EN: This Datasheet/Document is presented by the manufacturer.

Please visit our website for pricing and availability at www.hestore.hu.

UNI-T

Manual

UTD2000E Series Programmable Digital Oscilloscope

June 2017
Uni-Trend Technology (China) Co., Ltd.

Warranty and Statement

Copyright

2017 Uni-Trend Technology (China) Co., Ltd.

Brand Information

UNI-T is the registered trademark of Uni-Trend Technology (China) Co., Ltd.

Software Version

00.00.01

Software upgrade may have some change and add more function, please subscribe **UNI-T** website to get the new version or contact **UNI-T.**

Statement

- UNI-T products are protected by patents (including obtained and pending) in China and other countries and regions.
- UNI-T reserves the right to change specifications and prices.
- The information provided in this manual supersedes all previous publications.
- Information provided in this manual is subject to change without prior notice.
- UNI-T shall not be liable for any errors that may be contained in this manual. For any incidental or consequential damages, arising out of the use or the information and deductive functions provided in this manual.
- Without the written permission of UNI-T, this manual cannot photocopied, reproduced or adapted.

Product Certification

UNI-T has certified that the product conforms to China national product standard and industry product standard as well as ISO9001:2008 standard and ISO14001:2004 standard. UNI-T will go further to certificate product to meet the standard of other member of the international standards organization.

Contact Us

If you have any question or problem, you can contact us,

Website：https://www.uni-trend.com

e format, symbols, parameters, and abbreviations of the SCPI command.

Instruction Format

The SCPI command is a tree-like hierarchy consisting of multiple subsystems, each consisting of a root keyword and one or more hierarchical key words. **The command line usually begins with a colon ":"; Keywords are separated by the colon ":", followed by optional parameter settings. The command keyword is separated by spaces from the first parameter. The command string must end with a newline <NL> character. Add the question mark "? " after the command line. It is usually indicated that this feature is being queried.**

Symbol Description

The following four symbols are not part of SCPI command, it cannot send with the command. It usually used as supplementary description of command parameter.

- **Brace { }** usually contains multiple optional parameters, it should select one parameter when send command.

  Such as DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE} command

- **Vertical Bar |** used to separated multiple parameters, it should select one parameter when send command.

  Such as DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE} command

- **Square Brackets [ ]** the contents in square brackets (command keywords) can omissible. If the parameter is ignoret, the instrument will set the parameter as the default value.

  Such as MEASure:NDUTy? [<source>] command, it presents current channel

- **Triangular Brackets < >** The parameter in the brackets must be replaced with a valid value.

  Such as use DISPlay:GRID:BRIGhtness 30 form to send DISPlay:GRID:BRIGhtness <count> command.

Parameter Description

The parameter in this manual can divide into five types: Boolean, Integer, Real, Discrete, ASCII string

- **Boolean**

  Parameter value can set “ON” (1) or “OFF” (0)

  Such as SYSTem:LOCK {{1 | ON} | {0 | OFF}}

- **Integer**

  Parameter can take any valid integer value unless there have some other descriptions.

  Such as command: DISPlay:GRID:BRIGhtness <count> , parameter of <count> can take integer from 0~100

  Note: Do not set decimal as parameter, otherwise it may occur error.

- **Real**

  Parameter can take any valid integer value unless there have some other descriptions.

  Such as for command CH1, CHANnell: OFFSet <offset> , parameter of <offset> can take integer value.

- **Discrete**

  Parameter can only take some specified numbers or characters.

  Such as command DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE} parameter can only take FULL, GRID, CROSS, NONE

- **ASCII Character String**

  String parameter contain all ASCII string sets. Strings must begin and end with paired quotes; it can use single or double quotation marks. The quotation and delimiter can also be part of a string by typing it twice and not adding any characters.

  Such as set IP SYST:COMM:LAN:IPAD "192.168.1.10".

Shorthand Rule

All command can identify capital and small letter, if command need enter shorthand, it should be all capital letter.

Data Return

Data return is divided into single data and batch data. The single data return is the corresponding parameter type, in which the real return type is presents by the scientific notation method. The part before e retains three figure behind the decimal point, and the e part retains three figure; the batch return must be obey IEEE 488.2# string data format, '#'+ the length of character bits[fixed to one character]+ ASCII valid value+ valid data+ end string['\n']

Such as #3123xxxxxxxxxxxxxxxxxx\n presents 123 strings batch data return format, '3' presents "123" occupies three character bits.

Notes: * presents return valid data.

SCPI Command

IEEE488.2 Common Command

*IDN?

- Command format:
  *IDN?
- Functional description:
  For query manufacture name, model, product serial number and software version.
- Return format:
  Manufacture name, model, product serial number, software version separated by dot mark.
- For example:
  UNI-T Technologies, UPO2000CS, UPO1000, 00.00.01

*RST

- Command format:
  *RST
- Functional description:
- Restore factory settings and clear the entire error message, send and receive queue buffers.

SYSTem Command

The command used for the oscilloscope basic operation, including operating control, full qwerty lock, error queue and system setup.

:RUN

- Command format:
  :RUN
- Functional description:
  Oscilloscope start to sampling, execute :STOP command to stop it.

:STOP

- Command format:
  :STOP
- Functional description:
  Stop sampling function, execute :RUN command to start it.

:AUTO

- Command format:

  :AUTO

- Functional description:

  Set control parameter automatically, use auto function to achieve the best display effect.

:SYSTem:LOCK

- Command format:

  :SYSTem:LOCK {{1 | ON} | {0 | OFF}}

  :SYSTem:LOCK?

- Functional description:

  For lock/unlock full qwerty.

- Return format:

  Query return full qwerty lock status, 0 presents unlock，1 presents lock.

- For example:

:SYSTem:LOCK ON/:SYST:LOCK 1	full qwerty lock
:SYSTem:LOCK OFF/:SYST:LOCK 0	unlock full qwerty
:SYSTem:LOCK?	query return 1，it presents locked

:SYSTem:ERRor

- Command format:

  :SYSTem:ERRor

  :SYSTem:ERRor?

- Functional description:

  Empty error message queue.

- Return format:

  Query return the last time error message, return error message by query format <message number>,<message content>, <message number> is a integer，<message content> is ASCII character string with double quotation marks.

  like-113,"Undefined header; command cannot be found"

- For example:

:SYSTem:ERR	empty error message queue
:SYSTem:ERR?	query return： -113,"Undefined header; command cannot be found" presents undefined command head

:SYSTem:SETup

- Command format:

  :SYSTem:SETup <setup_data>

  :SYSTem:SETup?

- Functional description:

  Set system configuration data, <setup_data> is conform to IEEE 488.2 # format of binary data.

  In sending，data flow is followed command character and complete send in time.

  In reading, make sure there has enough buffer to receive data flow, otherwise it may occurred abnormal.

- Return format:

  Query return system setup data, return format is conform to IEEE 488.2 # format of binary data.

:SYSTem:LANGuage

- Command format:

  :SYSTem:LANGuage { ENGLish | SIMPlifiedchinese | TRADitionalchinese }

  :SYSTem:LANGuage?

- Functional description:

  Set system language.

- Return format:

  Query return{ ENGLish | SIMPlifiedchinese | TRADitionalchinese }.

- For example:

  :SYSTem:LANGuage ENGL　　set system language as English

  :SYSTem:LANGuage?　　query returnENGLish

:SYSTem:CAL

- Command format:

  :SYSTem:CAL

- Functional description:

  Set system self-calibration, during self-calibration, oscilloscope can not communication.

