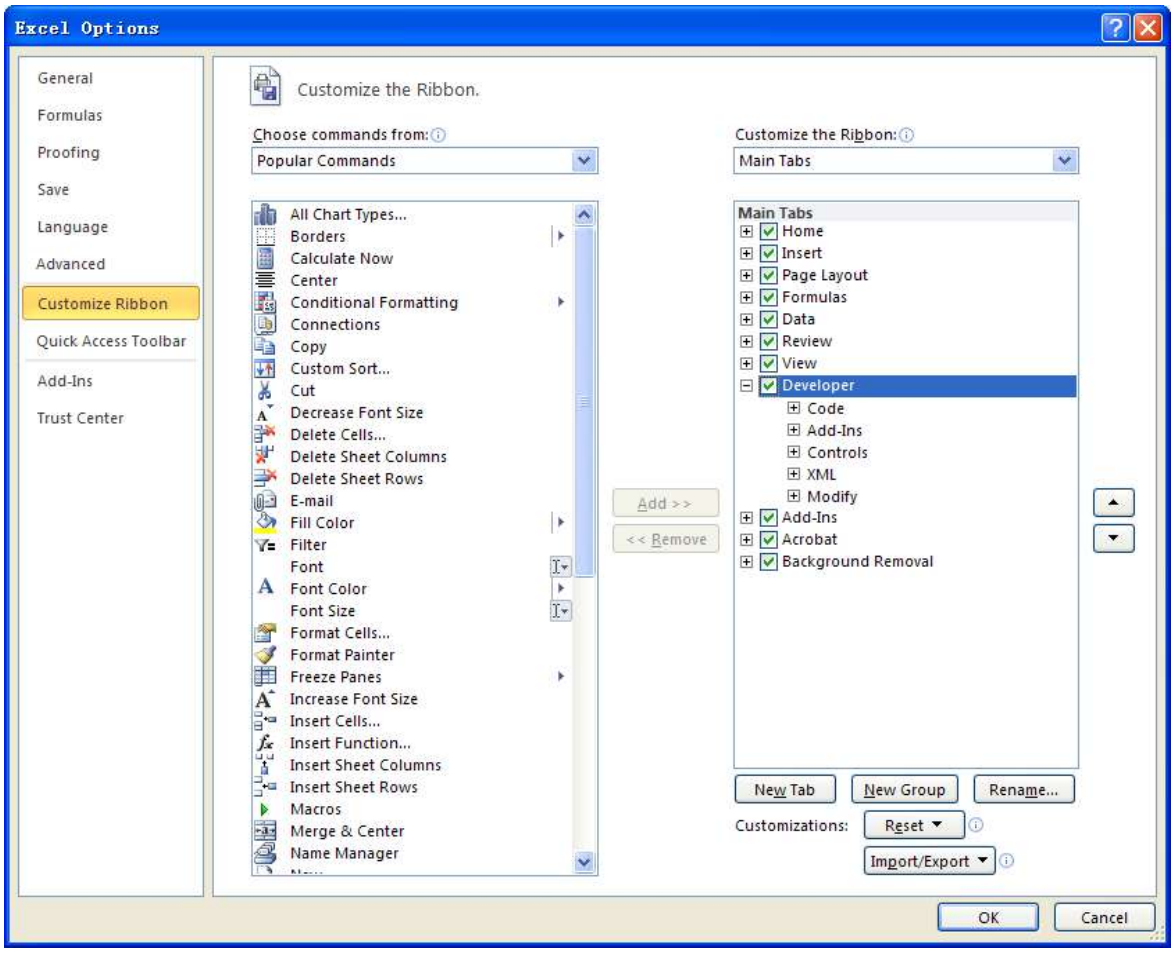
Excel Programming Demo

The program used in this demo: Microsoft Excel 2010

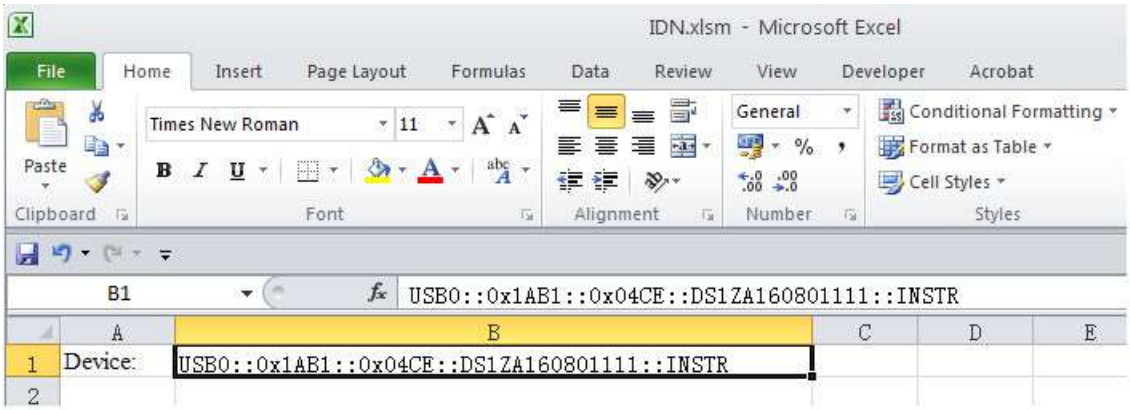
The function realized in this demo: send the *IDN? Command to read the device information.

1. Create a new Excel file that enables the Macros. In this demo, the file is named as IDN.xlsm.

2. Run the IDN.xlsm file. Click **File→Options** at the upper-left corner of the Excel file to open the interface as shown in the figure below. Click **Customize Ribbon** at the right, check **Developer** and click **OK**. At this point, the Excel menu bar displays the **Developer** menu.

