SCPI Remote communication control manual

Version number：V1.03

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.

Synctax

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.

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.

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.

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.

The list of commands:

- CHANnel<n>:BWLimit
- CHANnel<n>:COUPling
- CHANnel<n>:DISPlay
- CHANnel<n>:INVert
- CHANnel<n>:OFFSet
- CHANnel<n>:RANGe
- CHANnel<n>:SCALe
- CHANnel<n>:PROBe
- CHANnel<n>:VERNier

CHANnel<n>:BWLimit

Syntax	:CHANnel<n>:BWLimit <type> :CHANnel<n>:BWLimit?
Description	: Set or query the bandwidth limit parameter of the specified channel.
Parameter	:<type>::= {{1 | ON} | {0 | OFF} :<n>::= {1 | 2 | 3 | 4}
Explanation	:OFF: disable the bandwidth limit and the high frequency components of the signal under test can pass the channel. : ON: 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	: The query returns 0 or 1
Example	:CHANnel1:BWLimit 1 /* Enable the 20MHz bandwidth limit */ :CHANnel1:BWLimit? /* The query returns 1 */

CHANnel<n>:COUPling

Syntax :CHANnel<n>:COUPling <coupling>
:CHANnel<n>:COUPling?

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

Parameter :<coupling> ::= {AC | DC | GND}
:<n>::= {1 | 2 | 3 | 4}

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 : The query returns AC,DC,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 :<bool> ::= {{1 | ON} | {0 | OFF}}
:<n>::= {1 | 2 | 3 | 4}

Return :The query return 0 or 1

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 :<bool> ::= {{1 | ON} | {0 | OFF}}

:<n>::= {1 | 2 | 3 | 4}

Return : The query return 0 or 1

Example :CHANnel1:INVert ON /*Enable the waveform invert of CH1 */

:CHANnel1:INVert? /* The query returns 1*/

CHANnel<n>:OFFSet

Syntax :CHANnel<n>:OFFSet <offset> [<suffix>]

:CHANnel<n>:OFFSet?

Description :Set or query the vertical offset of the specified channel.

Parameter :<offset> ::= number

:<suffix> ::= {V | mV}

:<n>::= {1 | 2 | 3 | 4}

Explanation : The vertical displacement value is set by the vertical gear and the probe ratio. The range of legal values varies with the set vertical and probe ratios, and if you set a value that is outside the legal range, the offset value is automatically set to the nearest legal value

Return : The default unit is V

Example :CHANnel1:OFFSet 1V /*Set the vertical offset of CH1 to 1V*/

:CHANnel1:OFFSet? /* The query returns 1*/

CHANnel<n>:RANGe

Syntax :CHANnel<n>:RANGe <range> [<suffix>]

:CHANnel<n>:RANGe?

Description :Set or query the vertical range of the specified channel.

Parameter :<range> ::= number

:<suffix> ::= {V | mV}

:<n>::= {1 | 2 | 3 | 4}

Explanation :This command indirectly modifies the vertical scale of the specified channel

:When the probe ratio is 1X: 5mV-100V。

Return :The query returns the vertical range in scientific notation

Example :CHANnel1:RANGe 1V /*Set the vertical range of CH1 to 1V*/

:CHANnel1:RANGe? /* The query returns 1.000e+00*/

CHANnel<n>:SCALe

Syntax	:CHANnel<n>:SCALe <scale> [<suffix>]
	:CHANnel<n>:SCALe?
Description	:Set or query the vertical scale of the specified channel
Parameter	:<scale> ::= integer
	:<suffix> ::= {V | mV}
	:<n>::= {1 | 2 | 3 | 4}
Explanation	:The range of the vertical scale is related to the current probe ratio
	: The settable range of the vertical gear is related to the currently set probe ratio.
Return	:The query returns the vertical range in scientific notation
Example	:CHANnel1:SCALe 1V /*Set the vertical scale of CH1 to 1V */
	:CHANnel1:SCALe? /* The query returns 1.000e+00*/

CHANnel<n>:PROBe

Syntax	:CHANnel<n>:PROBe <atten>
	:CHANnel<n>:PROBe?
Description	:Set or query the probe ratio of the specified channel.
Parameter	:<atten> ::= {1 | 10 | 100 | 1000}
	:<n>::= {1 | 2 | 3 | 4}
Explanation	: Setting the probe ratio refers to multiply the signal sampled with the specified ratio and then display the result
	:Setting the probe ratio will affect the range of the vertical scale
Return	:The query returns the vertical range in scientific notation
Example	:CHANnel1:PROBe 10 /*Set the probe ratio of CH1 to 10X*/
	:CHANnel1:PROBe? /* The query returns 1.000e+01*/

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 , or query the fine adjustment status of the vertical scale of the specified channel

Parameter :<bool> ::= {{1 | ON} | {0 | OFF}}

:<n>::= {1 | 2 | 3 | 4}

Explanation : By default, the fine adjustment is off. At this point, you can only set the vertical scale in 1-2-5 step, 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 : The query returns 1 or 0

Example :CHANnel1:VERNier 1 /*Enable the fine adjustment function of the vertical scale of CH1*/

:CHANnel1:VERNier? /* The query returns 1*/

TIMebase commands

The list of commands:

- TIMebase:WINDow:ENABle
- TIMebase:WINDow:POSition
- TIMebase:WINDow:SCALe
- TIMebase:WINDow:RANGe
- TIMebase:POSition
- TIMebase:SCALe
- TIMebase:RANGe
- TIMebase:MODE
- TIMebase:VERNier
- TIMebase:XY:XSOUrce
- TIMebase:XY:XSOUrce

TIMebase:WINDow:ENABle

Syntax	:TIMebase:WINDow:ENABle <bool>
	:TIMebase:WINDow:ENABle?
Description	:Enable or disable the delayed sweep, or query the status of the delayed sweep
Parameter	:<bool> ::= {{1 | ON} | {0 | OFF}}
Explanation	:Delayed sweep can be used to enlarge a length of waveform horizontally to view waveform details.
Return	: The query returns ON or OFF
Example	:TIMebase:WINDow:ENABle 1 /*Enable the delayed sweep*/
	:TIMebase:WINDow:ENABle? /* The query returns ON*/

TIMebase:WINDow:POSition

Syntax	:TIMebase:WINDow:POSition <pos value>
	:TIMebase:WINDow:POSition?
Description	:Set or query the delayed timebase offset
Parameter	:<pos value> ::= The value of the horizontal position (The default unit is s)
Explanation	: 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 :The query returns the vertical range in scientific notation

Example :TIMebase:WINDow:POSition 0.001 /* Set the delayed timebase offset to 0.001s*/

:TIMebase:WINDow:POSition? /* The query returns1.000e-03*/

TIMebase:WINDow:SCALe

Syntax :TIMebase:WINDow:SCALe <scale_value>

:TIMebase:WINDow:SCALe?

Description :Set or query the delayed timebase scale (us/div)

Parameter :<sacle_value> ::= Real

Explanation : The main scan scale determines the range of this command, the maximum value of the main scan scale half

Return :The query returns the vertical range in scientific notation

Example :TIMebase:WINDow:SCALe 1 /* Set the child window time base 1us*/

:TIMebase:WINDow:SCALe? /* The query returns 1.000e+00*/

TIMebase:WINDow:RANGe

Syntax :TIMebase:WINDow:RANGe <range value>

:TIMebase:WINDow:RANGe?

Description : Set or query the delayed timebase range

Parameter :<range value> ::= Real (unit is s)

Explanation :The main scan range determines the range of this command, the maximum value of the main scan range of half

Return :The query returns the vertical range in scientific notation

Example :TIMebase:WINDow:RANGe 1 /* Set the subwindow horizontal range */

:TIMebase:WINDow:RANGe? /* The query returns 1.000e+00*/

TIMebase:POSition

Syntax :TIMebase:POSition <pos value>

:TIMebase:POSition?

Description : Set or query the main timebase offset

Parameter :<pos value> ::= Real (unit is s)

Return :The query returns the vertical range in scientific notation

Example :TIMebase:POSition 0.0002 /* Set the main time base offset to 200us*/

:TIMebase:POSition? /* The query returns 2.000e-4*/

TIMebase:SCALe

Syntax :TIMebase:SCALe <scale value>

:TIMebase:SCALe?

Description :Set or query the main timebase scale (s/div)

Parameter :<scale value> ::= The current number of seconds per grid of the main window

Return :The query returns the vertical range in scientific notation

Example :TIMebase:SCALe 0.0005 /* Set the main time base gear 500us*/

:TIMebase:SCALe? /*return 5.000e-4*/

TIMebase:RANGe

Syntax :TIMebase:RANGe <range value>

:TIMebase:RANGe?

Description : Set or query the main window full range time

Parameter :<range value> ::= time (unit is s)

Return : Returns the base range in the form of a scientific count to the main window full range

Example :TIMebase:RANGe 0.0016 /* Set the main time base gear 100us*/

:TIMebase:RANGe? /*return 1.600e-03*/

TIMebase:MODE

Syntax :TIMebase:MODE <value>

:TIMebase:MODE?

Description : Set or query horizontal base mode

Parameter :<value> ::= <MAIN | XY | ROLL>

Return :return MAIN,XY or ROLL

Example :TIMebase:MODE XY /*Set the level time base modeXY*/

:TIMebase:MODE? /*return XY*/

TIMebase:VERNier

Syntax	:TIMebase:VERNier <value> :TIMebase:VERNier?
Description	: Setting or querying time base fine-tuning switch state
Parameter	:<value> ::= {{1 | ON} | {0 | OFF}}
Explanation	: Fine tuning Settings by default. At this point, you can only press 1-2-5 step into the setting, and when the setting is opened, you can further adjust the base gear in a smaller range
Return	: Query return ON or OFF
Example	:TIMebase:VERNier ON /* Open the horizontal fine tune */ :TIMebase:VERNier? /*return ON*/

TIMebase:XY:XSOUrce

Syntax	:TIMebase:XY:XSOUrce <source> :TIMebase:XY:XSOUrce?
Description	: To set or query the YT mode, source x
Parameter	:<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 >
Return	: Query return CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example	:TIMebase:XY:XSOUrce CHANnel1 /* Set the source x for CH1*/ :TIMebase:XY:XSOUrce? /*return CHANnel1 */

TIMebase:XY:YSOUrce

Syntax	:TIMebase:XY:YSOUrce <source> :TIMebase:XY:YSOUrce?
Description	: Set or query the YT mode, source y
Parameter	:<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 >
Return	: Query return CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example	:TIMebase:XY:YSOUrce CHANnel1 /* Set the source y for CH1*/ :TIMebase:XY:YSOUrce? /*return CHANnel1 */

ACQ commands

The list of commands:

- ACQuire:POINts
- ACQuire:TYPE
- ACQuire:SRATe?
- ACQuire:COUNt

ACQuire:POINts

Syntax	:ACQuire:POINts <value>
	:ACQuire:POINts?
Description	:Set or query the memory depth of the oscilloscope
Parameter	:<value> ::= value of the memory depth
Explanation	:value memory depth
	4000 4K
	16000 16K
	32000 32K
	64000 64K
Return	:The query returns the actual number of points
Example	:ACQuire:POINts 4000 /*Set the memory depth to 4k*/
	:ACQuire:POINts? /*The query returns 4000*/

ACQuire:TYPE

Syntax :ACQuire:TYPE <value>
:ACQuire:TYPE?

Description :Set or query the acquisition mode of the oscilloscope

Parameter :<value> ::= < NORMal | AVERage | PEAK | HRESolution>

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.

: AVERage: 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. 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: 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: 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 :The query returns NORMal, AVERage, PEAK, HRESolution

Example :ACQuire:TYPE /*Select the average acquisition mode AVERage*/
:ACQuire:TYPE? /* The query returns AVERage */

ACQuire:SRATe?

Syntax :ACQuire:SRATe?