:SYSTem:CLEAr

- Command format:

  :SYSTem:CLEAr

- Functional description:

  Empty system saved waveform and setup data.

:SYSTem:CYMOmeter

- Command format:

  :SYSTem:CYMOmeter {1 | ON} | {0 | OFF}

  :SYSTem:CYMOmeter?

- Functional description:

  The switch of frequency meter.

- Return format:

  Query return frequency meter status， 1 presents on, 0 presents off.。

- For example:

  :SYSTem:CYMOmeter ON open frequency meter

  :SYSTem:CYMOmeter? query return 1

:SYSTem:CYMOmeter:FREQuency?

- Command format:

  :SYSTem:CYMOmeter:FREQuency?

- Functional description:

  Acquire measured frequency value of frequency meter.

- Return format:

  Query return measured frequency value of frequency meter, invalid value return *.

- For example:

  :SYSTem:CYMOmeter:FREQuency? query return 1.20000E+3

:SYSTem:SQUare:SELect

- Command format:

  :SYSTem:SQUare:SELect { 10Hz | 100Hz | 1KHz | 10KHz }

  :SYSTem:SQUare:SELect?

- Functional description:

  Select square wave output.

- Return format:

  Query return{ 10Hz | 100Hz | 1KHz | 10KHz }。

- For example:

  :SYSTem:SQUare:SELect 10Hz select 10Hz of saquare wave output

  :SYSTem:SQUare:SELect? query return 10Hz

:SYSTem:MNUDisplay

- Command format:

  :SYSTem:MNUDisplay { 5S | 10S | 20S | INFinite}

:SYSTem:MNUDisplay?

- Functional description:

  Set menu display time, INFinite presents menu is display all time.

- Return format:

  Query return{ 5S | 10S | 20S | INFinite }。

- For example:

  :SYSTem:MNUDisplay 5S set menu display base as 5s, automatic fold.

  :SYSTem:MNUDisplay? query return 5S

:SYSTem:BRIGhtness

- Command format:

  :SYSTem:BRIGhtness <count>

  :SYSTem:BRIGhtness?

- Functional description:

  Set screen luminance, <count> take value from 1~100，the bigger the number, the brighter the screen.

- Return format:

  Query return the current screen luminance.

- For example:

  :SYSTem:BRIGhtness 50 screen luminance as 50

  :SYSTem:BRIGhtness? query return 50

:SYSTem:VERSion?

- Command format:

  :SYSTem:VERSion?

- Return format:

  Query return version information, 128 bytes character string.

  HW is hardware version number, SW is software version numer，PD is created date, ICV is protocol version number.

- For example:

  :SYST:VERS? query return HW:1.0;SW:1.0;PD:2014-11-20;ICV:1.4.0

:SYSTem:COMMunicate:LAN:APPLy

- Command format:

  :SYSTem:COMMunicate:LAN:APPLy

- Functional description:

  The current internet parameter take effect immediately.

:SYSTem:COMMunicate:LAN:GATEway

- Command format:

  :SYSTem:COMMunicate:LAN:GATEway <gateway>

  :SYSTem:COMMunicate:LAN:GATEway?

- Functional description:

  For the default gateway, <gateway> is belong to ASCII character string parameter, format is xxx.xxx.xxx.xxx.

- Return format:

  Query return the default gateway.

- For example:

  :SYST:COMM:LAN:GATE "192.168.1.1" set default gateway as 192.168.1.1

  :SYST:COMM:LAN:GATE? query return 192.168.1.1

:SYSTem:COMMunicate:LAN:SMASK

- Command format:

  :SYSTem:COMMunicate:LAN:SMASK <submask>

  :SYSTem:COMMunicate:LAN:SMASK?

- Functional description:

  Set subnet mask, <submask> is belong to ASCII character string parameter, format is xxx.xxx.xxx.xxx.

- Return format:

  Query return subnet mask.

- For example:

  :SYST:COMM:LAN:SMASK "255.255.255.0" set subnet mask as 255.255.255.0

  :SYST:COMM:LAN:SMASK? query return 255.255.255.0

:SYSTem:COMMunicate:LAN:IPADdress

- Command format:

  :SYSTem:COMMunicate:LAN:IPADdress <ip>

  :SYSTem:COMMunicate:LAN:IPADdress?

- Functional description:

  Set IP address, <ip> is belong to ASCII character string parameter, format is xxx.xxx.xxx.xxx。

- Return format:

  Query return IP address

- For example:

:SYST:COMM:LAN:IPAD "192.168.1.10" set IP address as 192.168.1.10
:SYST:COMM:LAN:IPAD? query return192.168.1.10

:SYSTem:COMMunicate:LAN:DHCP

- Command format:
  :SYSTem:COMMunicate:LAN:DHCP {{1 | ON} | {0 | OFF}}
  :SYSTem:COMMunicate:LAN:DHCP?
- Functional description:
  For switch configuration mode（auto IP）and（manual IP）.
- Return format:
  Query return dynamic configuration mode, 0 presents（manual IP），1 presents（auto IP）.
- For example:
  :SYST:COMM:LAN:DHCP ON turn on IP dynamic configurarion mode
  :SYST:COMM:LAN:DHCP? query return 1

:SYSTem:COMMunicate:LAN:MAC?

- Command format:
  :SYSTem:COMMunicate:LAN:MAC?
- Return format:
  Query return MAC physical address.
- For example:
  :SYST:COMM:LAN:MAC? query return 00-2A-A0-AA-E0-56

KEY Command

This command is to control button and knob on the oscilloscope panel.

:KEY:<key>

- Command format:
  :KEY:<key>
  :KEY:<key>:LOCK { {1 | ON} | {0 | OFF} }
  :KEY:<key>:LOCK?
  :KEY:<key>:LED?
- Functional description:
  Set button function and button lock status, <key> definition and description, see Appendix 1: Key Table.
- Return format:
  Query return key lock status or LED status.
  Lock status：0 presents unlock, 1 presents lock；

LED light staus：0 presents light off, 1 presents light on.

- For example：

:KEY:AUTO	set oscilloscope control parameter automatically
:KEY:AUTO:LOCK ON/OFF	lock/unlock button
:KEY:AUTO:LOCK?	query return this key lock status, 1 presents locked
:KEY:AUTO:LED?	query return LED status，0 presents light off

CHANnel Command

This command is to set channel independently. <n> take value as 1/2/3/4/5/6/7/8/9, it presents {CH1/CH2/ CH3/ CH4/ MATH/ REF-A/ REF-B/ REF-C/ REF-D}.

Notes: this command in instruction set is not only valid for physical channel, but also valid for MATH and reference channel parameter.

:CHANnel<n>:BWLimit

- Command format：

  :CHANnel<n>:BWLimit {{1|ON}|{0|OFF}}

  :CHANnel<n>:BWLimit?
- Functional description：

  Set bandwidth limit function is ON (turn on bandwidth to 20MHz，to reduce display noise) or OFF (turn off bandwidth to full display）
- Return format：

  Query return 1 or 0，it presents ON or OFF.
- For example：

:CHAN1:BWL ON	turn on channel 1 bandwidth limit
:CHAN1:BWL?	query return 1，it presents channel 1 bandwidth limit is turn on

:CHANnel<n>:COUPling

- Command format：

  :CHANnel<n>:COUPling {DC|AC|GND}

  :CHANnel<n>:COUPling?
- Functional description：

  Set channel coupling mode, DC presents AC and DC component of input signal; AC presents DC component of blocking input signal; GND (grounding) presents break input signal.
- Return format：

  Query AC, DC or GND.
- For example：

:CHAN1:COUP DC	set channel 1 coupling mode as DC
:CHAN1:COUP?	query return DC

:CHANnel<n>:DISPlay

- Command format:

  :CHANnel<n>:DISPlay { {1|ON} | {0|OFF} }

  :CHANnel<n>:DISPlay?