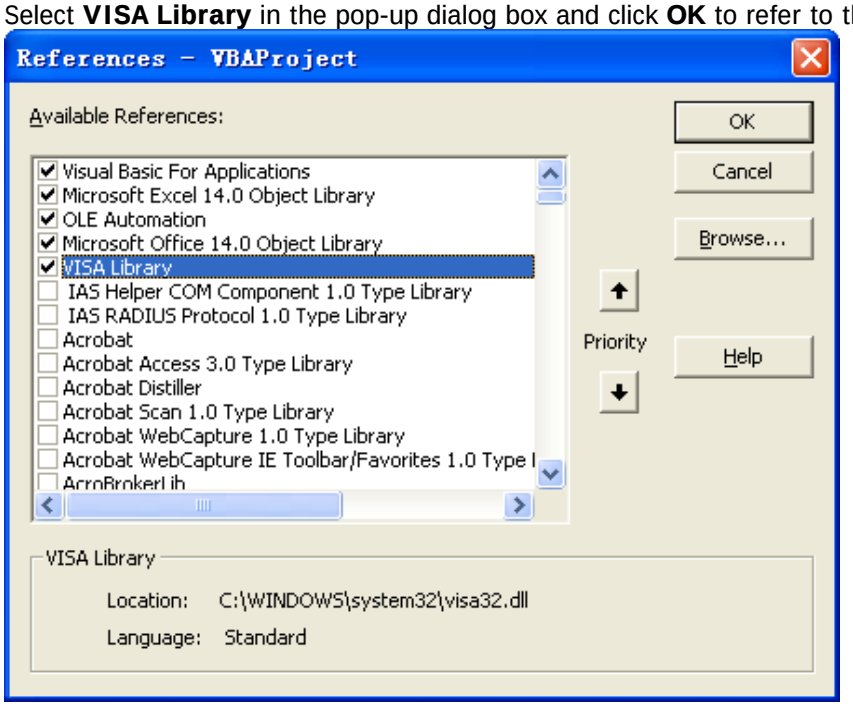


3. Enter the VISA descriptor into a cell of the file as shown in the figure below. For example, the VISA descriptor is USB0::0x1AB1::0x04CE::DS1ZA160801111::INSTR. Input it into SHEET1.CELLS(1,2) (namely the B1 cell in Sheet1). Click the **Developer** menu and select the **Visual Basic** option to open the Microsoft Visual Basic.



4. Select **Tools(T)** in the Microsoft Visual Basic menu bar and click **References**.

Select **VISA Library** in the pop-up dialog box and click **OK** to refer to the VISA Library.

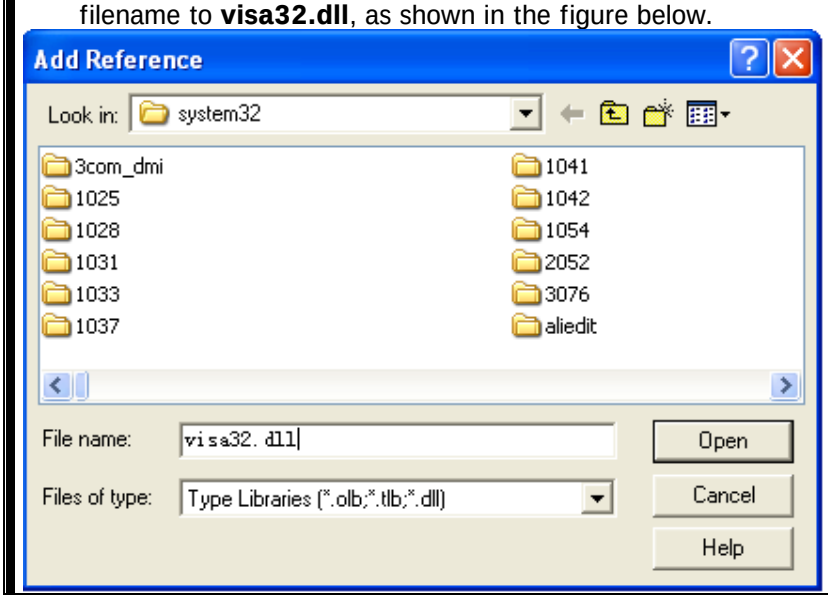


Explanation:

If you cannot find the VISA Library in the list at the left of the figure above, please follow the method below to find it.

(1) Make sure that your PC has installed the NI-VISA library.

(2) Click **Browse...** at the right and set the search range to **C:\WINDOWS\system32** and the filename to **visa32.dll**, as shown in the figure below.

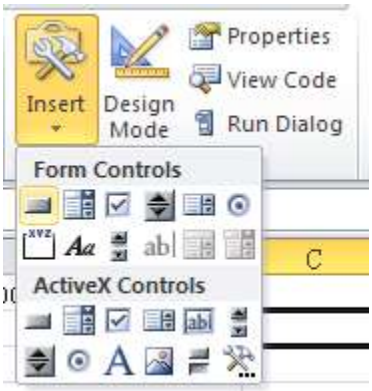


5. Click **View Code** in the **Developer** menu to enter the Microsoft Visual Basic interface. Add the following codes and save the file.