Description : Query the current sample rate

Explanation : Sample rate is the sample frequency of the oscilloscope, namely the waveform points sampled per second

Return :The query returns the actual number of points

Example :ACQuire:SRATe? /* The query returns 1.25e+06*/

ACQuire:COUNt

Syntax : ACQuire:COUNt <value>
: ACQuire:COUNt?
Description : Set or query Average number of points in average mode
Parameter :<value> ::= <4|8|16|32|64|128>
Return : The query returns current average number of points
Example : ACQuire:COUNt 64 /*Set average number of points 64*/
: ACQuire:COUNt? /* The query returns 64*/

TRIGger commands

The list of commands:

- TRIGger:FORCe
- TRIGger:MODE
- TRIGger:STATus?
- TRIGger:SWEep
- TRIGger:HOLDoff
- TRIGger:EDGe
- TRIGger:PULSe
- TRIGger:SLOPe
- TRIGger:TV
- TRIGger:TIMeout
- TRIGger:WINDOw
- TRIGger:INTERVAl
- TRIGger:UNDER_Am
- TRIGger:UART
- TRIGger:CAN
- TRIGger:LIN
- TRIGger:IIC
- TRIGger:SPI
- TRIGger:LOGIc

TRIGger:FORCe

Syntax :TRIGger:FORCe
Description : Set force trigger
Explanation : This command also allows the oscilloscope to acquire waveforms even if the trigger condition is not met

TRIGger:MODE

Syntax : TRIGger:MODE <mode>
: TRIGger:MODE?
Description : Select or query the trigger type.
Parameter :<mode> ::= < EDGE | PULSe | TV | SLOPe | TIMeout | WINdow | PATTern | INTerval | UNDerthrow | UART | LIN | CAN | SPI | IIC>
Return :return EDGE, PULSe, TV, SLOPe, TIMeout, WINdow, PATTern, INTerval, UNDerthrow, UART, LIN, CAN, SPI, IIC
Example : TRIGger:MODE SLOPe /*Select slope trigger*/
: TRIGger:MODE? /* The query returns SLOPe*/

TRIGger:STATus?

Syntax :TRIGger:STATus?
Description : Query the current trigger status
Return : The query returns TRIGed, NOTRIG

TRIGger:SWEep

Syntax :TRIGger:SWEep <value>
:TRIGger:SWEep?
Description : Set or query the trigger mode
Parameter :<value> ::= < AUTO | NORMal | SINGle>
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 : The query returns AUTO, NORMal, SINGle
Example :TRIGger:SWEep SINGle /*Select single trigger mode*/
:TRIGger:SWEep? /* The query returns SINGle*/

TRIGger:HOLDoff

Syntax	:TRIGger:HOLDoff <value> :TRIGger:HOLDoff?
Description	: Set or query the trigger holdoff time
Parameter	:<value> ::= holdoff time（unit is s）
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.
Return	: The query returns current average number of points
Example	:TRIGger:HOLDoff 0.0000002 /*Set the trigger holdoff time to 200ns*/ :TRIGger:HOLDoff? /* The query returns 2.000e-07*/

TRIGger:EDGe

The list of commands:

- 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	:<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | EXT>
Return	: The query returns CHANnel1,CHANnel2,CHANnel3,CHANnel4, EXT
Example	:TRIGger:EDGe:SOURce CHANnel1 /*Set the trigger source to CH1*/ :TRIGger:EDGe:SOURce? /* The query returns CHANnel1 */

TRIGger:EDGe:SLOPe

Syntax	:TRIGger:EDGe:SLOPe <slope> :TRIGger:EDGe:SLOPe?
Description	: Set or query the edge type in edge trigger
Parameter	:<slope> ::= < RISIng| FALLing | EITHer >
Explanation	:RISIng:rising edge :FALLing:falling edge :EITHer:ringsing/falling edge
Return	: The query returns RISIng, FALLing, EITHer
Example	:TRIGger:EDGe:SLOPe RISIng /*Set the edge type to falling edge*/ :TRIGger:EDGe:SLOPe? /* The query returns RISIng*/

TRIGger:EDGe:LEVel

Syntax	:TRIGger:EDGe:LEVel <level> :TRIGger:EDGe:LEVel?
Description	: Set or query the trigger level in edge trigger
Parameter	:<level> ::= trigger level (unit is V)
Return	: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 1.600e-1*/

TRIGger:PULSe

The list of commands:

- TRIGger:PULSe:SOURce
- TRIGger:PULSe:POLarity
- TRIGger:PULSe:WHEN
- TRIGger:PULSe:WIDth
- TRIGger:PULSe:LEVel

TRIGger:PULSe:SOURce

Syntax	:TRIGger:PULSe:SOURce <source> :TRIGger:PULSe:SOURce?
Description	:Set or query the trigger source in pulse width trigger
Parameter	:<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>
Return	: The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example	:TRIGger:PULSe:SOURce CHANnel1 /* Set the trigger source to CH1*/ :TRIGger:PULSe:SOURce? /* The query returns CHANnel1*/

TRIGger:PULSe:POLarity

Syntax	:TRIGger:PULSe:POLarity <polarity> :TRIGger:PULSe:POLarity?
Description	:Set or query the trigger polarity of the pulse width trigger
Parameter	:<polarity> ::= < POSItive | NEGAtive>
Explanation	:POSItive: Positive pulse trigger :NEGAtive: Negative pulse trigger

Return : The query returns POSItive, NEGAtive

Example :TRIGger:PULSe:POLarity POSItive /* Set positive pulse trigger */

:TRIGger:PULSe:POLarity? /* The query returns POSItive*/

TRIGger:PULSe:WHEN

Syntax :TRIGger:PULSe:WHEN <when>

:TRIGger:PULSe:WHEN?

Description : Set or query the trigger condition in pulse width trigger

Parameter :<when> ::= < EQUAl| NEQUal| GREAt| LESS>

Explanation :EQUAl: 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 equal than the specified Pulse Width.

:NEQUal: 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 not equal to the specified Pulse Width.

:GREAt: 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 than the specified Pulse Width.

:LESS: 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 lesser than the specified Pulse Width.

Return : The query returns EQUAl, NEQUal, GREAt, LESS

Example :TRIGger:PULSe:WHEN NEQUal /*Set the trigger condition to NEQUal*/

:TRIGger:PULSe:WHEN? /* The query returns NEQUal*/

TRIGger:PULSe:WIDth

Syntax :TRIGger:PULSe:WIDth <value>

:TRIGger:PULSe:WIDth?

Description : Set or query the pulse width in pulse width trigger

Parameter :<value> ::= pulse width in pulse width trigger (unit is s)
Return : The query returns the pulse width in scientific notation.
Example :TRIGger:PULSe:WIDth 0.000003 /*Set the pulse width 3us*/
:TRIGger:PULSe:WIDth? /* The query returns 3.000000e-06*/

TRIGger:PULSe:LEVel

Syntax :TRIGger:PULSe:LEVel <level>
:TRIGger:PULSe:LEVel?
Description : Set or query the trigger level in pulse width trigger
Parameter :<level> ::= trigger level (unit is V)
Return :The query returns the pulse width in scientific notation.
Example :TRIGger:PULSe:LEVel 0.16 /*Set the trigger level*/
:TRIGger:PULSe:LEVel? /* The query returns 1.600000e-01*/

TRIGger:SLOPe

The list of commands:

- TRIGger:SLOPe:SOURce
- TRIGger:SLOPe:POLarity
- TRIGger:SLOPe:WHEN
- TRIGger:SLOPe:WIDth
- TRIGger:SLOPe:ALEVel
- TRIGger:SLOPe:BLEVel

TRIGger:SLOPe:SOURce

Syntax :TRIGger:SLOPe:SOURce <source>
:TRIGger:SLOPe:SOURce?
Description :Set or query the trigger level in pulse width trigger
Parameter :<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>
Return : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example :TRIGger:SLOPe:SOURce CHANnel1 /*Set trigger source CH1*/
:TRIGger:SLOPe:SOURce? /* The query returns CHANnel1*/

TRIGger:SLOPe:POLarity

Syntax	:TRIGger:SLOPe:POLarity <polarity> :TRIGger:SLOPe:POLarity?
Description	: Sets or queries the trigger polarity of the slope trigger
Parameter	:<polarity> ::= < POSItive | NEGAtive>
Explanation	:POSItive: Positive slope trigger :NEGAtive: Negative slope trigger
Return	:The query returns POSItive, NEGAtive
Example	:TRIGger:SLOPe:POLarity POSItive /* Set positive slope trigger */ :TRIGger:SLOPe:POLarity? /* The query returns POSItive*/

TRIGger:SLOPe:WHEN

Syntax	:TRIGger:SLOPe:WHEN <when> :TRIGger:SLOPe:WHEN?
Description	: Set or query the trigger condition in slope trigger
Parameter	:<when> ::= < EQUAl| NEQUal| GREAt| LESS>
Explanation	:EQUAl: you need to specify a time value(refer to TRIGger:SLOPe:WIDth), The oscilloscope triggers when the positive slope time of the input signal is equal to the specified time :NEQUal: you need to specify a time value(refer to TRIGger:SLOPe:WIDth), The oscilloscope triggers when the positive slope time of the input signal is not equal to the specified time :GREAt: you need to specify a time value(refer to TRIGger:SLOPe:WIDth), The oscilloscope triggers when the positive slope time of the input signal is greater than the specified time :LESS: you need to specify a time value(refer to TRIGger:SLOPe:WIDth), The oscilloscope triggers when the positive slope time of the input signal is lesser than the specified time
Return	: The query returns EQUAl, NEQUal, GREAt, LESS
Example	:TRIGger:SLOPe:WHEN NEQUal /*Set the trigger condition to NEQUal*/

:TRIGger:SLOPe:WHEN? /* The query returns NEQUal*/

TRIGger:SLOPe:WIDth

Syntax :TRIGger:SLOPe:WIDth <value>
:TRIGger:SLOPe:WIDth?
Description : Set or query the time value in slope trigger
Parameter :<value> ::= time value (unit is s)
Return : The query returns the time value in scientific notation.
Example :TRIGger:SLOPe:WIDth 0.000003 /*Set the time value to3us*/
:TRIGger:SLOPe:WIDth? /* The query returns 3.000000e-06*/

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
Parameter :<level> ::= upper limit of the trigger level (unit is V)
Return : 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 */
:TRIGger:SLOPe:ALEVel? /* The query returns 1.600000e-01*/

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
Parameter :<level> ::=the lower limit of the trigger level(unit is V)
Return : The query returns the upper limit of the trigger level in scientific notation
Example :TRIGger:SLOPe:BLEVel 0.16 /*Set the lower limit of the trigger level */
:TRIGger:SLOPe:BLEVel? /* The query returns 1.600000e-01*/

TRIGger:TV

The list of commands:

- TRIGger:TV:SOURce
- TRIGger:TV:POLarity
- TRIGger:TV:MODE
- TRIGger:TV:LINE
- TRIGger:TV:STANdard
- TRIGger:VIDeo:LEVel

TRIGger:TV:SOURce

Syntax	:TRIGger:TV:SOURce <source>
	:TRIGger:TV:SOURce?
Description	: Select or query the trigger source in video trigger
Parameter	:<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>
Return	: The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example	:TRIGger:TV:SOURce CHANnel1 /*Set trigger source CH1*/
	:TRIGger:TV:SOURce? /* The query returns CHANnel1*/

TRIGger:TV:POLarity

Syntax	:TRIGger:TV:POLarity <polarity>
	:TRIGger:TV:POLarity?
Description	:Select or query the video polarity in video trigger
Parameter	:<polarity> ::= < POSItive | NEGAtive>
Explanation	:POSItive: Positive polarity trigger
	:NEGAtive: Negative polarity trigger
Return	: The query returns POSItive, NEGAtive
Example	:TRIGger:TV:POLarity POSItive /*Set POSItive */
	:TRIGger:TV:POLarity? /* The query returns POSItive*/