- Functional description:

  The switch of the specified channel.

- Return format:

  Query return 1 or 0，it presents ON or OFF.

- For example:

  :CHAN1:DISP ON　　turn on channel 1

  :CHAN1:DISP?　　query return 1，it presents channel 1 is turn on

:CHANnel<n>:INVert

- Command format:

  :CHANnel<n>:INVert { {1|ON} | {0|OFF} }

  :CHANnel<n>:INVert?

- Functional description:

  Set wave inversion function is ON (turn on) or OFF (back to normal display）.

- Return format:

  Query return 1 or　0，it presents ON or OFF.

- For example:

  :CHAN1:INV OFF　　turn off channel 1 inversion function

  :CHAN1:INV?　　query return 0，it presents channel 1 inversion function is off

:CHANnel<n>:PROBe

- Command format:

  :CHANnel<n>:PROBe　{ 0.01X | 0.02X | 0.05X | 0.1X | 0.2X | 0.5X | 1X | 2X | 5X | 10X |20X |50X |100X |200X |500X |1000X }

  :CHANnel<n>:PROBe?

- Functional description:

  Set the probe attenuation factor corresponding to the probe.

- Return format:

  Query return { 0.01X | 0.02X | 0.05X | 0.1X | 0.2X | 0.5X | 1X | 2X | 5X | 10X |20X |50X |100X |200X |500X |1000X }.

- For example:

  :CHAN1:PROB 10X　　set channel 1 probe attenuation factor as 10

  :CHAN1:PROB?　　query return 10X

:CHANnel<n>:OFFSet

- Command format:

  :CHANnel<n>:OFFSet <offset>

  :CHANnel<n>:OFFSet?

- Functional description:

  Set wave offset in vertical position, <n> take value as 1/2/3/4/5, it presents {CH1/CH2/CH3/ CH4/ MATH }.

- Return format:

  Query return offset value by scientific notation method, unit is related with the current channel unit.

- For example:

  :CHAN1:OFFS 20V　　set channel 1 vertival offset as 20V

  :CHAN1:OFFS?　　query return 2.000e001

:CHANnel<n>:SCALe

- Command format:

  :CHANnel<n>:SCALe　{<scale> | UP | DOWN}

  :CHANnel<n>:SCALe?

- Functional description:

  Set voltage base in vertical position.

  <scale>: voltage base value;

  UP: plus one position based on the current position;

  DOWN : minus one position based on the current position.

- Return format:

  Query return the current voltage position value by scientific notation method, unit is V.

- For example:

  :CHAN1:SCAL 20V　　set channel 1 voltage base as 20V

  :CHAN1:SCAL?　　query return 2.000e001。

  :CHAN1:SCAL UP　　plus one position based on 20V voltage base

:CHANnel<n>:UNITs

- Command format:

  :CHANnel<n>:UNITs {VOLTs|AMPeres }

  :CHANnel<n>:UNITs?

- Functional description:

  Set channel unit as VOLTs and AMPeres.

- Return format:

  Query return VOLTs, AMPeres.

- For example:

  :CHAN1:UNIT VOLT — set channel 1 unit as VOLTs

  :CHAN1:UNIT? — query return VOLTs

:CHANnel<n>:VERNier

- Command format:

  :CHANnel<n>:VERNier { {1|ON} | {0|OFF} }

  :CHANnel<n>:VERNier?

- Functional description:

  Set position accommodation mode, ON presents Fine tuning，to further subdivides between coarse tuning settings to improve vertical resolution; OFF presents Coarse tuning， based on 1-2-5 system to set vertical sensitivity.

- Return format:

  Query return 1 or 0，it presents ON or OFF.

- For example:

  :CHAN1:VERN ON — turn on channel 1 fine tuning function

  :CHAN1:VERN? — query return 1

:CHANnel<n>:SELect

- Command format:

  :CHANnel<n>:SELect

  :CHANnel<n>:SELect?

- Functional description:

  For channel selection.

- Return format:

  Query return 1 or 0，it presents ON or OFF.

- For example:

  :CHAN1:SELect — select channel 1

  :CHAN1:SELect? — query return 1，it presents channel 1 has been selected

TIMebase Command

This command is to change the current channel horizontal scale (time base) and horizontal position of trigger in memory (trigger offset). Horizontal scale change will make waveform expand or shrink relative to screen center. Horizontal position change will make wave shift relative to screen center.

:TIMebase:MODE

- Command format:

  :TIMebase:MODE {MAIN | WINDow}

  :TIMebase:MODE?
- Functional description:

  Set timebase mode, MAIN or WINDow (zoom timebase <Zoomed>)
- Return format:

  Query return MAIN or WINDow.
- For example:

  :TIM:MODE MAIN set timebase mode as MAIN

  :TIM:MODE? query return MAIN

:TIMebase:OFFSet

- Command format:

  :TIMebase:OFFSet <offset>

  :TIMebase:OFFSet?
- Functional description:

  For adjust MAIN offset value, wave shift relative to screen center.
- Return format:

  Query return <offset> value by scientific notation，unit is S.
- For example:

  :TIM:OFFS 1s set MAIN offset value as 1s

  :TIM:OFFS? query return 1.000e000

:TIMebase:WINDow:OFFSet

- Command format:

  :TIMebase:WINDow:OFFSet <offset>

  :TIMebase:WINDow:OFFSet?

- Functional description:

  For adjust WINDow offset value（zoom timebase <Zoomed>）,wave shift relative to screen center.

- Return format:

  Query return<offset> value by scientific notation method unit is S.

- For example:

  :TIM:WIND:OFFS 1　　set WINDow offset value as 1s

  :TIM:WIND:OFFS?　　query return 1.000e000

:TIMebase:SCALe

- Command format:

  :TIMebase:SCALe {<scale> | UP | DOWN}

  :TIMebase:SCALe?

- Functional description:

  Set timebase position of MAIN, s/div.

  <scale>：timebase value；

  UP：plus one position based on the current position；

  DOWN ：minus one position based on the current position.

- Return format:

  Query return< scale> value by scientific notation method, unit is s/div.

- For example:

  :TIM:SCAL 2　　set MAIN offset value as 2s/div

  :TIM:SCAL?　　query return 2.000e000

:TIMebase:WINDow:SCALe

- Command format:

  :TIMebase:WINDow:SCALe {<scale> | UP | DOWN}

  :TIMebase:WINDow:SCALe?

- Functional description:

  For adjust WINDow timebase（zoom timebase <Zoomed>）, that is s/div.

- Return format:

  query return < scale> value by scientific notation method, in s/div.

- For example:

  :TIM:WIND:SCAL 2　　set WINDow timebase offset value as 2s/div

  :TIM:WIND:SCAL?　　query return 2.000e000

FUNCtion Command

Display the operational result plus, minus, multiply, divide, AND, OR, negation, exclusive OR and FFT of CH1,CH2,CH3,CH4 waveform, set filter to use expression operation.