Note: If the Excel file created at step 2 does not enable the Macros, at this point, the prompt message "The following features cannot be saved in macro-free workbooks" will be displayed. In this situation, please save the Excel file as a file using the Macros (the filename suffix changes to .xlsm).

```
Sub QueryIdn()

    Dim viDefRm As Long
    Dim viDevice As Long
    Dim viErr As Long
    Dim cmdStr As String
    Dim idnStr As String * 128
    Dim ret As Long

    'Turn on the device, the device resource descriptor is in CELLS(1,2) of SHEET1'
    viErr = visa.viOpenDefaultRM(viDefRm)
    viErr = visa.viOpen(viDefRm, Sheet1.Cells(1, 2), 0, 5000, viDevice)

    'Send request, read the data, the return value is in CELLS(2,2) of SHEET1'
    cmdStr = "*IDN?"
    viErr = visa.viWrite(viDevice, cmdStr, Len(cmdStr), ret)
    viErr = visa.viRead(viDevice, idnStr, 128, ret)
    Sheet1.Cells(2, 2) = idnStr

    'Turn off the device'
    visa.viClose (viDevice)
    visa.viClose (viDefRm)

End Sub
```

6. Add button control: click **Insert** in the **Developer** menu, select the desired button in **Form Controls** and put it into the cell of the Excel. At this point, the **Assign Macro** interface is displayed, select "Sheet1.QueryIdn" and click "Run".



By default, the button mane is "Button 1". Right-click the button and select **Edit Text** in the pop-up menu to change the button name to "*IDN?".

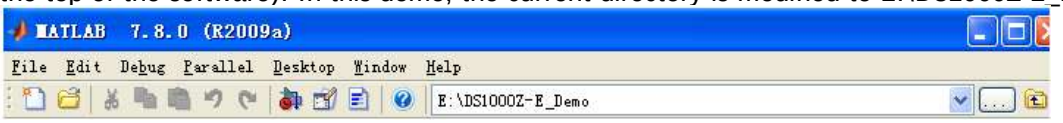
7. Click the "*IDN?" button to send request and read data. The returned data will be displayed in SHEET1 CELLS (2,2).

Matlab Programming Demo

The program used in this demo: MATLAB R2009a

The function realized in this demo: make FFT operation on the waveform data and draw the waveform.

1. Run the Matlab software and modify the current directory (namely modify the **Current Directory** at the top of the software). In this demo, the current directory is modified to E:\DS1000Z-E_Demo.



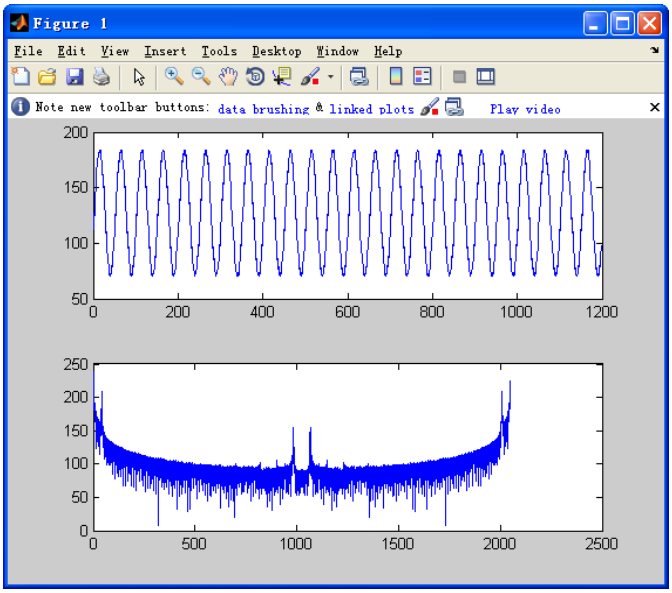
2. Click **File → New → Blank M-File** in the Matlab interface to create an empty M file.

3. Add the following codes in the M file:

```
% Create VISA object
DS1000Z-E = visa('ni','USB0::0x1AB1::0x04CE::DS1ZA160801111::INSTR');

% Set the device property. In this demo, the length of the input buffer is set to 2048.
DS1000Z-E.InputBufferSize = 2048;

% Open the VISA object created
fopen(DS1000Z-E);

% Read the waveform data
fprintf(DS1000Z-E, ':wav:data?' );

% Request the data
[data,len]= fread(DS1000Z-E,2048);

% Close the VISA object
fclose(DS1000Z-E);
delete(DS1000Z-E);
clear DS1000Z-E;

% Data processing. The waveform data read contains the TMC header. The length of the header is 11 bytes, wherein, the first 2 bytes are the TMC header denoter (#) and the width descriptor (9) respectively, the 9 bytes following are the length of the data which is followed by the waveform data and the last byte is the terminator (0x0A). Therefore, the effective waveform points read is from the 12nd to the next to last.
wave = data(12:len-1);
wave = wave';
subplot(211);
plot(wave);
fftSpec = fft(wave',2048);
fftRms = abs(fftSpec');
fftLg = 20*log(fftRms);
subplot(212);
plot(fftLg);
```

4. Save the M file under the current directory. In this demo, the M file is named as DS1000Z-E_Demo_MATLAB.m.

5. Run the M file and the running result is as follows.



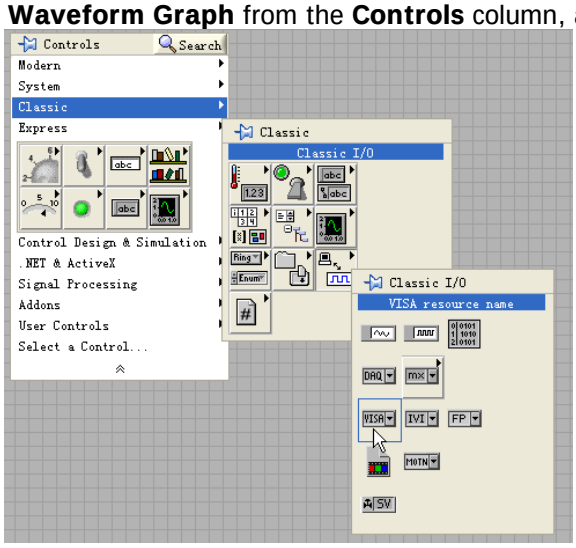
LabVIEW Programming Demo

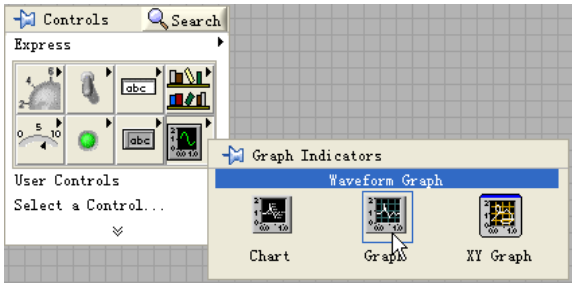
The program used in this demo: LabVIEW 2009