TRIGger:TV:MODE

Syntax :TRIGger:TV:MODE <mode>
:TRIGger:TV:MODE?
Description : Set or query the sync type in video trigger
Parameter :<mode> ::= <
SCAN_LINE|LINE_NUM|ODD_FIELD|EVEN_FIELD|ALL_FIELD>
Return :The query returns SCAN_LINE, LINE_NUM, ODD_FIELD,
EVEN_FIELD, ALL_FIELD
Example :TRIGger:TV:MODE ALL_FIELD /* Set the sync type to
ALL_FIELD */
:TRIGger:TV:MODE? /* The query returns FIEld1*/

TRIGger:TV:LINE

Syntax :TRIGger:TV:LINE <line>
:TRIGger:TV:LINE?
Description : Set or query the line number when the sync type in video
trigger is LINE
Parameter :<line> ::= line number
Explanation :NTSC:1-525
:PAL/SECAM:1-625
Return : The query returns an integer
Example :TRIGger:TV:LINE 100 /*Set line number 100*/
:TRIGger:TV:LINE? /*The query returns 100*/

TRIGger:TV:STANdard

Syntax :TRIGger:TV:STANdard <standard>
:TRIGger:TV:STANdard?
Description :Set or query the video standard in video trigger
Parameter :<standard> ::= < NTSC | PAL >
Return :The query returns NTSC,PAL
Example :TRIGger:TV:STANdard NTSC /*Set standard to NTSC*/
:TRIGger:TV:STANdard? /*The query returns NTSC*/

TRIGger:VIDeo:LEVel

Syntax	:TRIGger:VIDeo:LEVel <level> :TRIGger:VIDeo:LEVel?
Description	: Set or query the trigger level in video trigger
Parameter	:<level> ::= trigger level (unit is V)
Return	: The query returns the upper limit of the trigger level in scientific notation
Example	:TRIGger:VIDeo:LEVel 0.16 /*Set trigger level160mV*/ :TRIGger:VIDeo:LEVel? /* The query returns 1.600000e-01*/

TRIGger:TIMeout

The list of commands:

- TRIGger:TIMeout:SOURce
- TRIGger:TIMeout:LEVel
- TRIGger:TIMeout:WIDth
- TRIGger:TIMeout:POLarity

TRIGger:TIMeout:SOURce

Syntax	:TRIGger:TIMeout:SOURce <source> :TRIGger:TIMeout:SOURce?
Description	: Set or query the trigger source in timeout trigger
Parameter	:<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>
Return	: The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example	:TRIGger:TIMeout:SOURce CHANnel1 /*Set trigger source CH1*/ :TRIGger:TIMeout:SOURce? /* The query returns CHANnel1*/

TRIGger:TIMeout:LEVel

Syntax	:TRIGger:TIMeout:LEVel <level> :TRIGger:TIMeout:LEVel?
Description	: Set or query the trigger level for the timeout trigger
Parameter	:<level> ::= trigger level (unit is V)
Return	: The query returns the upper limit of the trigger level in scientific notation
Example	:TRIGger:TIMeout:LEVel 0.16 /*Set trigger level 160mV*/ :TRIGger:TIMeout:LEVel?/* The query returns 1.600000e-01*/

TRIGger:TIMeout:WIDth

Syntax	:TRIGger:TIMeout:WIDth <value> :TRIGger:TIMeout:WIDth?
Description	: Set or query the timeout time in timeout trigger
Parameter	:<value> ::= timeout time (unit is s, 8ns-10s)
Return	: The query returns the upper limit of the trigger level in scientific notation
Example	:TRIGger:TIMeout:WIDth 0.000003 /*Set timeout time to 3us*/

:TRIGger:TIMeout:WIDth? /* The query returns 3.000000e-06*/

TRIGger:TIMeout:POLarity

Syntax :TRIGger:TIMeout:POLarity <polarity>
:TRIGger:TIMeout:POLarity?

Description : Set or query the trigger polarity of the timeout trigger

Parameter :<polarity> ::= < POSItive | NEGAtive>

Explanation :POSItive: Positive polarity trigger
:NEGAtive: Negative polarity trigger

Return : The query returns POSItive, NEGAtive

Example :TRIGger:TIMeout:POLarity POSItive /*Set POSItive */
:TRIGger:TIMeout:POLarity? /* The query returns POSItive*/

TRIGger:WINDOw

The list of commands:

- TRIGger:WINDOw:SOURce
- TRIGger:WINDOw:ALEVel
- TRIGger:WINDOw:BLEVel

TRIGger:WINDOw:SOURce

Syntax :TRIGger:WINDOw:SOURce <source>
:TRIGger:WINDOw:SOURce?
Description : Set or query the trigger source in windows trigger
Parameter :<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>
Return : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example :TRIGger:WINDOw:SOURce CHANnel1 /*set trigger source CH1*/
:TRIGger:WINDOw:SOURce? /* The query returns CHANnel1*/

TRIGger:WINDOw:ALEVel

Syntax :TRIGger:WINDOw:ALEVel <level>
:TRIGger:WINDOw:ALEVel?
Description :Set or query the trigger level upper limit in windows trigger
Parameter :<level> ::= the trigger level upper limit (unit is V)
Return : The query returns the trigger level upper limit in scientific notation.
Example :TRIGger:WINDOw:ALEVel 0.16 /* Set the trigger level upper limit 160mV*/
:TRIGger:WINDOw:ALEVel?/* The query returns 1.600000e-01*/

TRIGger:WINDOw:BLEVel

Syntax :TRIGger:WINDOw:BLEVel <level>
:TRIGger:WINDOw:BLEVel?
Description : Set or query the tigger level lower limit in windows trigger

Parameter :<level> ::= the tigger level lower limit (unit is V)
Return : The query returns the trigger level upper limit in scientific notation.
Example :TRIGger:WINDOw:BLEVel 0.16 /* Set the trigger level lower limit 160mV*/
:TRIGger:WINDOw:BLEVel?/* The query returns 1.600000e-01*/

TRIGger:INTERVAl

The list of commands:

- TRIGger:INTERVAl:SOURce
- TRIGger:INTERVAl:SLOp
- TRIGger:INTERVAl:WHEN
- TRIGger:INTERVAl:TIME
- TRIGger:INTERVAl:ALEVel

TRIGger:INTERVAl:SOURce

Syntax :TRIGger:INTERVAl:SOURce <source>
:TRIGger:INTERVAl:SOURce?
Description :Set or query the trigger source for the interval trigger
Parameter :<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>
Explanation : Triggered when the interval between two rising edges (or falling edges) satisfies the set time condition
Return : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example :TRIGger:INTERVAl:SOURce CHANnel1 /* set trigger source CH1*/
:TRIGger:INTERVAl:SOURce? /* The query returns CHANnel1*/

TRIGger:INTERVAl:SLOp

Syntax : TRIGger:INTERVAl:SLOp <slope>
: TRIGger:INTERVAl:SLOp?
Description : Set or query the edge type for interval trigger
Parameter :<slope> ::= < RISIng| FALLing>

Explanation : RISIng
: FALLing
: DOUBle
Return : The query returns RISIng, FALLing, DOUBle
Example :TRIGger:INTERVAl:SLOp RISIng /* Set the edge type to the rising edge */
:TRIGger:INTERVAl:SLOp? /* The query returns POSItive*/

TRIGger:INTERVAl:WHEN

Syntax :TRIGger:INTERVAl:WHEN <when>
:TRIGger:INTERVAl:WHEN?
Description :Set or query the trigger conditions for the interval trigger
Parameter :<when> ::= < EQUAl| NEQUal| GREAt| LESS>
Explanation : Triggered when the interval between two rising edges (or falling edges) satisfies the set time condition
Return : The query returns EQUAl, NEQUal, GREAt, LESS
Example :TRIGger:INTERVAl:WHEN NEQUal /*Set the trigger condition toNEQUal*/
:TRIGger:INTERVAl:WHEN? /*The query returns NEQUal*/

TRIGger:INTERVAl:TIME

Syntax :TRIGger:INTERVAl:TIME <value>
:TRIGger:INTERVAl:TIME?
Description : Set or query the time value of the interval trigger
Parameter :<value> ::= time value (unit is s, 8ns-10s)
Return : The query returns the trigger level upper limit in scientific notation.
Example :TRIGger:INTERVAl:TIME 0.000003 /*set time value 3us*/
:TRIGger:INTERVAl:TIME? /* The query returns 3.000000e-06*/

TRIGger:INTERVAl:ALEVel

Syntax :TRIGger:INTERVAl:ALEVel <level>
:TRIGger:INTERVAl:ALEVel?

Description : Set or query the trigger level for the interval trigger

Parameter :<level> ::= trigger level (unit is V)

Return : The query returns the trigger level upper limit in scientific notation.

Example :TRIGger:INTERVAl:ALEVel 0.16 /*set trigger level 160mV*/
:TRIGger:INTERVAl:ALEVel? /* The query returns 1.600000e-01*/

TRIGger:UNDER_Am

: A runout trigger is used to trigger a pulse that crosses a trigger level but does not cross another trigger level

The list of commands:

- TRIGger:UNDER_Am:SOURce
- TRIGger:UNDER_Am:POLarity
- TRIGger:UNDER_Am:WHEN
- TRIGger:UNDER_Am:TIME
- TRIGger:UNDER_Am:ALEVel
- TRIGger:UNDER_Am:BLEVel

TRIGger:UNDER_Am:SOURce

Syntax :TRIGger:UNDER_Am:SOURce <source>
:TRIGger:UNDER_Am:SOURce?

Description : Set or query the trigger source for the under_am
Parameter :<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>
Return : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example :TRIGger:UNDER_Am:SOURce CHANnel1 /*set trigger source CH1*/
:TRIGger:UNDER_Am:SOURce? /* The query returns CHANnel1*/

TRIGger:UNDER_Am:POLarity

Syntax :TRIGger:UNDER_Am:POLarity <polarity>
:TRIGger:UNDER_Am:POLarity?
Description : Set or query the trigger polarity of the trigger
Parameter :<polarity> ::= < POSItive | NEGAtive>
Explanation :POSItive: Positive polarity trigger
:NEGAtive: Negative polarity trigger
Return : The query returns POSItive, NEGAtive
Example :TRIGger:UNDER_Am:POLarity POSItive /*set POSItive */
:TRIGger:UNDER_Am:POLarity?/* The query returns POSItive*/

TRIGger:UNDER_Am:WHEN

Syntax :TRIGger:UNDER_Am:WHEN <when>
:TRIGger:UNDER_Am:WHEN?
Description : Set or query the trigger condition for under_am
Parameter :<when> ::= < EQUAl| NEQUal| GREAt| LESS>
Return : The query returns EQUAl, NEQUal, GREAt, LESS
Example :TRIGger:UNDER_Am:WHEN NEQUal /*set condition NEQUal*/
:TRIGger:UNDER_Am:WHEN? /* The query returns NEQUal*/

TRIGger:UNDER_Am:TIME

Syntax :TRIGger:UNDER_Am:TIME <value>
:TRIGger:UNDER_Am:TIME?
Description : Set or query the time value of the under_am
Parameter :<value> ::= time value (unit is s, 8ns-10s)
Return : The query returns the trigger level upper limit in scientific notation.