:FUNCtion:MATH:MODE

- Command format:

  FUNCtion:MATH:MODE {MATH | FFT | FILTer}

  FUNCtion:MATH:MODE?
- Functional description:

  Select MATH function mode.
- Return format:

  Query return {MATH | FFT | FILTer }
- For example:

  FUNC:MATH:MODE FFT select MATH mode as FFT mode

  FUNC:MATH:MODE? query return FFT

:FUNCtion:OPERation

- Command format:

  :FUNCtion:OPERation {ADD | SUBTract | MULTiply | DIVide }

  :FUNCtion:OPERation?
- Functional description:

  Set functional operator, including basic operation and logic operation, they are plus, minus, multiply, divide, AND, OR, negation, exclusive OR.
- Return format:

  query return {ADD | SUBTract | MULTiply | DIVide }
- For example:

  :FUNCtion:OPERation ADD use puls operator: src1+src2

  :FUNCtion:OPERation? query return ADD

:FUNCtion:SOURce<m>

- Command format:

  :FUNCtion:SOURce<m> <value>

  :FUNCtion:SOURce<m>?
- Functional description:

  SOURce <m> presents source 1 or source 2, <m> take value as 1, 2.

  SOURce1 is to select the first source of operator and math function or it can be single source of Filter and FFT.

  SOURce2 is to select the second source of operator and math function. Single source of Filter,

and FFT is not applicable.

<value> presents CHANnel<n>， <n> take vaule as 1/2/3/4{CH1/ CH2/ CH3/ CH4}.

- Return format：

Query return <n> take valu as 1/2/3/4/5/6/7/8/9.

- For example：

:FUNCtion:SOUR1 CHAN1	choose channel 1 as source 1
:FUNCtion:SOUR1?	query return1
:FUNCtion:SOUR2 CHAN2	choose channel 2 as source 2
:FUNCtion:SOUR2?	query return 2
:FUNCtion:OPERation ADD	add source 1 and source 2

:FUNCtion:FFT:WINDow

- Command format：

:FUNCtion:FFT:WINDow {RECTangular|HANNing|HAMMing|BMAN}

:FUNCtion:FFT:WINDow?

- Functional description：

FFT windowing to intercept signal. RECT, HANN, HAMM, BMAN is rectangular window, Hanning window, Hamming window and Blacman window respectively.

- Return format：

Query return {RECTangular|HANNing|HAMMing|BMAN}.

- For example：

:FUNCtion:SOUR1 CHAN1	set channel 1 as source 1
:FUNC:FFT:WIND HAMM	windowing Hamming window
:FUNC:FFT:WIND?	query return HAMMing

:FUNCtion:FFT:VTYPe

- Command format：

:FUNCtion:FFT:VTYPe {VRMS|DBRMS}

:FUNCtion:FFT:VTYPe?

- Functional description：

Set unit of FFT vertical diection as dbrms or VRMS. dbrms presents root mean square of power; VRMS presents root mean square of voltage.

- Return format：

Query return VRMS, DBRMS

- For example：

:FUNCtion:SOUR1 CHAN1	set channel 1 as source 1
:FUNC:FFT:VTYP VRMS	set unit of FFT vertical diection as root mean square of voltage
:FUNC:FFT:VTYP?	query return VRMS

:FUNCtion:FFT:FREQuency

- Command format:

  :FUNCtion:FFT:FREQuency?

- Functional description:

  The center frequency of spectrum and waveform after obtaining FFT.

- Return format:

  Query return1.000e003，unit is Hz.

- For example:

  :FUNCtion:SOUR1 CHAN1　　set channel 1 as source 1

  :FUNC:FFT:FREQ?　　query return 1.000e003

:FUNCtion:FILTer:TYPE

- Command format:

  :FUNCtion:FILTer:TYPE {LP|HP|BP|BS}

  :FUNCtion:FILTer:TYPE?

- Functional description:

  Filter type setting. LP, HP, BP, BS is low-pass filter, high-pass filter, band-pass filter and band-stop filer respectively.

- Return format:

  Query return LP, HP, BP, BS.

- For example:

  :FUNCtion:SOUR1 CHAN1　　set channel 1 as source 1

  :FUNC:FILT:TYPE BP　　set to band-pass filter

  :FUNC:FILT:TYPE?　　query return BP

:FUNCtion:FILTer:FREQuency:HIGH

- Command format:

  :FUNCtion:FILTer:FREQuency:HIGH < freq>

  :FUNCtion:FILTer:FREQuency:HIGH?

- Functional description:

  The setting of the upper frequency limit of filter. It is applicable for high-pass filter, band-pass filer and band-stop filter.

- Return format:

  Query return 1.000e003，unit is Hz.

- For example:

  :FUNCtion:SOUR1 CHAN1　　set channel 1 as source 1

  :FUNC:FILT:FREQ:HIGH 1KHz　　set the upper frequency limit of filter as 1kHz

:FUNC:FILT:FREQ:HIGH? query return 1.000e003

:FUNCtion:FILTer:FREQuency:LOW

- Command format:

  :FUNCtion:FILTer:FREQuency:LOW <freq>

  :FUNCtion:FILTer:FREQuency:LOW?

- Functional description:

  The setting of the lower frequency limit of filter. It is applicable for high-pass filter, band-pass filer and band-stop filter.

- Return format:

  Query return 6.000e001，unit is Hz.

- For example:

  :FUNC:SOUR1 CHAN1 set channel 1 as source 1

  :FUNC:FILT:FREQ:LOW 60Hz set the upper frequency limit of filter as 60Hz

  :FUNC:FILT:FREQ:LOW? query return 6.000e001

MEASure Command

This command is for oscilloscope basic measurement, auto turn on parameter measurement and get test result, return test result by scientific notation method.

:MEASure:ALL

- Command format:

  :MEASure:ALL {{1 | ON} | {0 | OFF}}

  :MEASure:ALL?

- Functional description:

  The switch setting of all measurement function.

- Return format:

  Query return turn on all measurement function whether is turn on

- For example:

  :MEASure:ALL ON turn on all measurement function

  :MEASure:ALL? query return 1

:MEASure:CLEar

- Command format:

  :MEASure:CLEar

- Functional description:

  Empty the current measured parameter.

- For example:

  :MEAS:CLE delete the current measured parameter

:MEASure:SOURce

- Command format:

  :MEASure:SOURce <source>

  :MEASure:SOURce?

- Functional description:

  For select the main measure source, <source> is CHANnel<n>， n take value as 1, 2, 3, 4.

- Return format:

  Query return{CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH}.

- For example:

  :MEAS:SOUR CHAN1 select channel 1 as measure source

  :MEAS:SOUR? return CHANnel1

:MEASure:SLAVe:SOURce

- Command format:

  :MEASure:SLAVe:SOURce <source>

  :MEASure:SLAVe:SOURce?

- Functional description:

  For select slave information source, <source> is CHANnel<n>， n take value as 1, 2, 3, 4.

- Return format:

  query return{CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH}.

- For example:

  :MEAS:SLAV:SOUR CHAN1 select channel 1 as slave information source

  :MEAS:SLAV:SOUR? return CHANnel1

:MEASure:PDUTy?

- Command format:

  :MEASure:PDUTy? [<source>]

- Functional description:

  Measuring positive duty ratio of the specified channel waveform, <source> take vaule as CHANnel1, CHANnel2, CHANnel3 or CHANnel4. Ignore measns the current channel.