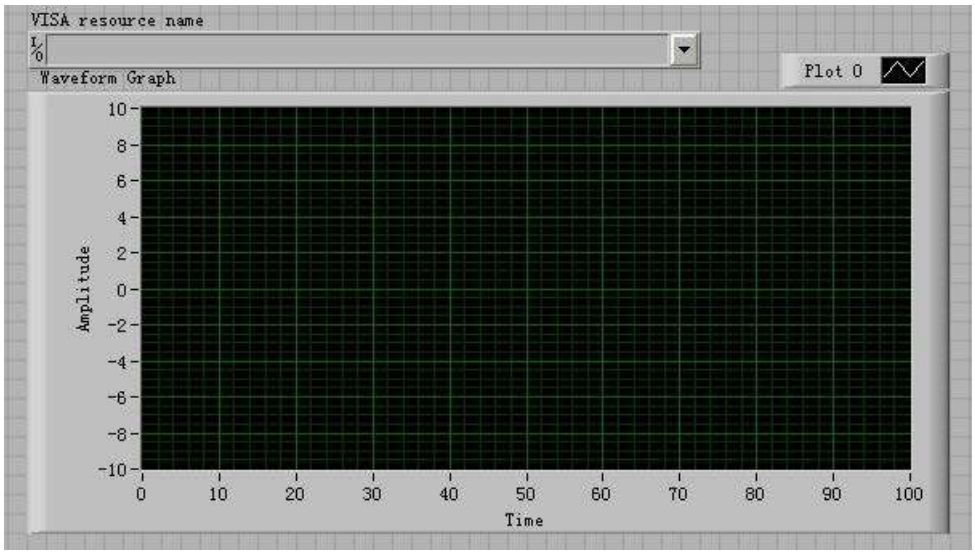
The functions realized in this demo: read the waveform data of CH1 on the screen.

1. Run LabVIEW 2009, create a VI file and name it as DS1000Z-E_Demo.

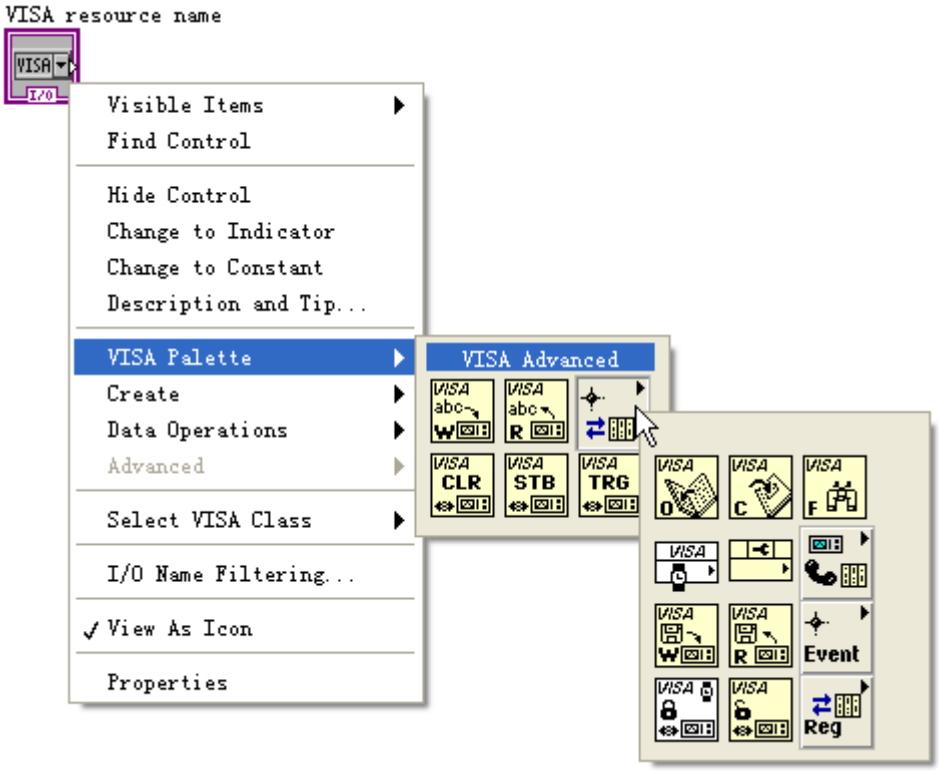
2. Add controls. Right-click in the **Front Panel** interface, select and add **VISA resource name** and **Waveform Graph** from the **Controls** column, as shown in the figures below.