Example :TRIGger:UNDER_Am:TIME 0.000003 /*set time value 3us*/
:TRIGger:UNDER_Am:TIME? /* The query returns 3.000000e-06*/

TRIGger:UNDER_Am:ALEVel

Syntax :TRIGger:UNDER_Am:ALEVel <level>
:TRIGger:UNDER_Am:ALEVel?
Description :Set or query the trigger level upper limit in under_am trigger
Parameter :<level> ::= trigger level upper limit (unit is V)
Return : The query returns the trigger level upper limit in scientific notation.
Example :TRIGger:UNDER_Am:ALEVel 0.16 /* set the trigger level limit 160mV*/
:TRIGger:UNDER_Am:ALEVel? /*The query returns 1.600000e-01*/

TRIGger:UNDER_Am:BLEVel

Syntax :TRIGger:UNDER_Am:BLEVel <level>
:TRIGger:UNDER_Am:BLEVel?
Description : Set or query the trigger level lower limit in under_am trigger
Parameter :<level> ::= the trigger level lower limit (unit is V)
Return : The query returns the trigger level upper limit in scientific notation.
Example :TRIGger:UNDER_Am:BLEVel 0.16 /*set the trigger level lower limit 160mV*/
:TRIGger:UNDER_Am:BLEVel? /* The query returns 1.600000e-01*/

TRIGger:UART

The list of commands:

- TRIGger:UART:SOURce
- TRIGger:UART:CONdition

- TRIGger:UART:BAUd
- TRIGger:UART:ALEVel
- TRIGger:UART:DATA
- TRIGger:UART:WIDTh
- TRIGger:UART:PARIty
- TRIGger:UART:WHEN
- TRIGger:UART:IDLe

TRIGger:UART:SOURce

Syntax	:TRIGger:UART:SOURce <source> :TRIGger:UART:SOURce?
Description	: Set or query the trigger source for the UART trigger
Parameter	:<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>
Return	: The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example	:TRIGger:UART:SOURce CHANnel1 /*set trigger source CH1*/ :TRIGger:UART:SOURce? /* The query returns CHANnel1*/

TRIGger:UART:CONdition

Syntax	:TRIGger:UART:CONdition <condition> :TRIGger:UART:CONdition?
Description	: Set or query the trigger conditions for the UART
Parameter	:<condition> ::= < START| STOP| READ_DATA| PARITY_ERR| COM_ERR>
Return	: The query returns START| STOP| READ_DATA| PARITY_ERR|COM_ERR
Example	:TRIGger:UART:CONdition START /*set trigger condition to START */ :TRIGger:UART:CONdition? /* The query returns START*/

TRIGger:UART:BAUd

Syntax	:TRIGger:UART:BAUd <baud>

:TRIGger:UART:BAUd?

Description : Set or query the baud rate triggered by the UART, unit is bps

Parameter :<baud> ::=<110|300|600|1200|2400|4800|9600|14400|19200|38400|57600|115200|230400|380400|460400|921600| customize >

Return : The query returns baud rate

Example :TRIGger:UART:BAUd 4800 /*set baud rate 4800*/
:TRIGger:UART:BAUd? /* The query returns 4800*/

TRIGger:UART:ALEVel

Syntax :TRIGger:UART:ALEVel <level>
:TRIGger:UART:ALEVel?

Description : Set or query the trigger level triggered by the UART

Parameter :<level> ::= trigger level (unit is V)

Return : The query returns the trigger level upper limit in scientific notation.

Example :TRIGger:UART:ALEVel 0.16 /*set trigger level 160mV*/
:TRIGger:UART:ALEVel? /* The query returns 1.600000e-01*/

TRIGger:UART:DATA

Syntax :TRIGger:UART:DATA <data>
:TRIGger:UART:DATA?

Description : Set or query the data value when the UART trigger condition is data

Parameter :<data> ::= （0-- (2^{n-1}-1）)

Explanation : In the above expression, n is the data width, in the range of 5,6,7,8

Return : The query returns data value

Example :TRIGger:UART:DATA 10 /*set data to 10*/
:TRIGger:UART:DATA? /* The query returns 10*/

TRIGger:UART:WIDTh

Syntax :TRIGger:UART:WIDTh <value>
:TRIGger:UART:WIDTh?

Description : Set or query the data bit width when the UART trigger condition is data

Parameter :<value> ::= <5,6,7,8>

Return : The query returns 5,6,7,8

Example :TRIGger:UART:WIDTh 5 /*set data width to 5*/

:TRIGger:UART:WIDTh? /* The query returns 5*/

TRIGger:UART:STOP

Syntax :TRIGger:UART:STOP <stop>

:TRIGger:UART:STOP?

Description :Set or query the stop bit when the trigger condition is ERRor in UART trigger

Parameter :<stop> ::= <1|2>

Return : The query returns 1 or 2

Example :TRIGger:UART:STOP 2 /*set stop bit to 2*/

:TRIGger:UART:STOP? /* The query returns 2*/

TRIGger:UART:PARIty

Syntax :TRIGger:UART:PARIty <parity>

:TRIGger:UART:PARIty?

Description : Set or query the parity type when the trigger condition is ERRor or PARity in UART trigger.

Parameter :<parity> ::= < NONE| ODD| EVEN>

Return : The query returns NONE| ODD| EVEN

Example :TRIGger:UART:PARIty EVEN /*set parity is EVEN*/

:TRIGger:UART:PARIty? /* The query returns EVEN */

TRIGger:UART:WHEN

Syntax :TRIGger:UART:WHEN <when>

:TRIGger:UART:WHEN?

Description : Set or query the trigger condition in UART trigger

Parameter :<when> ::= < EQUAl|NEQUal|GREAt|LESS>

Return : The query returns EQUAl, NEQUal, GREAt, LESS

Example :TRIGger:UART:WHEN LESS /*set trigger condition LESS */

:TRIGger:UART:WHEN? /* The query returns LESS */

TRIGger:UART:IDLe

Syntax :TRIGger:UART:IDLe <when>

:TRIGger:UART:IDLe?

Description : Set or query the idle level in UART trigger

Parameter :<when> ::= < HIGH| LOW>

Return : The query returns HIGH, LOW

Example :TRIGger:UART:IDLe HIGH /*set idle level HIGH*/

```
:TRIGger:UART:IDLe?       /* The query returns HIGH*/
```

TRIGger:CAN

The list of commands:

- TRIGger:CAN:SOURce
- TRIGger:CAN:IDLe
- TRIGger:CAN:BAUd
- TRIGger:CAN:CONdition
- TRIGger:CAN:ID
- TRIGger:CAN:DLC
- TRIGger:CAN:DATA
- TRIGger:CAN:VALId
- TRIGger:CAN:ALEVel

TRIGger:CAN:SOURce

```
     Syntax  :TRIGger:CAN:SOURce <source>
             :TRIGger:CAN:SOURce?
Description  : Set or query the trigger source for the CAN trigger
  Parameter  :<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>
     Return  : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
    Example  :TRIGger:CAN:SOURce CHANnel1  /*set trigger source CH1*/
             :TRIGger:CAN:SOURce?   /* The query returns CHANnel1*/
```

TRIGger:CAN:IDLe

```
     Syntax  :TRIGger:CAN:IDLe <idle>
             :TRIGger:CAN:IDLe?
Description  : Set or query the idle level in CAN trigger
  Parameter  :<idle> ::= <LOW | HIGH>
     Return  : The query returns LOW,HIGH
    Example  :TRIGger:CAN:IDLe LOW     /* set idle level LOW*/
             :TRIGger:CAN:IDLe?        /* The query returns LOW*/
```

TRIGger:CAN:BAUd

Syntax : TRIGger:CAN:BAUd <baud>
: TRIGger:CAN:BAUd?

Description : Set or query the baud rate triggered by the CAN, unit is bps

Parameter :<baud> ::=
<10000|20000|33300|50000|62500|83300|100000|125000|250000|500000|800000|1000000| customize >

Return : The query returns baud rate

Example : TRIGger:CAN:BAUd 4800 /*set baud rate 4800*/
: TRIGger:CAN:BAUd? /* The query returns 4800*/

TRIGger:CAN:CONdition

Syntax :TRIGger:CAN:CONdition <condition>
:TRIGger:CAN:CONdition?

Description : Set or query the trigger conditions for the UART

Parameter :<condition> ::= < FRAM_STARE | FRAM_REMO_ID | FRAM_DATA_ID | REMO/DATA_ID | DATA_ID/DATA | FRAM_REE | FRAM_OVERLOAD | ERR_ALL | ACK_ERR >

Return : The query returns FRAM_STARE | FRAM_REMO_ID | FRAM_DATA_ID | REMO/DATA_ID | DATA_ID/DATA | FRAM_REMO_ID_EXT | FRAM_DATA_ID_EXT | REMO/DATA_ID_EXT | DATA_ID/DATA_EXT | FRAM_REE | FRAM_OVERLOAD | ERR_ALL | ACK_ERR

Example :TRIGger:CAN:CONdition FRAM_STARE /*set FRAM_STARE*/
:TRIGger:CAN:CONdition? /* The query returns FRAM_STARE*/

TRIGger:CAN:ID

Syntax :TRIGger:CAN:ID <id>
:TRIGger:CAN:ID?

Description : Set or query IDENTIFIER for CAN

Parameter :<id> ::= 0 -- 28

Return : The query returns ID

Example :TRIGger:CAN:ID 25 /*set IDENTIFIERto 25*/
:TRIGger:CAN:ID? /* The query returns 25*/

TRIGger:CAN:DLC

Syntax :TRIGger:CAN:DLC <dlc>
:TRIGger:CAN:DLC?
Description : Set or query the data length code for CAN
Parameter :<dlc> ::= 4 bit
Return : The query returns DLC
Example :TRIGger:CAN:DLC 10 /*set dlc to 10*/
:TRIGger:CAN:DLC? /* The query returns 10*/

TRIGger:CAN:DATA

Syntax :TRIGger:CAN:DATA <index> <data>
:TRIGger:CAN:DATA? <index>
Description : Set or query the data value for CAN
Parameter :<data> ::= 8 bit
:<index> ::= Data index 0-3
Return : The query returns data
Example :TRIGger:CAN:DATA 2 10 /* Set the index to 2 for the data 10*/
:TRIGger:CAN:DATA? 2 /* The query returns 10*/

TRIGger:CAN:VALId

Syntax :TRIGger:CAN:VALId <index> <bool>
:TRIGger:CAN:VALId? <index>
Description : Set or query data mask for CAN
Parameter :<index> ::= 0—3
:<bool> ::= <0|1>
Return : The query returns 0, 1
Example :TRIGger:CAN:VALId 0 1 /* The masked index is 0 for data */
:TRIGger:CAN:VALId? 0 /* The query returns 1*/

TRIGger:CAN:ALEVel

Syntax : TRIGger:CAN:ALEVel <level>
: TRIGger:CAN:ALEVel?
Description : Set or query the trigger level of the CAN trigger

Parameter :<level> ::= trigger level (unit is V)
Return : The query returns the pulse width in scientific notation.
Example : TRIGger:CAN:ALEVel 0.16 /*set trigger level 160mV*/
: TRIGger:CAN:ALEVel? /* The query returns 1.600000e-01*/

TRIGger:LIN

The list of commands:

- TRIGger:LIN:SOURce
- TRIGger:LIN:IDLe
- TRIGger:LIN:BAUd
- TRIGger:LIN:CONdition
- TRIGger:LIN:ID
- TRIGger:LIN:VALId
- TRIGger:LIN:DATA
- TRIGger:LIN:ALEVel

TRIGger:LIN:SOURce

Syntax :TRIGger:LIN:SOURce <source>
:TRIGger:LIN:SOURce?
Description : Set or query the trigger source for LIN trigger
Parameter :<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>
Return : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example :TRIGger:LIN:SOURce CHANnel1 /*set trigger source CH1*/
:TRIGger:LIN:SOURce? /* The query returns CHANnel1*/

TRIGger:LIN:IDLe

Syntax :TRIGger:LIN:IDLe <idle>
:TRIGger:LIN:IDLe?
Description : Set or query the LIN-triggered idle level
Parameter :<idle> ::= <LOW | HIGH>

Return : The query returns LOW,HIGH
Example :TRIGger:LIN:IDLe LOW /*set idle level LOW*/
:TRIGger:LIN:IDLe? /* The query returns LOW*/