- Return format:

  Query return 5.000e001， i unit is %.

:MEASure:NDUTy?

- Command format:

  :MEASure:NDUTy? [<source>]

- Functional description:

Measuring positive duty ratio of the specified channel waveform, <source> take vaule as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH，Ignore measns the current channel.

- Return format：

  Query return 5.000e001，unit is %.

:MEASure:PDELay?

- Command format：

  :MEASure:PDELay? [<source1>,<source2>]

- Functional description：

  Measuring <source1> and <source2> relative to time delay of rise edge, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH，Ignore measns the current channel.

- Return format：

  query return-1.000e-004，unit is S.

- For example：

  Measuring the relative to time delay of rise edge.

  :MEASure:PDEL? CHAN1,CHAN2

:MEASure:NDELay?

- Command format：

  :MEASure:NDELay? [<source1>,<source2>]

- Functional description：

  Measuring <source1> and <source2> relative to time delay of fall edge, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH，Ignore measns the current channel.

- Return format：

  Query return -1.000e-004，unit is S.

- For example：

  Measuring the relative to time delay of fall edge.

  :MEASure:NDEL? CHAN1,CHAN2

:MEASure:PHASe?

- Command format：

  :MEASure:PHASe? [<source1>,<source2>]

- Functional description：

  Timing measuring <source1> relative to <source2> the advance or hysteretic of time quantum, it presents by degree, 360°as a period, source take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH. Ingore means the current channel.

- Return format：

  query return 1.000e001，unit is degree.

- For example：

Measuring <source1> relative to <source2> the advance or hysteretic of time quantum.

:MEASure:PHAS? CHAN1,CHAN2

:MEASure:VPP?

- Command format:

  :MEASure:VPP? [<source>]

- Functional description:

  Measuring peak-to-peak value of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH. Ingore means the current channel.

- Return format:

  Query return 3.120e000，unit is V.

:MEASure:VMAX?

- Command format:

  :MEASure:VMAX? [<source>]

- Functional description:

  Measuring the maximum value of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH，Ingore means the current channel.

- Return format:

  Query return 2.120e000，unit is V.

:MEASure:VMIN?

- Command format:

  :MEASure:VMIN? [<source>]

- Functional description:

  Measuring the minimum value of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH，Ingore means the current channel.

- Return format:

  Query return -2.120e000，unit is V.

:MEASure:VAMPlitude?

- Command format:

  :MEASure:VAMPlitude? [<source>]

- Functional description:

  Measuring the amplitude value of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH，Ingore means the current channel.

- Return format:

  Query return 3.120e000，unit is V.

:MEASure:VTOP?

- Command format:

  :MEASure:VTOP? [<source>]

- Functional description:

  Measuring the top value of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH，Ingore means the current channel.

- Return format:

  Query return 3.120e000，unit is V.

:MEASure:VBASe?

- Command format:

  :MEASure:VBASe? [<source>]

- Functional description:

  Measuring the bottom value of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH，Ingore means the current channel.

- Return format:

  query return-3.120e000，unit is V.

:MEASure:VMIDdle?

- Command format:

  :MEASure:VMIDdle? [<source>]

- Functional description:

  Measuring the middle value of the specified channel waveform,<source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH，Ingore means the current channel.

- Return format:

  Query return0.120e000，unit is V.

:MEASure:VAVerage?

- Command format:

  :MEASure:VAVerage? [<interval>][,<source>]

- Functional description:

  Measuring the average value of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH. If there has no designated <source>, the current channel as <source> by default; <interval> appoint the measurement separation, take value as CYCLe and DISPlay, CYCLe presents integer cycle period, DISPlay presents full screen. If there has no designated <interval>, take DISPlay by default.

- Return format:

  Query return1.120e000，unit is V.

:MEASure:VRMS?

- Command format:

  :MEASure:VRMS? [<interval>][,<source>]

- Functional description:

  Measuring the root mean square value of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH. If there has no designated <source>, the current channel as <source> by default; <interval> appoint the measurement separation, take value as CYCLe and DISPlay, CYCLe presents integer cycle period, DISPlay presents full screen. If there has no designated <interval>, take DISPlay by default.

- Return format:

  Query return 1.230e000，unit is V.

:MEASure:AREa?

- Command format:

  :MEASure:AREa? [<interval>][,<source>]

- Functional description:

  Measuring the area of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH. If there has no designated <source>, the current channel as <source> by default; <interval> appoint the measurement separation, take value as CYCLe and DISPlay, CYCLe presents integer cycle period, DISPlay presents full screen. If there has no designated <interval>, take DISPlay by default.

- Return format:

  Query return 3.456e002，unit is V.s.

:MEASure:OVERshoot?

- Command format:

  :MEASure:OVERshoot? [<source>]

- Functional description:

  Measuring overshoot of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH. Ignore means the current channel.

- Return format:

  query return1.230e002，unit is V.

:MEASure:PREShoot?

- Command format:

  :MEASure:PREShoot? [<source>]

- Functional description:

  Measuring prshoot of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3, CHANnel4 and MATH. Ignore means the current channel.

- Return format:

  Query return 1.230e-002，unit is V.

:MEASure:FREQuency?

- Command format:

  :MEASure:FREQuency? [<source>]

- Functional description:

  Measuring frequency of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return 2.000e003，unit is Hz.

:MEASure:RISetime?

- Command format:

  :MEASure:RISetime? [<source>]

- Functional description:

  Measuring rise time of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return5.000e-005，unit is s.

:MEASure:FALLtime?

- Command format:

  :MEASure:FALLtime? [<source>]

- Functional description:

  Measuring fall time of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return 5.000e-005，unit is s.

:MEASure:PERiod?

- Command format:

  :MEASure:PERiod? [<source>]

- Functional description:

  Measuring period of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return 5.000e-003，unit is s.

:MEASure:PWIDth?

- Command format:

  :MEASure:PWIDth? [<source>]

- Functional description:

  Measuring period of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return 5.000e-003，unit is s.

:MEASure:NWIDth?

- Command format:

  :MEASure:NWIDth? [<source>]

- Functional description:

  Measuring negative pulse width of the specified channel waveform, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return 5.000e-003，unit is s.

:MEASure:FRR?

- Command format:

  :MEASure:FRR? [<source1>,<source2>]

- Functional description:

  Measuring time of the first rise edge between <source1> and <source2>, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return 5.000e-003，unit is s.

:MEASure:FRF?

- Command format:

  :MEASure:FRF? [<source1>,<source2>]

- Functional description:

  Measuring time between the first rise edge of <source1> and the first fall edge of <source2>, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return 5.000e-003，unit is s.

:MEASure:FFR?

- Command format:

  :MEASure:FFR? [<source1>,<source2>]

- Functional description:

  Measuring time between the first fall edge of <source1> and the first fall edge of <source2>, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return 5.000e-003，unit is s.

:MEASure:FFF?

- Command format:

  :MEASure:FFF? [<source1>,<source2>]

- Functional description:

  Measuring time of the first fall edge between <source1> and <source2>, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return 5.000e-003，unit is s.

:MEASure:LRR?

- Command format:

  :MEASure:LRR? [<source1>,<source2>]

- Functional description:

  Measuring time between the first rise edge of <source1> and the last rise edge of <source2>, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return 5.000e-003，unit is s.

:MEASure:LRF?

- Command format:

  :MEASure:LRF? [<source1>,<source2>]

- Functional description:

  Measuring time between the first rise edge of <source1> and the last fall edge of <source2>, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return 5.000e-003，unit is s.