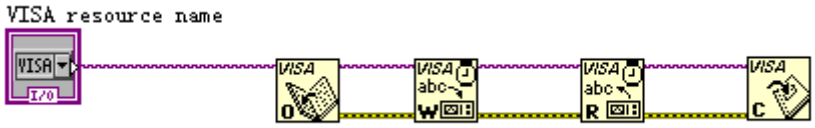




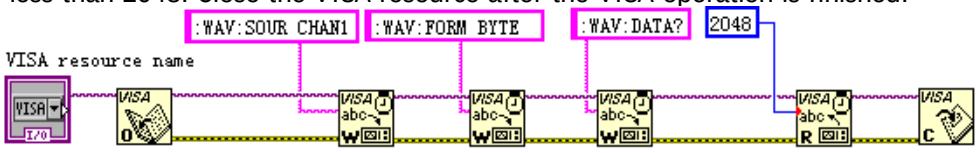
3. Open the **Block Diagram** interface. Right-click on the **VISA resource name** and you can select and add the following functions from **VISA Palette** from the pop-up menu: **VISA Write**, **VISA Read**, **VISA Open**, and **VISA Close**.



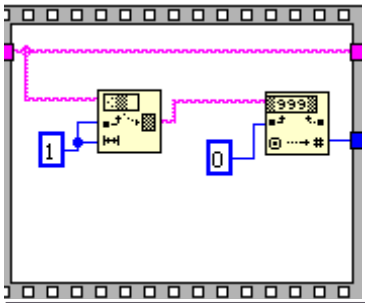
4. Connect the **VISA resource name** with the **VISA Open**. Then, connect the VISA resource name outputs of all the functions with the **VISA resource name** and connect the error output with the error input as shown in the figure below.

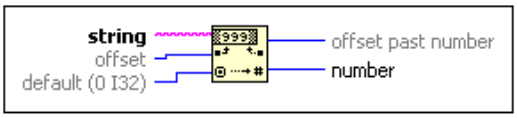


5. Add text boxes in the write buffer areas of the **VISA Write** control and input ":WAV:SOUR CHAN1", ":WAV:FORM BYTE", and ":WAV:DATA?" respectively. The first sets the channel source to CH1, the second sets the waveform reading format to BYTE and the last reads the screen waveform data. Waveform data is read through the **VISA Read** function which requires users to input the total number of bytes to be read. In this example, the total number of bytes of waveform data to be read is less than 2048. Close the VISA resource after the VISA operation is finished.

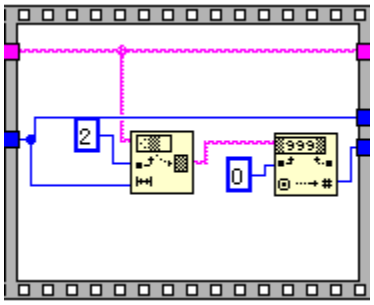


6. The data format is TMC header + waveform data points + terminator. The TMC header is in #NXXXXXX format, wherein, # is the TMC header denoter, N represents that there are N bytes following. The length of the waveform data points is described in ASCII character, and the terminator represents the ending of the communication. For example, for #9000001200XXXX, 9 bytes are used to describe the length of the data, 000001200 represents the length of the waveform data (namely 1200 bytes). Use the **String Subset** and **Decimal String To Number** functions to get the value of N.

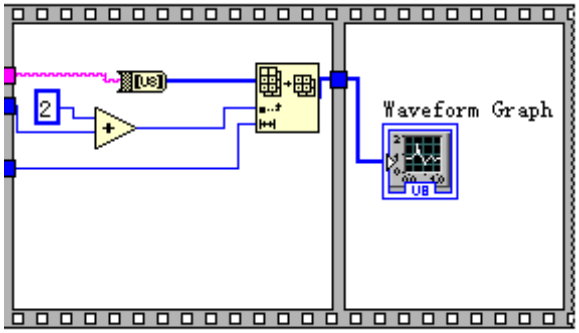


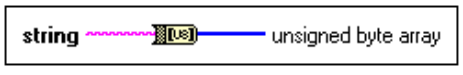


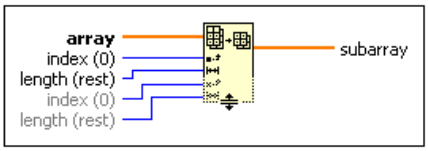
Get the length of the effective waveform data.



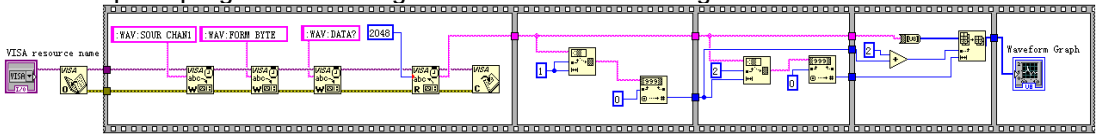
7. Convert the character data into array format using the **String To Byte Array**, namely display the waveform data on the **Waveform Graph** control and then remove the TMC data header using the **Array Subset** function.