TRIGger:LIN:BAUd

Syntax :TRIGger:LIN:BAUd <baud>
:TRIGger:LIN:BAUd?
Description : Set or query LIN trigger baud rate, the default unit bps
Parameter :<baud> ::=<110|300|600|1200|2400|4800|9600|14400|
19200|38400|57600|115200|230400|380400|460400|921600
| customize >
Return : The query returns baud rate
Example :TRIGger:LIN:BAUd 4800 /*set baud rate 4800*/
:TRIGger:LIN:BAUd? /* The query returns 4800*/

TRIGger:LIN:CONdition

Syntax :TRIGger:LIN:CONdition <condition>
:TRIGger:LIN:CONdition?
Description : Set or query LIN trigger conditions
Parameter :<condition> ::= < INTERVAL_FIELD|SYNC_FIELD|ID_FIELD|
DATA|IDENTIFIER|ID_DATA >
Return : The query returns INTERVAL_FIELD|SYNC_FIELD|ID_FIELD|
DATA|IDENTIFIER|ID_DATA
Example :TRIGger:LIN:CONdition DATA /*set INTERVAL_END */
:TRIGger:LIN:CONdition? /* The query returns
INTERVAL_END*/

TRIGger:LIN:ID

Syntax :TRIGger:LIN:ID <id>
:TRIGger:LIN:ID?
Description : Set or query the LIN-triggered identifier
Parameter :<id> ::= 6bit
Return : The query returns ID
Example :TRIGger:LIN:ID 25 /*set ID to 25*/

:TRIGger:LIN:ID? /* The query returns 25*/

TRIGger:LIN:VALId

Syntax :TRIGger:LIN:VALId <index> <bool>
:TRIGger:LIN:VALId? <index>
Description : Set or query LIN trigger data mask
Parameter :<index> ::= 0—3
:<bool> ::= <0|1>
Return : The query returns 0, 1
Example :TRIGger:LIN:VALId 0 1 /* The masked index is 0 for data */
:TRIGger:LIN:VALId? 0 /* The query returns 1*/

TRIGger:LIN:DATA

Syntax :TRIGger:LIN:DATA <index> <data>
:TRIGger:LIN:DATA? <index>
Description : Set or query the data values that LIN triggers
Parameter :<data> ::= 8bit
:<index> ::= Data index 0-3
Return : The query returns data
Example :TRIGger:LIN:DATA 2 10 /* Set the data with index to 2 to 10*/
:TRIGger:LIN:DATA? 2 /* The query returns 10*/

TRIGger:LIN:ALEVel

Syntax :TRIGger:LIN:ALEVel <level>
:TRIGger:LIN:ALEVel?
Description : Sets or queries the trigger level when LIN triggers
Parameter :<level> ::= trigger level (unit is V)
Return : The query returns the pulse width in scientific notation.
Example :TRIGger:LIN:ALEVel 0.16 /*set trigger level 160mV*/

```
:TRIGger:LIN:ALEVel?      /* The query returns
                          1.600000e-01*/
```

TRIGger:IIC

The list of commands:

- TRIGger:IIC:SDA:SOURce
- TRIGger:IIC:SCL:SOURce
- TRIGger:IIC:CONdition
- TRIGger:IIC:ADDer
- TRIGger:IIC:DATA
- TRIGger:IIC:VALId
- TRIGger:IIC:ALEVel
- TRIGger:IIC:BLEVel

TRIGger:IIC:SDA:SOURce

Syntax	:TRIGger:IIC:SDA:SOURce <source> :TRIGger:IIC:SDA:SOURce?
Description	: Set or query the channel source of SDA in I2C trigger
Parameter	:<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>
Return	: The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example	:TRIGger:IIC:SDA:SOURce CHANnel1 /*set source to CH1*/ :TRIGger:IIC:SDA:SOURce? /* The query returns CHANnel1*/

TRIGger:IIC:SCL:SOURce

Syntax	:TRIGger:IIC:SCL:SOURce <source>

:TRIGger:IIC:SCL:SOURce?

Description : Set or query the channel source of SCL in I2C trigger

Parameter :<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>

Return : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4

Example :TRIGger:IIC:SCL:SOURce CHANnel1 /*set source to CH1*/

:TRIGger:IIC:SCL:SOURce? /* The query returns CHANnel1*/

TRIGger:IIC:CONdition

Syntax :TRIGger:IIC:CONdition <condition>

:TRIGger:IIC:CONdition?

Description : Set or query the trigger condition in I2C trigger

Parameter :<condition> ::= < START| STOP| ACK_LOST| ADDR_NO_ACK| RESTART| READ_DATA>

Return : The query returns START| STOP| ACK_LOST| ADDR_NO_ACK| RESTART| READ_DATA

Example :TRIGger:IIC:CONdition START /*set START */

:TRIGger:IIC:CONdition? /* The query returns START*/

TRIGger:IIC:ADDer

Syntax :TRIGger:IIC:ADDer <addr>

:TRIGger:IIC:ADDer?

Description : Set or query the address when the trigger condition is ADDRess or ADATa in I2C trigger

Parameter :<addr> ::= 8 位

Return : The query returns addr

Example :TRIGger:IIC:ADDer 20 /*set addr to 20*/

:TRIGger:IIC:ADDer? /* The query returns 20*/

TRIGger:IIC:DATA

Syntax :TRIGger:IIC:DATA <index> <data>

:TRIGger:IIC:DATA? <index>

Description : Set or query the data when the trigger condition is DATA or ADATa in I2C trigger

Parameter :<data> ::= 8bit

:<index> ::= Data index 0-3

Return : The query returns data

Example :TRIGger:IIC:DATA 2 10 /* Set the data with index to 2 to 10*/

:TRIGger:IIC:DATA? 2 /* The query returns 10*/

TRIGger:IIC:VALId

Syntax :TRIGger:IIC:VALId <index> <bool>

:TRIGger:IIC:VALId? <index>

Description : Set or query IIC to trigger data masking

Parameter :<index> ::= 0—3

:<bool> ::= <0|1>

Return : The query returns 0, 1

Example :TRIGger:IIC:VALId 0 1 /* The masked index is 0 for data */

:TRIGger:IIC:VALId? 0 /* The query returns 1*/

TRIGger:IIC:ACT:LEVEl

Syntax :TRIGger:IIC:ACT:LEVEl <level >

:TRIGger:IIC:ACT:LEVEl?

Description : Set or query the IIC trigger level

Parameter :<level> ::= < SCL| SDA>

Return : The query returns SCL,SDA

Example :TRIGger:IIC:ACT:LEVEl SDA /*set level SDA*/

:TRIGger:IIC:ACT:LEVEl? /* The query returns SDA*/

TRIGger:IIC:ALEVel

Syntax :TRIGger:IIC:ALEVel <level>

:TRIGger:IIC:ALEVel?

Description : Set or query the trigger level of SCL in I2C trigger

Parameter :<level> ::= trigger level (unit is V)

Return :The query returns the trigger level of SCL in scientific notation

Example :TRIGger:IIC:ALEVel 0.16 /*set trigger 160mV*/

:TRIGger:IIC:ALEVel? /* The query returns 1.600000e-01*/

TRIGger:IIC:BLEVel

Syntax :TRIGger:IIC:BLEVel <level>

:TRIGger:IIC:BLEVel?

Description : Set or query the trigger level of SDA in I2C trigger

Parameter :<level> ::= trigger level (unit is V)

Return : The query returns the trigger level of SCL in scientific notation

Example :TRIGger:IIC:BLEVel 0.16 /*set trigger level 160mV*/
:TRIGger:IIC:BLEVel? /* The query returns 1.600000e-01*/

TRIGger:SPI

The list of commands:

- TRIGger:SPI:SDA:SOURce
- TRIGger:SPI:SCL:SOURce
- TRIGger:SPI:SCK
- TRIGger:SPI:WIDth
- TRIGger:SPI:DATA
- TRIGger:SPI:MASK
- TRIGger:SPI:OVERtime
- TRIGger:SPI:ACT:LEVEl
- TRIGger:SPI:ALEVel
- TRIGger:SPI:BLEVel

TRIGger:SPI:SDA:SOURce

Syntax :TRIGger:SPI:SDA:SOURce <source>
:TRIGger:SPI:SDA:SOURce?

Description : Set or query the channel source of SDA in SPI trigger

Parameter :<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>

Return : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4

Example :TRIGger:SPI:SDA:SOURce CHANnel1 /*set source to CH1*/
:TRIGger:SPI:SDA:SOURce? /* The query returns CHANnel1*/

TRIGger:SPI:SCL:SOURce

Syntax :TRIGger:SPI:SCL:SOURce <source>
:TRIGger:SPI:SCL:SOURce?
Description : Set or query the channel source of SCL in SPI trigger
Parameter :<source> ::= < CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4>
Return : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example :TRIGger:SPI:SCL:SOURce CHANnel1 /* set source to CH1*/
:TRIGger:SPI:SCL:SOURce? /* The query returns CHANnel1*/

TRIGger:SPI:SCK

Syntax :TRIGger:SPI:SCK <slope>
:TRIGger:SPI:SCK?
Description : Set or query the clock edge in SPI trigger
Parameter :<slope> ::= < Rising| Falling>
Return : The query returns Rising, Falling
Example :TRIGger:SPI:SCK Falling /*set to Falling */
:TRIGger:SPI:SCK? /* The query returns Falling */

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 :<width> ::= 4--32
Return : The query returns data bits
Example :TRIGger:SPI:WIDth 20 /*set data bits to20*/
:TRIGger:SPI:WIDth? /* The query returns 20*/

TRIGger:SPI:DATA

Syntax :TRIGger:SPI:DATA <data>
:TRIGger:SPI:DATA?
Description : Set or query the data in SPI trigger
Parameter :<data> ::= 0—(2^{32}-1)
Return : The query returns data
Example :TRIGger:SPI:DATA 20 /*set data to 20*/
:TRIGger:SPI:DATA? /* The query returns 20*/

TRIGger:SPI:MASK

Syntax :TRIGger:SPI:MASK <mask>

:TRIGger:SPI:MASK?

Description : Set or query the mask value under SPI trigger

Parameter :<mask> ::= 0—(2^{32}-1)

Return : The query returns mask data

Example :TRIGger:SPI:MASK 20 /*set mask data to 20*/

:TRIGger:SPI:MASK? /* The query returns 20*/

TRIGger:SPI:OVERtime

Syntax :TRIGger:SPI:OVERtime <value>

:TRIGger:SPI:OVERtime?

Description : Set or query the timeout value when the trigger condition is TIMeout in SPI trigger

Parameter :<value> ::= the timeout value(unit is s)

Return : The query returns the timeout value in scientific notation.

Example :TRIGger:SPI:OVERtime 0.000003 /*set timeout value to 3us*/

:TRIGger:SPI:OVERtime? /* The query returns 3.000000e-06*/

TRIGger:SPI:ACT:LEVEl

Syntax :TRIGger:SPI:ACT:LEVEl <level >

:TRIGger:SPI:ACT:LEVEl?

Description : Set or query the SPI trigger level

Parameter :<level> ::= < SCL| SDA>

Return : The query returns SCL,SDA

Example :TRIGger:SPI:ACT:LEVEl SDA /*set to SDA*/

:TRIGger:SPI:ACT:LEVEl? /* The query returns SDA*/

TRIGger:SPI:ALEVel

Syntax :TRIGger:SPI:ALEVel <level>

:TRIGger:SPI:ALEVel?

Description : Set or query the trigger level of the SCL channel in SPI trigger

Parameter :<level> ::= the trigger level (unit is V)

Return :The query returns the timeout value in scientific notation.