:MEASure:LFR?

- Command format:

  :MEASure:LFR? [<source1>,<source2>]

- Functional description:

  Measuring time between the first fall edge of <source1> and the last rise edge of <source2>, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

  Query return 5.000e-003，unit is s.

:MEASure:LFF?

- Command format:

  :MEASure:LFF? [<source1>,<source2>]

- Functional description:

  Measuring time between the first fall edge of <source1> and the last fall edge of <source2>, <source> take value as CHANnel1, CHANnel2, CHANnel3 and MATH. Ignore means the current channel.

- Return format:

Query return 5.000e-003，unit is s.

TRIGger Command

This command is to control trigger sweep mode and trigger standard, trigger decide when oscilloscope start to collect data and display waveform.

Trigger Control

:TRIGger:MODE

- Command format：

  :TRIGger:MODE　<mode>

  :TRIGger:MODE?

- Functional description：

  Set trigger mode.

  <mode> divided into EDGE（edge trigger）, PULSe（pulse width trigger）, VIDeo（video trigger）, SLOPe（slope trigger）, ALTernat（alternative trigger）.

- Return format：

  Query return trigger mode

- For example：

  :TRIGger:MODE EDGE　　edge trigger

  :TRIGger:MODE?　　query return EDGE

:TRIGger:SWEep

- Command format：

  :TRIGger:SWEep　{AUTO|NORMal|SINGle}

  :TRIGger:SWEep?

- Functional description：

  Set trigger sweep mode.

  AUTO：If there has no trigger term, internal will produce trigger signal to force trigger.

  NORMal：Trigger only when in trigger term.

  SINGle：In trigger term, trigger one time and stop.

- Return format：

  Query return trigger sweep mode {AUTO|NORMal|SINGle}.

- For example：

  :TRIGger:SWEep AUTO　　set channel 1 as AUTO mode

  :TRIGger:SWEep?　　query return AUTO

:TRIGger:LEVel:ASETup

- Command format:
  :TRIGger:LEVel:ASETup
- Functional description:
  Set trigger level on the vertical center point of signal amplitude.
- For example:
  :TRIG:LEVel:ASETup set trigger level on the center point

:TRIGger:STATus?

- Command format:
  :TRIGger:STATus?
- Functional description:
  Query the current the oscilloscope running status.
- Return format:
  Query return STOP/ARMED/READY/TRIGED/AUTO/SCAN /RESET / REPLAY/ WAIT.
- For example:
  :TRIGger:STATus? query return AUTO

:TRIGger:LEVel

- Command format:
  :TRIGger:LEVel <level>
  :TRIGger:LEVel?
- Functional description:
  Set the trigger level value of NORMal mode. <level> value must set based on voltage base and the conversion of screen information.
- Return format:
  Query return<level> value，unit is V.
- For example:
  :TRIG:LEV 2 set the trigger leve as 2V
  :TRIG:LEV? query return 2.000e000

:TRIGger:LEVel:LOW

- Command format:
  :TRIGger:LEVel:LOW <level>

:TRIGger:LEVel:LOW?

- Functional description:

  Set the low-level value of slope trigger, <level> value must set based on voltage base and the conversion of screen information.

- Return format:

  query return <level> value，unit is V.

- For example:

  :TRIG:LEV:LOW 2　　set the trigger leve as 2V

  :TRIG:LEV:LOW?　　query return 2.000e000

:TRIGger:LEVel:HIGH

- Command format:

  :TRIGger:LEVel:HIGH　<level>

  :TRIGger:LEVel:HIGH?

- Functional description:

  Set the high-level value of slope trigger, <level> value must set based on voltage base and the conversion of screen information.

- Return format:

  Query return <level> value，unit is V.

- For example:

  :TRIG:LEV:HIGH 2　　set the trigger leve as 2V

  :TRIG:LEV:HIGH?　　query return 2.000e000

:TRIGger:SOURce

- Command format:

  :TRIGger:SOURce　<source>

  :TRIGger:SOURce?

- Functional description:

  Set single information source trigger, input channel(CHANnel1, CHANnel 2, CHANnel 3, CHANnel 4）, external trigger(EXT, EXT5), AC Line. Only EDGE/ PULSe / VIDeo support AC Line, EXT and EXT5.

  <source> presents information source trigger,

  CHANnel<n>|EXT|EXT5|ACLine，<n> take value as 1, 2, 3, 4.

- Return format:

  Query return information source trigger { CHANnel1| CHANnel2| CHANnel3| CHANnel4|EXT|EXT5|ACLINE}.

- For example:

  :TRIGger:SOUR CHAN1　　set channel 1 as edge tigger

  :TRIGger:SOUR?　　query return CHANnel1

:TRIGger:COUPling

- Command format:
  :TRIGger:COUPling {DC|AC|LF|HF|NOISE}
  :TRIGger:COUPling?
- Functional description:
  For coupling mode, DC, AC, LF, HF and NOISE, except VIDeo.
- Return format:
  Query return coupling mode {DC|AC|LF|HF|NOISE}.
- For example:
  :TRIGger:COUPling AC set edge trigger as AC
  :TRIGger:COUPling? query return AC

Edge Trigger

:TRIGger:EDGE:SLOPe

- Command format:
  :TRIGger:EDGE:SLOPe {POSitive|NEGative|ALTernation}
  :TRIGger:EDGE:SLOPe?
- Functional description:
  Set the type of edge trigger, POSitive, NEGative and ALTernation.
- Return format:
  Query return edge trigger { POSitive | NEGative | ALTernation }.
- For example:
  :TRIGger:EDGE:SLOP POS set the edge trigger as POSitive
  :TRIGger:EDGE:SLOP? query return POSitive

Pulse Width Trigger

:TRIGger:PULSe:QUALifier

- Command format:
  :TRIGger:PULSe:QUALifier {GREaterthan | LESSthan | EQUal| RANG}
  :TRIGger:PULSe:QUALifier?
- Functional description:

Set the term of pulse time, GREaterthan, LESSthan and EQUal.

- Return format:

  Query return{ GREaterthan | LESSthan | EQUal }.

- For example:

  :TRIGger:PULSe:QUALifier GRE set the pulse term as GREaterthan

  :TRIGger:PULSe:QUALifier? query return GREaterthan

:TRIGger:PULSe:POLarity

- Command format:

  :TRIGger:PULSe:POLarity {POSitive | NEGative}

  :TRIGger:PULSe:POLarity?

- Functional description:

  Set pulse polarity, POSitive and NEGative.

- Return format:

  Query return{ POSitive | NEGative }.

- For example:

  :TRIGger:PULSe:POL POS set pulse polarity as POSitive

  :TRIGger:PULSe:POL? query return POSitive

:TRIGger:PULSe:TIME

- Command format:

  :TRIGger:PULSe:TIME <time>

  :TRIGger:PULSe:TIME?

- Functional description:

  Set the time interval of pulse width trigger.

- Return format:

  Query return the current time interval, unit is s.

- For example:

  :TRIGger:PULSe:TIME 1 set the pulse width as 1s

  :TRIGger:PULSe:TIME? query return 1.000e000

:TRIGger:PULSe:TIME:UPPer

- Command format:

  :TRIGger:PULSe:TIME <time>

  :TRIGger:PULSe:TIME?

- Functional description:

  Set the time interval of pulse width trigger.

- Return format:

  Query return the current time interval, unit is s.