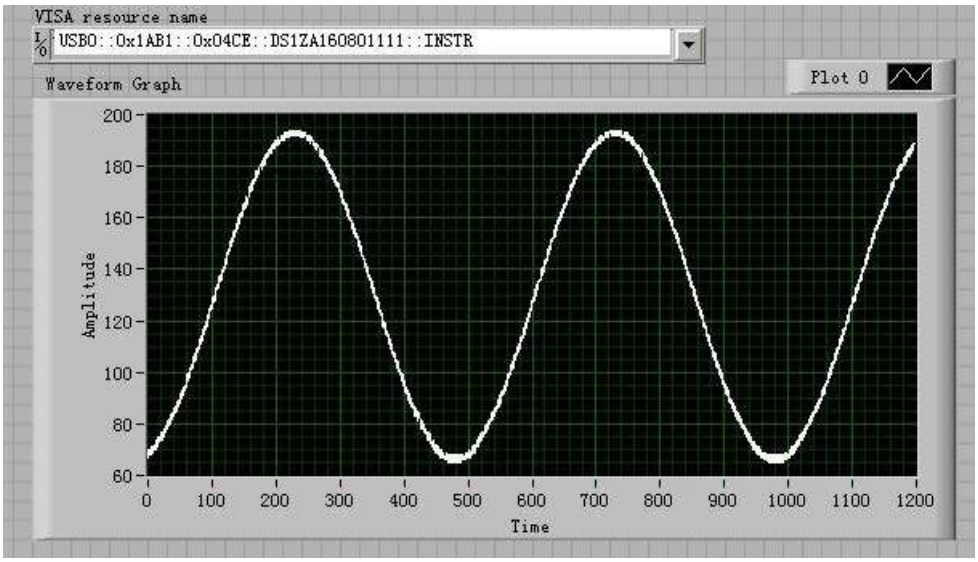




8. The complete program block diagram is as shown in the figure below.



9. Select the device resource from the **VISA Resource Name** list box and run the program.



Visual Basic Programming Demo

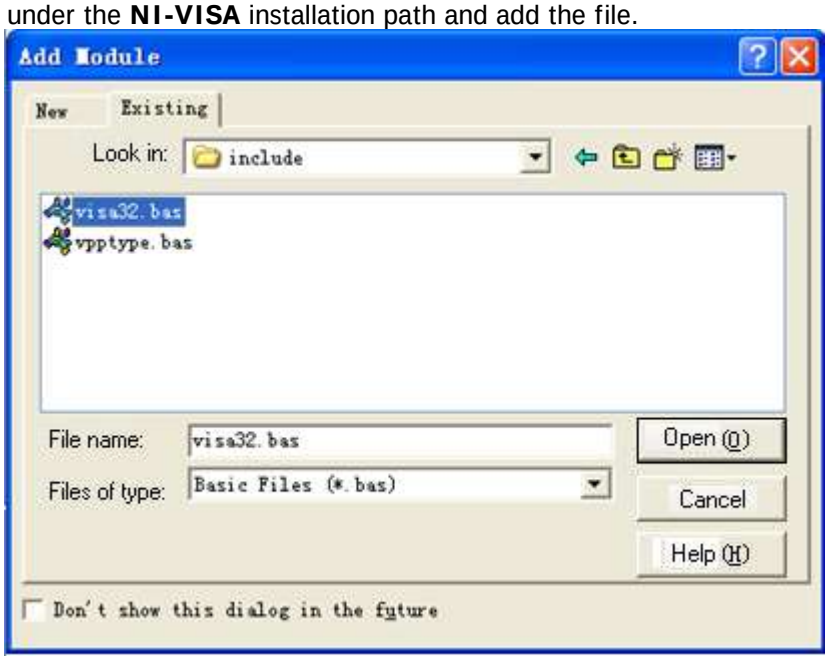
The program used in this demo: Visual Basic 6.0

The function realized in this demo: control the on/off state of any channel.

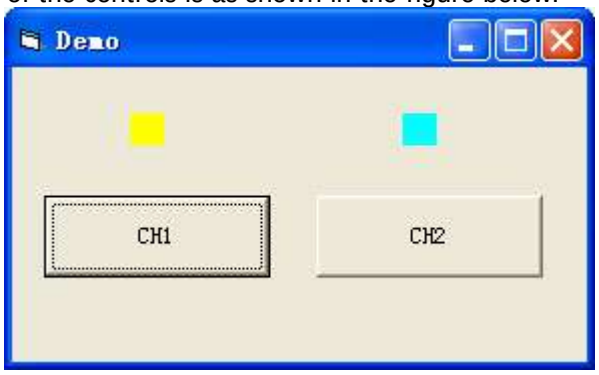
Enter Visual Basic 6.0 and operate according to the following steps:

1. Build a standard application program project (Standard EXE) and name it as Demo.

2. Click the **Existing** tab of **Project→Add Module**. Search for the **visa32.bas** file in the **include** folder under the **NI-VISA** installation path and add the file.



3. Add four **CommandButton** controls to represent CH1 and CH2 respectively. Add two Label controls (Label1(0) and Label1(1)) to represent the status of CH1 and CH2 respectively (when the channel is enabled, it displays the color of the channel; when the channel is disabled, it displays gray). The layout of the controls is as shown in the figure below.



4. Open the **General** tab in **Project→Project1 Properties** and select **Form1** in the **Startup Object** dropdown box.

5. Double-click **CH1** to enter the programming environment. Add the following codes to control CH1 and CH2. The codes of CH1 are as shown below; the codes of other channels are similar.

```
Dim defrm As Long
Dim vi As Long
```

```
Dim strRes As String * 200
Dim list As Long
Dim nmatches As Long
Dim matches As String * 200 ' keep the device number acquired
Dim s32Disp As Integer
' acquire the usb source of visa
Call viOpenDefaultRM(defrm)
Call viFindRsrc(defrm, "USB?*", list, nmatches, matches)
' Open the device
Call viOpen(defrm, matches, 0, 0, vi)
' Send the command to query the status of CH1
Call viVPrintf(vi, ":CHAN1:DISP?" + Chr$(10), 0)
' Acquire the status of CH1
Call viVScanf(vi, "%t", strRes)
s32Disp = CInt(strRes)
If (s32Disp = 1) Then
' Send the setting command
Call viVPrintf(vi, ":CHAN1:DISP 0" + Chr$(10), 0)
Label1(0).ForeColor = &H808080 'Gray
Else
Call viVPrintf(vi, ":CHAN1:DISP 1" + Chr$(10), 0)
Label1(0).ForeColor = &HFFFF& 'Yellow
End If
' Close the device
Call viClose(vi)
Call viClose(defrm)
```

6. Save and run the project and a single exe program will be obtained. When the oscilloscope is correctly connected to the PC, the ON/OFF control of any channel can be realized.