Example :TRIGger:SPI:ALEVel 0.16 /*set level to 160mV*/
:TRIGger:SPI:ALEVel? /* The query returns
1.600000e-01*/

TRIGger:SPI:BLEVel

Syntax :TRIGger:SPI:BLEVel <level>
:TRIGger:SPI:BLEVel?
Description :Set or query the trigger level of the SDA channel in SPI trigger
Parameter :<level> ::= the trigger level (unit is V)
Return : The query returns the timeout value in scientific notation
Example :TRIGger:SPI:BLEVel 0.16 /*set level to 160mV*/
:TRIGger:SPI:BLEVel? /* The query returns
1.600000e-01*/

TRIGger:LOGIc

The list of commands:

- TRIGger:LOGIc:POLarity
- TRIGger:LOGIc:WHEN
- TRIGger:LOGIc:TIME
- TRIGger:LOGIc:ALEVel
- TRIGger:LOGIc:BLEVel
- TRIGger:LOGIc:CLEVel
- TRIGger:LOGIc:DLEVel

TRIGger:LOGIc:CH<n>:POLarity

Syntax :TRIGger:LOGIc:CH<n>:POLarity <type>
:TRIGger:LOGIc:CH<n>:POLarity
Description : Set or query the pattern of the specified channel pattern trigger
Parameter :<type>::= {H|L|X|R|F|D}
:<n>::= {1 | 2 | 3 | 4}
Return : The query returns H|L|X|R|F|D

Example :TRIGger:LOGIc:CH1:POLarity H /*set ch1 code to H*/
:TRIGger:LOGIc:CH1:POLarity? /* The query returns H*/

TRIGger:LOGIc:WHEN

Syntax :TRIGger:LOGIc:WHEN <when>
:TRIGger:LOGIc:WHEN?
Description : Set or query the logic conditions for pattern triggering
Parameter :<when> ::= < AND|OR >
Return : The query returns AND,OR
Example :TRIGger:LOGIc:WHEN AND /*set logic condition to AND*/
:TRIGger:LOGIc:WHEN? /* The query returns AND*/

TRIGger:LOGIC:ACT:LEVEl

Syntax :TRIGger:LOGIC:ACT:LEVEl <level >
:TRIGger:LOGIC:ACT:LEVEl?
Description : Set or query the pattern trigger level
Parameter :<level> ::= < CHANnel1| CHANne12| CHANnel3| CHANnel4>
Return : The query returns CHANnel1, CHANne2, CHANnel3, CHANnel4
Example :TRIGger:LOGIC:ACT:LEVEl CHANnel1 /*set to CHANnel1*/
:TRIGger:LOGIC:ACT:LEVEl? /* The query returns CHANnel1*/

TRIGger:LOGIc:ALEVel

Syntax :TRIGger:LOGIc:ALEVel <level>
:TRIGger:LOGIc:ALEVel?
Description : Set or query the level of ch1
Parameter :<level> ::= trigger level (unit is V)
Return : The query returns the timeout value in scientific notation
Example :TRIGger:LOGIc:ALEVel 0.16 /*set level to 160mV*/
:TRIGger:LOGIc:ALEVel? /* The query returns 1.600000e-01*/

TRIGger:LOGIc:BLEVel

Syntax	:TRIGger:LOGIc:BLEVel <level> :TRIGger:LOGIc:BLEVel?
Description	: Set or query the level of ch2
Parameter	:<level> ::= trigger level (unit is V)
Return	: The query returns the timeout value in scientific notation
Example	:TRIGger:LOGIc:BLEVel 0.16 /* set level to 160mV*/ :TRIGger:LOGIc:BLEVel? /* The query returns 1.600000e-01*/

TRIGger:LOGIc:CLEVel

Syntax	:TRIGger:LOGIc:CLEVel <level> :TRIGger:LOGIc:CLEVel?
Description	: Set or query the level of ch3
Parameter	:<level> ::= trigger level (unit is V)
Return	: The query returns the timeout value in scientific notation
Example	:TRIGger:LOGIc:CLEVel 0.16 /* set level to 160mV*/ :TRIGger:LOGIc:CLEVel? /* The query returns 1.600000e-01*/

TRIGger:LOGIc:DLEVel

Syntax	:TRIGger:LOGIc:DLEVel <level> :TRIGger:LOGIc:DLEVel?
Description	: Set or query the level of ch4
Parameter	:<level> ::= trigger level (unit is V)
Return	: The query returns the timeout value in scientific notation
Example	:TRIGger:LOGIc:DLEVel 0.16 /* set level to 160mV*/ :TRIGger:LOGIc:DLEVel? /* The query returns 1.600000e-01*/

CALibrate commands

The list of commands:

- CALibrate:STARt
- CALibrate:STATus?
- CALibrate:QUIT

CALibrate:STARt

Syntax :CALibrate:STARt

Description : The oscilloscope begins to perform the self-calibration operation

Explanation : The self-calibration operation enables the oscilloscope to reach the optimal working state rapidly to obtain the most accurate measurements. Ensure that all channels have no access to the signal until the calibration operation is completed. Most of the key function of the self-calibration operation is disabled.

CALibrate:STATus?

Syntax :CALibrate:STATus?

Description : Returns the status of calibrated

CALibrate:QUIT

Syntax :CALibrate:QUIT

Description :stop self-calibration

MATH commands

The list of commands:

- MATH:DISPlay
- MATH:OPERator
- MATH:SOURce1
- MATH:SOURce2
- MATH:SCALe
- MATH:OFFSet
- MATH:FFT:SOURce
- MATH:FFT:WINDow
- MATH:FFT:UNIT
- MATH:FFT:HSCale
- MATH:FFT:HCENter

MATH:DISPlay

Syntax	:MATH:DISPlay <bool> :MATH:DISPlay?
Description	:Enable or disable the math operation function or query the math operation status
Parameter	:<bool> ::= {{1 | ON} | {0 | OFF}
Return	: The query returns ON,OFF
Example	:MATH:DISPlay ON /*Enable the math operation function*/ :MATH:DISPlay? /* The query returns ON*/

MATH:OPERator

Syntax	:MATH:OPERator <type> :MATH:OPERator?
Description	: Set or query the operator of the math operation
Parameter	:<type> ::= < ADD| SUBTract| MULTiply| DIVision| FFT>
Return	: The query returns ADD,SUBTract,MULTiply,DIVision,FFT
Example	:MATH:OPERator ADD /*Set the operator of the math operation to integration*/

```
:MATH:OPERator?          /* The query returns ADD*/
```

MATH:SOURce1

```
    Syntax  :MATH:SOURce1 <source>
            :MATH:SOURce1?
Description : Set or query the mathematical operation source A
 Parameter  :<source> ::= <CHANnel1| CHANnel2| CHANnel3| CHANnel4>
    Return  : The query returns CHANnel1| CHANnel2| CHANnel3| CHANnel4
   Example  :MATH:SOURce1 CHANnel1 /*Set source A of algebraic
              operation to CHANnel1 */
            :MATH:SOURce1?          /* The query returns CHANnel1 */
```

MATH:SOURce2

```
    Syntax  :MATH:SOURce2 <source>
            :MATH:SOURce2?
Description : Set or query the mathematical operation source B
 Parameter  :<source> ::= <CHANnel1| CHANnel2| CHANnel3| CHANnel4>
    Return  : The query returns CHANnel1| CHANnel2| CHANnel3| CHANnel4
   Example  :MATH:SOURce2 CHANnel1 /* Set source B of algebraic
              operation to CHANnel1 */
            :MATH:SOURce2?          /* The query returns CHANnel1 */
```

MATH:SCALe

```
    Syntax  :MATH:SCALe <value>
            :MATH:SCALe?
Description : Set or query the vertical scale of the operation result
 Parameter  :<value> ::= vertical scale (1-2-5 step, unit V)
    Return  : 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*/
```

MATH:OFFSet

```
    Syntax  :MATH:OFFSet <value>
            :MATH:OFFSet?
Description : Set or query the vertical offset of the operation result
 Parameter  :<value> ::= vertical offset (unit V)
```

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

Example :MATH:OFFSet 2 /*set vertical offset to 2V */
:MATH:OFFSet? /* The query returns 2.000000e+00*/

MATH:FFT:SOURce

Syntax :MATH:FFT:SOURce <source>
:MATH:FFT:SOURce?

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

Parameter :<source> ::= < CHANnel1|CHANnel2|CHANnel3|CHANnel4>

Return : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4

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

MATH:FFT:WINDow

Syntax :MATH:FFT:WINDow <window>
:MATH:FFT:WINDow?

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

Parameter :<window> ::= < RECTangle| HANNing| HAMMing| BLACkman| TRIangle| FLATtop>

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 : The query returns RECTangle,HANNing,HAMMing,BLACkman,TRIangle,FLATtop

Example :MATH:FFT:WINDow RECTangle /*set to RECTangle*/
:MATH:FFT:WINDow? /* The query returns RECTangle*/

MATH:FFT:UNIT

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

Description : Set or query the vertical unit of the FFT operation result.
Parameter :<unit> ::= < VRMS| DB>
Return : The query returns VRMS, DB
Example :MATH:FFT:UNIT DB /*Set the vertical unit of the FFT operation result to DB*/
:MATH:FFT:UNIT? /* The query returns DB */

MATH:FFT:HSCale

Syntax :MATH:FFT:HSCale <hscale>
:MATH:FFT:HSCale?
Description : Set or query the horizontal scale of the FFT operation result,unit Hz
Parameter :<hscale> ::= <125000|250000|625000|1250000>
Explanation : You can view the detailed information of the frequency spectrum by reducing the horizontal scale.
Return : The query returns the horizontal scale in scientific notation.
Example :MATH:FFT:HSCale 125000 /*Set the horizontal scale of the FFT operation result to 125KHz*/
:MATH:FFT:HSCale? /* The query returns 1.250000e+05*/

MATH:FFT:HCENter

Syntax :MATH:FFT:HCENter
: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
Parameter : ::= the center frequency, unit is Hz
Return : The query returns the current center frequency in scientific notation.
Example :MATH:FFT:HCENter 10000000 /*set center frequency 10MHz */
:MATH:FFT:HCENter? /* The query returns 1.000000e+07*/

WAVeform commands

The list of commands：

- WAVeform:DATA:ALL?

WAVeform:DATA:ALL?

Syntax :WAVeform:DATA:ALL?

Description : Gets any stored data

Return : Returns the waveform packet containing the data header in the form of a string

remark : For the first time, this command is parsed to data[x]

data[0]-data[1] (2 bits): #9

data[2]-data[10](9 bits): Represents the byte length of the current packet

data[11]-data[19](9 bits): The total length of bytes representing the amount of data

data[20]-data[28](9 bits): Represents the byte length of the data that has been uploaded

data[29](1 bits)： Represents the current running state

data[30](1 bits)： Represents the state of the trigger

data[31]-data[34]（ 4 bits） :bias of ch1

data[35]-data[38]（ 4 bits） :bias of ch2

data[39]-data[42]（ 4 bits） :bias of ch3

data[43]-data[46]（ 4 bits） :bias of ch4

data[47]-data[53]（ 7 bits） :volt of ch1

data[54]-data[60]（ 7 bits） :volt of ch2

data[61]-data[67]（ 7 bits） :volt of ch3

data[68]-data[74]（ 7 bits） :volt of ch4

data[75]-data[78]（ 4 bits） :Represents the enabling of the channel [1-4]

data[79]-data[87]（ 9 bits） :Indicated sampling rate

data[88]-data[93]（ 6 bits） :Sampling multiple

data[94]-data[102]（ 9 bits）:The current frame displays trigger time

data[103]-data[111]（9 bits）:The current frame shows the starting point of the data start point

data[112]-data[127]（16 bits）:Reserved bit

The data of waveform data[x] was resolved before the data was read again

data[0]-data[1] (2 bits) : #9

data[2]-data[10](9 bits):Represents the byte length of the current packet

data[11]-data[19](9bits):The total length of bytes representing the amount of data

data[20]-data[28](9 bits):Represents the byte length of the data that has been uploaded

data[29]-data[x]: Represents the waveform data corresponding to the current data header

DISPlay commands

The list of commands:

- DISPlay:TYPE
- DISPlay:WBRightness
- DISPlay:GRID
- DISPlay:GBRightness

DISPlay:TYPE

Syntax	:DISPlay:TYPE <type> :DISPlay:TYPE?
Description	: Set or query the display mode of the waveform on the screen
Parameter	:<type> ::= < VECTors| DOTS>
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	: The query returns VECTors, DOTS
Example	:DISPlay:TYPE DOTS /*set type to DOTS*/ :DISPlay:TYPE? /* The query returns DOTS*/

DISPlay:WBRightness

Syntax	:DISPlay:WBRightness <value> :DISPlay:WBRightness?
Description	: Set or query the waveform brightness.
Parameter	:<value> ::=0—100
Return	: The query returns value
Example	:DISPlay:WBRightness 50 /*set to 50*/ :DISPlay:WBRightness? /* The query returns 50*/

DISPlay:GRID

Syntax	:DISPlay:GRID <type>
	:DISPlay:GRID?
Description	: Set or query the grid type of screen display.
Parameter	:<type> ::= < DOTTed| REAL>
Return	: The query returns DOTTed, REAL
Example	:DISPlay:GRID DOTTed /*set grid type to DOTTed */
	:DISPlay:GRID? /* The query returns DOTTed */

DISPlay:GBRightness

Syntax	:DISPlay:GBRightness <value>
	:DISPlay:GBRightness?
Description	: Set or query the brightness of the screen grid
Parameter	:<value> ::=0—100
Return	: The query returns value
Example	:DISPlay:GBRightness 50 /*set to 50*/
	:DISPlay:GBRightness? /* The query returns 50*/

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.