- For example:

  :TRIGger:PULSe:TIME 1 set the pulse width as 1s

  :TRIGger:PULSe:TIME? query return 1.000e000

:TRIGger:PULSe:TIME:LOWer

- Command format:

  :TRIGger:PULSe:TIME <time>

  :TRIGger:PULSe:TIME?

- Functional description:

  Set the time interval of pulse width trigger.

- Return format:

  the current time interval, unit is s.

- For example:

  :TRIGger:PULSe:TIME 1 set the pulse width as 1s

  :TRIGger:PULSe:TIME? query return 1.000e000

Video Trigger

:TRIGger:VIDeo:MODE

- Command format:

  :TRIGger:VIDeo:MODE { ODD | EVEN | LINE| ALINes}

  :TRIGger:VIDeo:MODE?

- Functional description:

  Set the syc mode of video trigger, ODD, EVEN, LINE and ALINes.

- Return format:

  Query return{ ODD | EVEN | LINE| ALIN}.

- For example:

  :TRIGger:VIDeo:MODE ODD set the sync mode as ODD

  :TRIGger:VIDeo:MODE? query return ODD

:TRIGger:VIDeo:STANdard

- Command format:

:TRIGger:VIDeo:STANdard { NTSC | PAL | SECam }

:TRIGger:VIDeo:STANdard?

- Functional description:

Set the video standard.

- Return format:

Query return { NTSC | PAL | SECam }.

- For example:

:TRIGger:VIDeo:STANdard NTSC set the video standard as NTSC

:TRIGger:VIDeo:STANdard? query return NTSC

:TRIGger:VIDEO:LINE

- Command format:

:TRIGger:VIDEO:LINE <value>

:TRIGger:VIDEO:LINE?

- Functional description:

Set the specified line of video sync, <value> presents the specified line, the range is related to the video standard.

- Return format:

Query return the current specified line.

- For example:

:TRIG:VIDEO:LINE 50 set the specified line of video sync as 50

:TRIG:VIDEO:LINE? query return 50

:TRIGger:VIDEO:SRATe?

- Command format:

:TRIGger:VIDEO:SRATe?

- Functional description:

Acquire the slew rate of video.

- Return format:

Query return the current slew rate, return format is conform to the format of binary data IEEE 488.2#.

Slope Trigger

:TRIGger:SLOPe:QUALifier

- Command format:

:TRIGger:SLOPe:QUALifier {GREaterthan | LESSthan | EQUal}

:TRIGger:SLOPe:QUALifier?

- Functional description:

Set the term of slope time, GREaterthan, LESSthan and EQUal.

- Return format:

Query return {GREaterthan | LESSthan | EQUal}.

- For example:

:TRIGger:SLOPe:QUALifier GRE set the slope term as GREaterthan

:TRIGger:SLOPe:QUALifier? query return GREaterthan

:TRIGger:SLOPe:SLOPe

- Command format:

:TRIGger:SLOPe:SLOPe {POSitive|NEGative}

:TRIGger:SLOPe:SLOPe?

- Functional description:

Set the type of slope trigger, POSitive and NEGative.

- Return format:

Query return{POSitive|NEGative}.

- For example:

:TRIGger:SLOPe:SLOPe POS set the slope trigger as POSitive

:TRIGger:SLOPe:SLOPe? query return POSitive

:TRIGger:SLOPe:TIME

- Command format:

:TRIGger:SLOPe:TIME <time>

:TRIGger:SLOPe:TIME?

- Functional description:

Set the time interval of slope trigger.

- Return format:

Query return the current time interval, unit is s.

- For example:

:TRIGger:SLOPe:TIME 1 set the time interval of slope trigger as 1s

:TRIGger:SLOPe:TIME? query return 1.000e000

:TRIGger:SLOPe:THReshold

- Command format:
  :TRIGger:SLOPe:THReshold {LOW|HIGH|LH}
  :TRIGger:SLOPe:THReshold?
- Functional description:
  Set the threshold mode of slope trigger.
- Return format:
  Query return{LOW|HIGH|LH}.
- For example:
  :TRIGger:SLOPe:THR HIGH set the threshold mode of slope trigger as HIGH
  :TRIGger:SLOPe:THR? query return HIGH

Alternate Trigger

:TRIGger:ALTernat:TYPE

- Command format:
  :TRIGger:ALTernat:TYPE { EDGE | PULSe | SLOPe}
  :TRIGger:ALTernat:TYPE?
- Functional description:
  Set alternate trigger mode, edge, pulse width and slope.
- Return format:
  Query return{ EDGE | PULSe | SLOPe}.
- For example:
  :TRIGger:ALTernat:TYPE EDGE set alternate trigger mode as edge trigger
  :TRIGger:ALTernat:TYPE? query return EDGE

CURSor Command

This command is to set cursor parameter, measure waveform data on the screen.

:CURSor:MODE

- Command format:
  :CURSor:MODE { TRACK | INDependent }
  :CURSor:MODE?
- Functional description:

Set cursor mode, TRACK or INDependent.

- Return format:

  Query return{ TRACK | INDependent }.

- For example:

  :CURSor:MODE TRACK set cursor mode as TRACK

  :CURSor:MODE? query return TRACK

:CURSor:TYPE

- Command format:

  :CURSor:TYPE {AMPlitude | TIME | CLOSe }

  :CURSor:TYPE?

- Functional description:

  Set cursor mode , AMPlitude, TIME and CLOSe.

- Return format:

  Query return{AMPlitude | TIME | CLOSe }.

- For example:

  :CURSor:TYPE AMP set cursor mode as AMPlitude

  :CURSor:TYPE? query return AMPlitude

:CURSor:SOURce

- Command format:

  :CURSor:SOURce <source>

  :CURSor:SOURce?

- Functional description:

  Set cursor measurement source.

  <source> take value as { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH}.

- Return format:

  Query return{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH}.

- For example:

  :CURSor:SOURce CHAN1 set channel 1 as cursor source

  :CURSor:SOURce? query return CHANnel1

:CURSor:CURA

- Command format:

  :CURSor:CURA <value>

:CURSor:CURA?

- Functional description:

  Set the horizontal or vertical position of A cursor line. It is related with command CURSor:TYPE, amplitude presents the vertical position, time presents the horizontal position. Vertical line range [0,699], horizontal line range [28,227].

- Return format:

  Query return the position of A cursor line.

- For example:

  :CURSor:CURA 50 set manual A cursor line position as 50

  :CURSor:CURA? query return 50

:CURSor:CURB

- Command format:

  :CURSor:CURB <value>

  :CURSor:CURB?

- Functional description:

  Set the horizontal or vertical position of B cursor line. It is related with directive CURSor:TYPE.

- Return format:

  Query return the position of B cursor line.

- For example:

  :CURSor:CURB 50 set manual B cursor line position as 50

  :CURSor:CURB? query return 50

:CURSor:AXValue?

- Command format:

  :CURSor:AXValue?

- Functional description:

  Query X value of A cursor point, unit is select by the amplitude unit of the current channel.

- Return format:

  Query returns X value of A cursor point by scientific notation method.

- For example:

  :CURSor:AXV? query return 2.000000E+02

:CURSor:AYValue?

- Command format:

  :CURSor:AYValue?

- Functional description:

Query Y value of A cursor point, unit is select by the amplitude unit of the current channel.

- Return format:

  Query returns Y value of A cursor point by scientific notation method.