Visual C++ Programming Demo

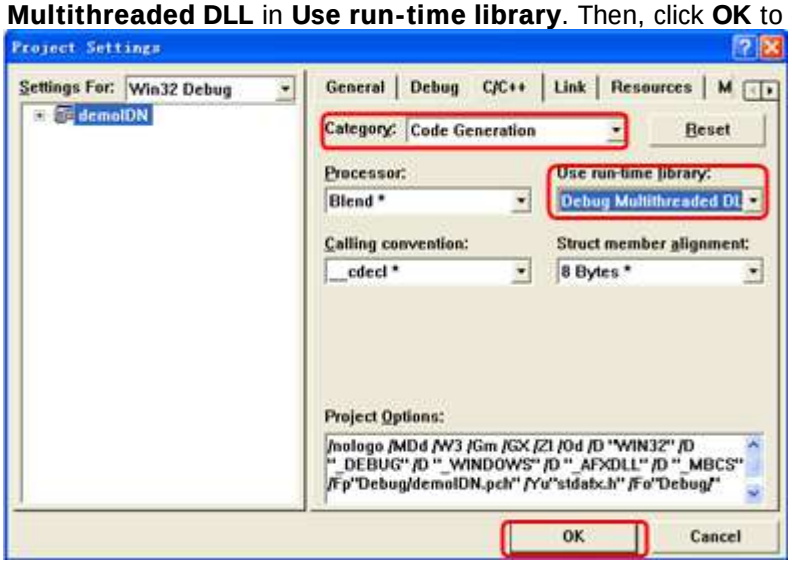
The program used in this demo: Microsoft Visual C++ 6.0

The functions realized in this demo: search for the instrument address, connect the instrument, send command and read the return value.

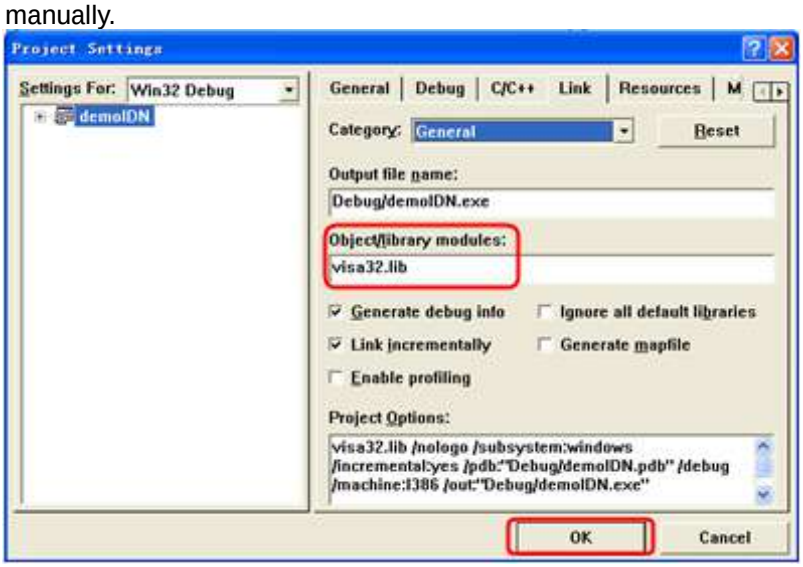
Enter Visual C++6.0 and operate according to the following steps:

1. Create a MFC project based on dialog box.

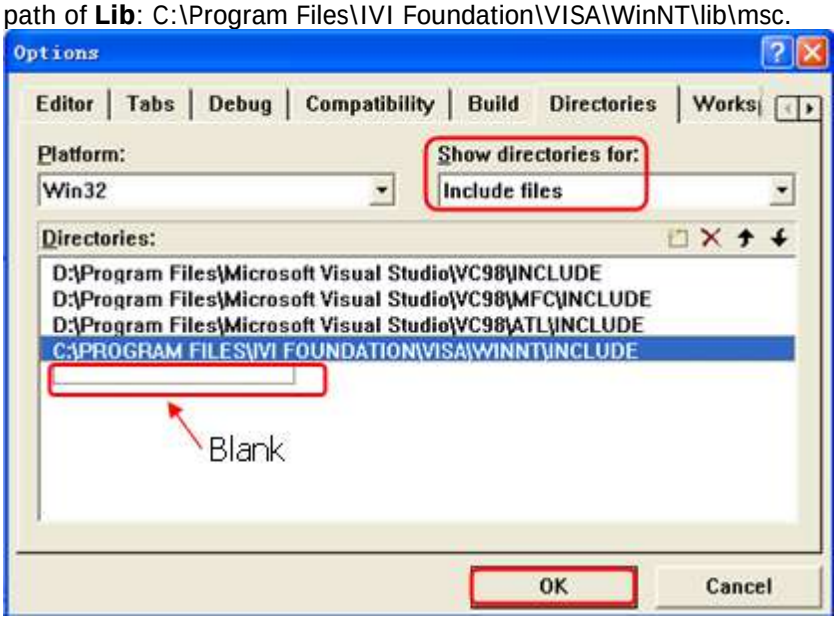
2. Open the **C/C++** tab in **Project→Settings**, select **Code Generation** in **Category** and **Debug Multithreaded DLL** in **Use run-time library**. Then, click **OK** to close the dialog box.



3. Open the **Link** tab in **Project→Settings** and add **visa32.lib** to the **Object/library modules** manually.



4. Open the **Directories** tab in **Tools→Options**.
   Select **Include files** in **Show directories for** and double-click the blank in **Directories** to add the path of **Include**: C:\Program Files\IVI Foundation\VISA\WinNT\include.
   Select **Library files** in **Show directories for** and double-click the blank in **Directories** to add the path of **Lib**: C:\Program Files\IVI Foundation\VISA\WinNT\lib\msc.