The list of commands:

- CURSor:MODE
- CURSor:MANual:TYPE
- CURSor:MANual:SOURce
- CURSor:MANual:AX
- CURSor:MANual:AXValue?
- CURSor:MANual:AY
- CURSor:MANual:AYValue?
- CURSor:MANual:BX
- CURSor:MANual:BXValue?
- CURSor:MANual:BY
- CURSor:MANual:BYValue?
- CURSor:TRACk:SOURcea
- CURSor:TRACk:SOURceb
- CURSor:TRACk:AX
- CURSor:TRACk:AXValue?
- CURSor:TRACk:AY?
- CURSor:TRACk:AYValue?
- CURSor:TRACk:BX
- CURSor:TRACk:BXValue?
- CURSor:TRACk:BY?
- CURSor:TRACk:BYValue?

CURSor:MODE

Syntax :CURSor:MODE <type>
:CURSor:MODE?

Description : Set or query the cursor measurement mode.

Parameter :<type> ::= < OFF| MANual| TRACk >

Explanation :MANual: enable the manual cursor measurement mode
:TRACK: enable the track cursor measurement mode

Return : The query returns OFF,MANual, TRACK

Example :CURSor:MODE TRACK /*set measure mode to TRACK*/
:CURSor:MODE? /* The query returns TRACK*/

CURSor:MANual:TYPE

Syntax :CURSor:MANual:TYPE <type>
:CURSor:MANual:TYPE?

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

Parameter :<type> ::= <X | Y | XY>

Return : The query returns X,Y,XY

Example :CURSor:MANual:TYPE X /*set to X*/
:CURSor:MANual:TYPE? /* The query returns X*/

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 :<source> ::= < CHANnel1|CHANnel2|CHANnel3|CHANnel4| MATH>

Return : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4, MATH

Example :CURSor:MANual:SOURce CHANnel1 /*set source to CH1*/
:CURSor:MANual:SOURce? /* The query returns CHANnel1*/

CURSor:MANual:AX

Syntax :CURSor:MANual:AX <value>
:CURSor:MANual:AX?

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

Parameter :<type> ::= 0--770

Explanation : The horizontal and vertical positions of the cursor are defined by the pixel coordinate ofthe screen.

Return : The query returns 0—770

Example :CURSor:MANual:AX 200 /*Set the horizontal position of cursor A to 200*/
:CURSor:MANual:AX? /* 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 : The query returns the X value of cursor A in scientific notation.

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

CURSor:MANual:AY

Syntax :CURSor:MANual:AY <value>
:CURSor:MANual:AY?

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

Parameter :<type> ::= 0--400

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

Return : The query returns 0—400

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

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

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

CURSor:MANual:BX

Syntax :CURSor:MANual:BX <value>

:CURSor:MANual:BX?

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

Parameter :<type> ::= 0--770

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

Return : The query returns 0—770

Example :CURSor:MANual:BX 200 /*Set the horizontal position of cursor B to 200*/

:CURSor:MANual:BX? /* The query returns 200*/

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 : The query returns the X value of cursor B in scientific notation

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

CURSor:MANual:BY

Syntax :CURSor:MANual:BY <value>

:CURSor:MANual:BY?

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

Parameter :<type> ::= 0--400

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

Return : The query returns 0—400

Example :CURSor:MANual:BY 200 /*Set the vertical position of cursor B to 200*/

:CURSor:MANual:BY? /* The query returns 200*/

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 : The query returns the Y value of cursor B in scientific notation

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

CURSor:TRACk:SOURcea

Syntax :CURSor:TRACk:SOURcea <source>

:CURSor:TRACk:SOURcea?

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

Parameter :<source> ::= < CHANnel1|CHANnel2|CHANnel3|CHANnel4| MATH >

Return : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4, MATH

Example :CURSor:TRACk:SOURcea CHANnel1 /*set the channel source to CH1*/

:CURSor:TRACk:SOURcea? /* The query returns CHANnel1*/

CURSor:TRACk:SOURceb

Syntax :CURSor:TRACk:SOURceb <source>

:CURSor:TRACk:SOURceb?

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

Parameter :<source> ::= < CHANnel1|CHANnel2|CHANnel3|CHANnel4| MATH >

Return : The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4, MATH

Example :CURSor:TRACk:SOURceb CHANnel1 /*set to CH1*/

:CURSor:TRACk:SOURceb? /* The query returns CHANnel1*/

CURSor:TRACk:AX

Syntax :CURSor:TRACk:AX <value>

:CURSor:TRACk:AX?

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

Parameter :<type> ::= 0--770

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

Return : The query returns 0—770

Example :CURSor:TRACk:AX 200 /*Set the horizontal position of cursor A to 200*/

:CURSor:TRACk:AX? /* 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 : The query returns the X value of cursor A in scientific notation.

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

CURSor:TRACk:AY?

Syntax :CURSor:TRACk:AY?

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

Return : The query returns value

Example :CURSor:TRACk:AY? /* The query returns 288*/

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

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

CURSor:TRACk:BX

Syntax :CURSor:TRACk:BX <value>

:CURSor:TRACk:BX?

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

Parameter :<type> ::= 0--770

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

Return : The query returns 0—770

Example :CURSor:TRACk:BX 200 /*Set the horizontal position of cursor to 200*/

:CURSor:TRACk:BX? /* The query returns 200*/

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

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

CURSor:TRACk:BY?

Syntax :CURSor:TRACk:BY?

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

Return : The query returns value

Example :CURSor:TRACk:BY? /* The query returns 288*/

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

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

MEASure commands

The list of commands:

- MEASure:ENABle
- MEASure:SOURce
- MEASure:ADISplay
- MEASure:ITEM
- MEASure:GATE:ENABle
- MEASure:GATE:AY
- MEASUre:GATE:BY

MEASure:ENABle

Syntax	:MEASure:ENABle <bool>
	:MEASure:ENABle?
Description	: Set or query the measurement function status
Parameter	:<bool> ::= {{1 | ON} | {0 | OFF}
Return	: The query returns ON,OFF
Example	:MEASure:ENABle ON /*open measure */
	:MEASure:ENABle? /* The query returns ON*/

MEASure:SOURce

Syntax	:MEASure:SOURce <source>
	:MEASure:SOURce?
Description	: Set or query the source of the current measurement parameter
Parameter	:<source> ::= < CHANnel1|CHANnel2|CHANnel3|CHANnel4>
Return	: The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4
Example	:MEASure:SOURce CHANnel1 /*set to CH1*/
	:MEASure:SOURce? /* The query returns CHANnel1*/

MEASure:ADISplay

Syntax :MEASure:ADISplay <bool>
:MEASure:ADISplay?

Description : Open or close all measurements, or query the current full measurement status.

Parameter :<bool> ::= {{1 | ON} | {0 | OFF}

Return : The query returns ON,OFF

Example :MEASure:ADISplay ON /*open*/
:MEASure:ADISplay? /* The query returns ON*/

MEASure:CHANnel<n>:ITEM

Syntax :MEASure:CHANnel<n>:ITEM <type>

Description : Open the corresponding parameter measurements

Parameter :<type>::=<FREQuency|PERiod|VAVG|VMAX|VMIN|VPP|VTOP|VMID|VBASE|VAMP|RMS|OVERshoot|PREShoot|PERIodrms|PERIodmean|RTIMe|FTIMe|PPULses|NPULses|PDUTy|NDUTy|FRR|FFF|FOVshoot|RPREshoot|BWIDth|FRF|FFR|LRR|LRF|LFR|LFF>

Example :MEASure:CHANnel1:ITEM VPP /*open vpp of ch1*/

MEASure:CHANnel<n>:ITEM?

Syntax :MEASure:CHANnel<n>:ITEM? <type>

Description : The query channel n corresponds to the measurement of the item type

Parameter :<n> ::= <1|2|3|4>
:<type>::=<FREQuency|PERiod|VAVG|VMAX|VMIN|VPP|VTOP|VMID|VBASE|VAMP|RMS|OVERshoot|PREShoot|PERIodrms|PERIodmean|RTIMe|FTIMe|PPULses|NPULses|PDUTy|NDUTy|FRR|FFF|FOVshoot|RPREshoot|BWIDth|FRF|FFR|LRR|LRF|LFR|LFF>

Return : The query returns the measured value of the corresponding item

Example :MEASure:CHANnel1:ITEM? VPP

MEASure:GATE:ENABle

Syntax :MEASure:GATE:ENABle <bool>
:MEASure:GATE:ENABle?

Description : Set or query the current gate state

Parameter :<bool> ::= {{1 | ON} | {0 | OFF}

Return : The query returns ON,OFF

```
Example  :MEASure:GATE:ENABle  ON    /*open gete function*/
         :MEASure:GATE:ENABle?       /* The query returns ON*/
```

MEASure:GATE:AY

```
Syntax  :MEASure:GATE:AY <value>
        :MEASure:GATE:AY?
Description  : Sets or queries the value of cursor A
Parameter  :<value> ::= 0--400
Return  : The query returns value
Example  :MEASure:GATE:AY  100   /*set A value to 100*/
         :MEASure:GATE:AY?       /* The query returns 100*/
```

MEASUre:GATE:BY

```
Syntax  :MEASure:GATE:BY <value>
        :MEASure:GATE:BY?
Description  : Sets or queries the value of cursor B
Parameter  :<value> ::= 0--400
Return  : The query returns value
Example  :MEASure:GATE:BY  100   /* set B value to 100 */
         :MEASure:GATE:BY?       /* The query returns 100*/
```

MASK commands

The list of commands:

- MASK:EANBle
- MASK:SOURce
- MASK:MDISplay
- MASK:SOOutput
- MASK:OUTPut
- MASK:X
- MASK:Y
- MASK:CREate

MASK:EANBle

Syntax	:MASK:EANBle <bool> :MASK:EANBle?
Description	: Enable or disable the pass/fail test or query the status of the past/fail test.
Parameter	:<bool> ::= {{1 | ON} | {0 | OFF}
Return	: The query returns ON,OFF
Example	:MASK:EANBle ON /*Enable the pass/fail test*/ :MASK:EANBle? /* The query returns ON*/

MASK:SOURce

Syntax	:MASK:SOURce <source> :MASK:SOURce?
Description	: Set or query the source of the pass/fail test
Parameter	:<source> ::= < CHANnel1|CHANnel2|CHANnel3|CHANnel4| MATH>
Return	: The query returns CHANnel1, CHANnel2, CHANnel3, CHANnel4, MATH
Example	:MASK:SOURce CHANnel1 /*set source to CH1*/ :MASK:SOURce? /* The query returns CHANnel1*/

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 :<bool> ::= {{1 | ON} | {0 | OFF}

Return : The query returns ON,OFF

Example :MASK:MDISplay ON /**/

:MASK:MDISplay? /* The query returns ON*/

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 :<bool> ::= {{1 | ON} | {0 | OFF}

Explanation :open: 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) at 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 at the rear panel outputs a pulse each time a failed waveform is detected.

Return : The query returns ON,OFF

Example :MASK:SOOutput ON /**/

:MASK:SOOutput? /* The query returns ON*/

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 :<bool> ::= {{1 | ON} | {0 | OFF}

Explanation :open: when failed waveforms are detected, there are display and output and the beeper sounds (not related to the on/off state of the sound).

:off: when failed waveforms are detected, there are display and output but the beeper does not sound.

Return : The query returns ON,OFF

Example :MASK:OUTPut ON /**/
:MASK:OUTPut? /* The query returns ON*/

MASK:X

Syntax :MASK:X <value>
:MASK:X?

Description : Set or query the horizontal adjustment parameter in the pass/fail test mask. The default unit is div.

Parameter :<value> ::= 0.02 -- 4

Return : 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 <value>
:MASK:Y?

Description : Set or query the vertical adjustment parameter in the pass/fail test mask. The default unit is div.

Parameter :<value> ::= 0.04 -- 5.12

Return : 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 and is not in the run state

SYSTem commands

The list of commands:

- SYSTem:GAM?
- SYSTem:RAM?
- SYSTem:PON
- SYSTem:LANGuage
- SYSTem:LOCKed

SYSTem:GAM?

Syntax :SYSTem:GAM?

Description :Query the number of grids in the horizontal direction of the instrument screen.

Explanation : The query always returns 12.

SYSTem:RAM?

Syntax :SYSTem:RAM?

Description : Query the number of analog channels of the instrument.

Explanation : The query always returns 4.

SYSTem:PON

Syntax :SYSTem:PON <value>

:SYSTem:PON?

Description : Set or query the system configuration to be recalled when the oscilloscope is powered on again after power-off.

Parameter :<value> ::= < LATest| DEFault>

Return : The query returns LATest, DEFault

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 LATest */

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 :<bool> ::= {{1 | ON} | {0 | OFF}

Return : The query returns ON,OFF

SYSTem:SET:TIME

Syntax :SYSTem:SET:TIME <time>

Description :set system time

Parameter :<time> ::= <year;month;day;hour;min;sec;>

Example :SYSTem:SET:TIME 2017;6;12;13;18;35;

SYSTem:TIME?

Description : query system time

DDS commands

The list of commands:

- DDS:SWITch
- DDS:TYPE
- DDS:FREQ
- DDS:AMP
- DDS:OFFSet
- DDS:DUTY
- DDS:WAVE:MODE
- DDS:MODE:TYPE
- DDS:MODE:WAVE:TYPE
- DDS:MODE:FREQ
- DDS:MODE:DEPThordeviation
- DDS:BURSt:SWITch
- DDS:BURSt:TYPE
- DDS:BURSt:CNT
- DDS:BURSt:SRC
- DDS:BURSt:SLOPE
- DDS:BURSt:GATE:POLArity
- DDS:BURSt:TRIGger

DDS:SWITch

Syntax	:DDS:SWITch <bool> :DDS:SWITch?
Description	: Sets or queries the source status
Parameter	:<bool> ::= {{1 | ON} | {0 | OFF}
Return	: The query returns ON,OFF
Example	:DDS:SWITch ON /*open dds*/ :DDS:SWITch? /* The query returns ON*/

DDS:TYPE

Syntax :DDS:TYPE <type>
:DDS:TYPE?

Description : Set or query the signal source waveform type

Parameter :<type> ::= { SINE| SQUAre| RAMP| EXP| NOISe| DC| ARB1 | ARB2| ARB3| ARB4}

Return : The query returns SINE,SQUAre,RAMP,EXP,NOISe,DC,ARB1 ,ARB2,ARB3,ARB4

Example :DDS:TYPE SINE /* Sets the signal source waveform type as sine wave */
:DDS:TYPE? /* The query returns SINE*/

DDS:FREQ

Syntax :DDS:FREQ <freq>
:DDS:FREQ?

Description : To set or query the frequency of the signal

Parameter :<freq> ::= unit is Hz

Return : Query returns frequency with scientific count method

Example :DDS:FREQ 1000 /* Sets the frequency of the signal source waveform 1khz*/
:DDS:FREQ? /* The query returns 1.00000e+03*/

DDS:AMP

Syntax :DDS:AMP <amp>
:DDS:AMP?

Description : Set or query the range of signal source signals

Parameter :<amp> ::= unit is V

Return : The query is returned by the scientific count method

Example :DDS:AMP 1 /* Set signal source waveform amplitude 1V*/
:DDS:AMP? /* The query returns 1.00000e+00*/

DDS:OFFSet

Syntax :DDS:OFFSet <offset>
:DDS:OFFSet?

Description : Set or query the offset of signal source signals

Parameter :<offset> ::= unit is V

Return : The query returns the offset by the scientific count method

```
Example  :DDS:OFFSet  0.5        /* Set signal source waveform
                                    offset 0.5V*/
         :DDS:OFFSet?          /*return 5.00000e-01*/
```

DDS:DUTY

```
    Syntax  :DDS:DUTY <duty>
            :DDS:DUTY?
Description  : Sets or queries the space ratio of signal source signals
  Parameter  :<duty> ::= 0--99
     Return  : The query returns the null ratio value
    Example  :DDS:DUTY  50      /* Sets the signal source signal to the
                                  air ratio for 50%*/
            :DDS:DUTY?          /*return 50*/
```

DDS:WAVE:MODE

```
    Syntax  :DDS:WAVE:MODE <bool>
            :DDS:WAVE:MODE?
Description  : Sets or queries the signal source modulation status
  Parameter  :<bool> ::= {{1 | ON} | {0 | OFF}
     Return  :return ON,OFF
    Example  :DDS:WAVE:MODE  ON    /* Turn on the signal source
                                    modulation */
            :DDS:WAVE:MODE?        /* return ON*/
```

DDS:MODE:TYPE

```
    Syntax  :DDS:MODE:TYPE <type>
            :DDS:MODE:TYPE?
Description  : Set or query the source modulation type
  Parameter  :<type> ::= { AM| FM}
     Return  : Returned by the query AM, FM
    Example  :DDS:MODE:TYPE  AM     /*Set the signal source modulation
                                      type AM*/
            :DDS:MODE:TYPE?         /*Returned by the queryAM*/
```

DDS:MODE:WAVE:TYPE

Syntax :DDS:MODE:WAVE:TYPE <type>
:DDS:MODE:WAVE:TYPE?

Description : Set or query the modulation wave type when the signal source is modulated

Parameter :<type> ::= { SINE| SQUAre| RAMP}

Return :return SINE, SQUAre, RAMP

Example
```
:DDS:MODE:WAVE:TYPE SINE  /*Set the signal source
                           modulated wave typeSINE*/
:DDS:MODE:WAVE:TYPE?      /*return SINE*/
```

DDS:MODE:FREQ

Syntax :DDS:MODE:FREQ <freq>
:DDS:MODE:FREQ?

Description : Set or query the frequency of modulated waves when the signal source is modulated

Parameter :<freq> ::= unit is Hz

Return : Query returns frequency with scientific count method

Example
```
:DDS:MODE:FREQ  1000    /* Set the frequency of the signal
                           source modulation 1khz*/
:DDS:MODE:FREQ?         /*return  1.00000e+03*/
```

DDS:MODE:DEPThordeviation

Syntax :DDS:MODE:DEPThordeviation <value>
:DDS:MODE:DEPThordeviation?

Description : Setting or querying signal source modulation for deviation or depth

Parameter :AM:<value> ::= Modulation depth
:FM:<value> ::= deviation

Return : The modulation type returns the value of the modulation depth for AM
: The modulation type is the query return deviation for FM

Example
```
:AM
:DDS:MODE:DEPThordeviation 50 /* Set the modulation depth
                                 for 50*/
:DDS:MODE:DEPThordeviation?   /*return 50*/
```

```
:FM
:DDS:MODE:DEPThordeviation 1000 /* Set the bias 1khz*/
:DDS:MODE:DEPThordeviation?   /*return 1000*/
```

DDS:BURSt:SWITch

```
     Syntax  :DDS:BURSt:SWITch <bool>
             :DDS:BURSt:SWITch?
Description  : Set or query the signal source burst state
  Parameter  :<bool> ::= {{1 | ON} | {0 | OFF}
     Return  :return ON,OFF
    Example  :DDS:BURSt:SWITch  ON    /* Open source burst */
             :DDS:BURSt:SWITch?       /*return ON*/
```

DDS:BURSt:TYPE

```
     Syntax  :DDS:BURSt:TYPE <type>
             :DDS:BURSt:TYPE?
Description  : Set or query the source burst type
  Parameter  :<type> ::= { N_CYCLE| GATE}
     Return  :return N_CYCLE, GATE
    Example  :DDS:BURSt:TYPE  GATE    /*Set the type of burstGATE*/
             :DDS:BURSt:TYPE?       /*return GATE*/
```

DDS:BURSt:CNT

```
     Syntax  :DDS:BURSt:CNT <cnt>
             :DDS:BURSt:CNT?
Description  : Set or query the number of signal cycles of the signal
               source
  Parameter  :<value> ::= value
     Return  : Query return integer
    Example  :DDS:BURSt:CNT  2    /* Set the number of cycles 2*/
             :DDS:BURSt:CNT?       /*return 2*/
```

DDS:BURSt:SRC

Syntax :DDS:BURSt:SRC <src>
:DDS:BURSt:SRC?
Description : Set or query the source of the signal source
Parameter :<src> ::= { EXT| MANU}
Return : Query return EXT, MANU
Example :DDS:BURSt:SRC EXT /* Setting the burst source is external */
:DDS:BURSt:SRC? /*return EXT*/

DDS:BURSt:SLOPE

Syntax :DDS:BURSt:SLOPE <slope>
:DDS:BURSt:SLOPE?
Description : Set or query the signal source burst edge
Parameter :<slope> ::= { RAISING| FALLING}
Return : Query return RAISING, FALLING
Example :DDS:BURSt:SLOPE FALLING /* Set the burst edge to drop along */
:DDS:BURSt:SLOPE? /*return FALLING */

DDS:BURSt:GATE:POLArity

Syntax :DDS:BURSt:GATE:POLArity <slope>
:DDS:BURSt:GATE:POLArity?
Description : Set or query the signal source burst gate control polarity
Parameter :<slope> ::= { POSITIVE| NEGATIVE}
Return : Query return POSITIVE, NEGATIVE
Example :DDS:BURSt:GATE:POLArity NEGATIVE /* Setting gate control polarity is negative */
:DDS:BURSt:GATE:POLArity? /*return NEGATIVE */

DDS:BURSt:TRIGger

Syntax :DDS:BURSt:TRIGger
Description : The next signal source burst

DDS:ARB:DAC16:BIN

Syntax :DDS:ARB:DAC16:BIN <binary_block_data>

Description : Download waveform data, <binary_block_data> Represents the download of binary data, <binary_block_data> Binary data blocks that begin with a #, “#508192” Binary data, “The "5" after "#" indicates that there are five characters representing the length of the data,“8192” There are 8192 bytes representing binary data。Each wave point corresponds to a binary number of two bytes（For example, point 1024 corresponds to the binary number 0x0400, the low byte of data in front, and the high byte in the front, so 0004）, So the number of bytes has to be even

injection：The number of arbitrary wave shapes must be 4096