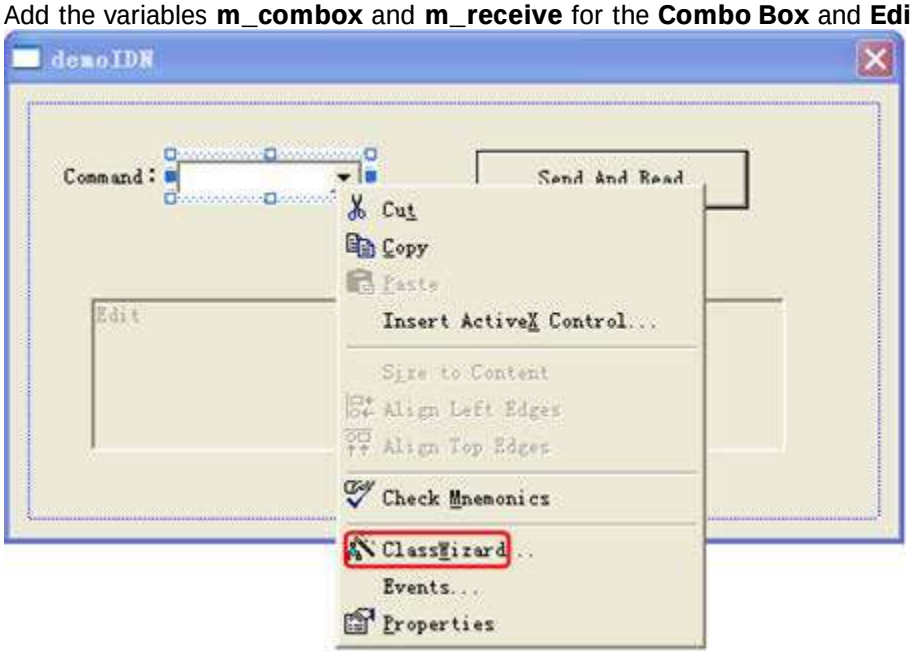


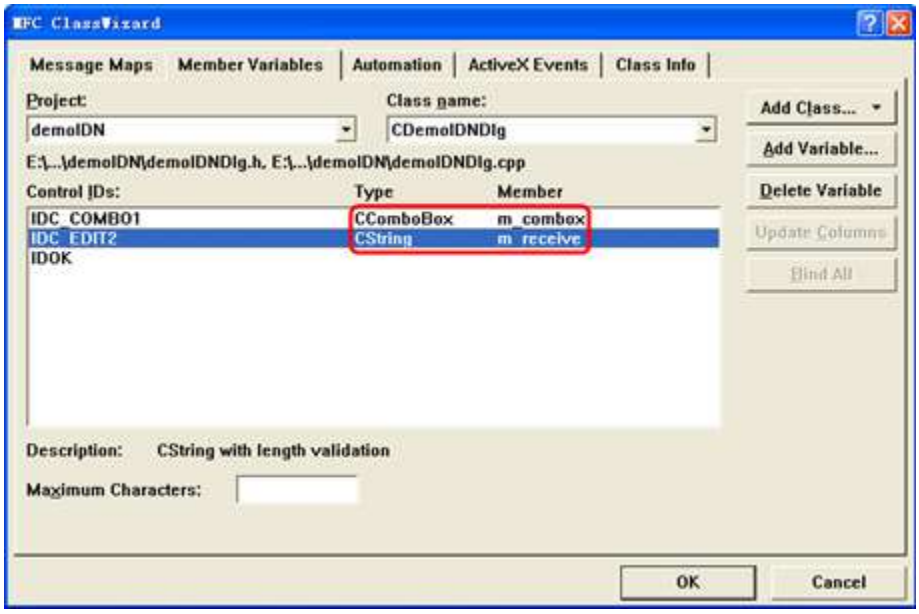
Note: at present, VISA library has been added.

5. Add the **Text**, **Combo Box**, **Button** and **Edit Box** controls as shown in the figure below.

6. Modify the control attribute.
   1) Name **Text** as "Command".
   2) Open the **Data** item in **Combo Box** attribute and input *IDN? Manually.
   3) Open the **General** item in **Edit Box** attribute and select **Disabled**.
   4) Name **Button** as **Send and Read**.

7. Add the variables **m_combox** and **m_receive** for the **Combo Box** and **Edit** controls respectively.





8. Add codes.
   Double-click **Send and Read** to enter the programming environment. Declare the **#include <visa.h>** of the visa library in the header file and then add the following codes.

```
ViSession defaultRM, vi;
char buf [256] = {0};
CString s,strTemp;
char* stringTemp;

ViChar buffer [VI_FIND_BUFLEN];
ViRsrc matches=buffer;
ViUInt32 nmatches;
ViFindList list;

viOpenDefaultRM (&defaultRM);
//Acquire the USB resource of visa
```

```
viFindRsrc(defaultRM, "USB?*", &list,&nmatches, matches);
viOpen (defaultRM,matches,VI_NULL,VI_NULL,&vi);

// Send the command received
m_combox.GetLBText(m_combox.GetCurSel(),strTemp);
strTemp = strTemp + "\n";
stringTemp = (char*)(LPCTSTR)strTemp;
viPrintf (vi,stringTemp);

// Read the result
viScanf (vi, "%t\n", &buf);

// Display the result
UpdateData (TRUE);
m_receive = buf;
UpdateData (FALSE);
viClose (vi);
viClose (defaultRM);
```

9. Save, compile and run the project and a single exe file can be obtained. When the oscilloscope is correctly connected to the PC, you can select the command *IDN? and click **Send and Read** to display the return result of the oscilloscope.