- For example:

  :CURSor:AYV? query return 2.000000E+02

:CURSor:BXValue?

- Command format:

  :CURSor:BXValue?

- Functional description:

  Query X value of B cursor point, unit is select by the amplitude unit of the current channel.

- Return format:

  Query returns X value of B cursor point by scientific notation method.

- For example:

  :CURSor:BXV? query return2.000000E+02

:CURSor:BYValue?

- Command format:

  :CURSor:BYValue?

- Functional description:

  Query Y value of B cursor point, unit is select by the amplitude unit of the current channel.

- Return format:

  Query returns Y value of B cursor point by scientific notation method.

- For example:

  :CURSor:BYV? query return 2.000000E+02

:CURSor:XDELta?

- Command format:

  :CURSor:XDELta?

- Functional description:

  In cursor track measuring, the difference value △X between cursor A point and cursor B point.

- Return format:

  Query return the actual difference value △X between cursor A point and cursor B point by scientific notation method.

- For example:

  :CURSor:XDEL? query return 2.000000E+02

:CURSor:YDELta?

- Command format:
  :CURSor:YDELta?
- Functional description:
  In cursor track measuring, the difference value △Y between cursor A point and cursor B point.Unit is the same as the current chnnel unit.
- Return format:
  Query return the actual difference value △Y between cursor A point and cursor B point by scientific notation method.
- For example:
  :CURSor:YDEL? query return 2.000000E+02

ACQuire Command

This command is to set sampling mode of oscilloscope.

:ACQuire:TYPE

- Command format:
  :ACQuire:TYPE {NORMal | AVERage | PEAKdetect | HRESolution }
  :ACQuire:TYPE?
- Functional description:
  Set sampling mode of oscilloscope.
  NORMal, AVERage, PEAKdetect and HRESolution.
- Return format:
  Query return {NORMal | AVERage | PEAKdetect | ENVelope | HRESolution }.
- For example:
  :ACQ:TYPE AVER set sampling mode as AVERage
  :ACQ:TYPE? query return AVERage

:ACQuire:AVERages:COUNt

- Command format:
  :ACQuire:AVERages:COUNt <count>
  :ACQuire:AVERages:COUNt?
- Functional description:
  Set the average sampling time during average sampling mode, <count> stepped by 2 of Nth power, Nth take value from 2~8192, 1≤N≤30.
- Return format:
  Query return the current sampling time.
- For example:

:ACQ:AVER:COUN 32	set the average sampling time as 32
:ACQ:AVER:COUN?	query return 32

:ACQuire:MODE

- Command format:

  :ACQuire:MODE { EQUivalent | REALtime}

  :ACQuire:MODE?
- Functional description:

  Set samplimg mode, it divided into equivalent sampling and real-time sampling.
- Return format:

  Query return{ EQUivalent | REALtime}.
- For example:

:ACQ:MODE EQU	set sampling mode as EQUivalent
:ACQ:MODE?	query return EQUivalent

:ACQuire:FAST

- Command format:

  :ACQuire:FAST {{1 | ON} | {0 | OFF}}

  :ACQuire:FAST?
- Functional description:

  Set the switch of quick sampling.
- Return format:

  Query return quick sampling status, 0 presents turn on quick sampling, 1 presents turn off quick sampling.
- For example:

:ACQuire:FAST ON	turn on quick sampling
:ACQuire:FAST?	query return 1，it presents quick sampling is turn on

DISPlay Command

This command is to set or query oscilloscope display function or data.

:DISPlay:DATA?

- Command format:

  :DISPlay:DATA?
- Functional description:

  For query the current screen image data.
- Return format:

  Query return BMP image data，return data conform to binary data format IEEE 488.2 #.

- For example:

  :DISPlay:DATA? query return image data

  data format: #800012345+bitmap data

:DISPlay:FORMat

- Command format:

  :DISPlay:FORMat { VECTors | DOTS }

  :DISPlay:FORMat?

- Functional description:

  Set display format of sampling point ,VECTors and DOTS.

- Return format:

  Query return { VECTors | DOTS }.

- For example:

  :DISPlay:FORMat VECT set sampling point as VECTors

  :DISPlay:FORMat? query return VECTors

:DISPlay:GRADing:TIME

- Command format:

  :DISPlay:GRADing:TIME {CLOSe|SHORt|LONG|INFinite}

  :DISPlay:GRADing:TIME?

- Functional description:

  For set or query afterglow time.

- Return format:

  Query return{CLOSe|SHORt|LONG|INFinite}.

- For example:

  :DISPlay:GRADing:TIME SHORt set afterglow time as SHORt

  :DISPlay:GRADing:TIME? query return SHORt

:DISPlay:GRID

- Command format:

  :DISPlay:GRID {FULL|HALF|CORSS|NONE }

  :DISPlay:GRID?

- Functional description:

  For set or query grid type display on screen.

- Return format:

  Query return{FULL|HALF|CORSS|NONE }.

- For example:

  :DISPlay:GRID HALF set grind type as HALF, only display grid not coordinate

  :DISPlay:GRID? query return HALF

:DISPlay:GRID:BRIGhtness

- Command format:

  :DISPlay:GRID:BRIGhtness <count>

  :DISPlay:GRID:BRIGhtness?

- Functional description:

  Set grid brightness, <count> take value from 1~100，the larger the number, the brighter the grid.

- Return format:

  Query return the current grid brightness.

- For example:

  :DISPlay:GRID:BRIGhtness 50 set grid brightness as 50

  :DISPlay:GRID:BRIGhtness? query return 50

:DISPlay:WAVE:BRIGhtness

- Command format:

  :DISPlay:WAVE:BRIGhtness <count>

  :DISPlay:WAVE:BRIGhtness?

- Functional description:

  Set waveform brightness, <count>take value from 1~100，the larger the number, the brighter the waveform.

- Return format:

  Query return the current brightness.

- For example:

  :DISPlay:WAVE:BRIGhtness 50 set waveform brightness as 50

  :DISPlay:WAVE:BRIGhtness? query return 50

:DISPlay:CLEar

- Command format:

  :DISPlay:CLEar

- Functional description:

  For empty waveform displayed on the screen, if oscilloscope is on RUN status, empty displayed waveform then display new waveform.

:DISPlay:TYPE

- Command format:

  :DISPlay:TYPE {XY12 |YT}

  :DISPlay:TYPE?

- Functional description:

Set timebase display mode as XY12（X-Y mode：display amplitude of channel 1 on the horizontal axis, display amplitude of channel 2 on the vertical axis）; YT（Y-T mode：display the relative relationship of vertical voltage and horiziontal time）。

- Return format：

  Query return XY12, YT.

- For example：

  :DISP:TYPE YT　　set timebase format as YT mode

  :DISP:TYPE?　　query return YT

WAVeform Command

This command is to read waveform data on the screen and set parameter.

:WAVeform:MODE

- Command format：

  :WAVeform:MODE　{NORMal | RAW}

  :WAVeform:MODE?

- Functional description：

  NORMal：read the display waveform data on the screen, the waveform data points is fixed points.

  RAW：read the waveform data in memory storage, the waveform data points is related to the storage depth. Notes: Read the data of memory storage is valid only when oscilloscope stops, this command is invalid when in MATH channel.

- Return format：

  query return {NORMal | RAW}.

- For example：

  :WAVeform:MODE RAW　　set the read mode as RAW

  :WAVeform:MODE?　　query return RAW

:WAVeform:FORMat

- Command format：

  :WAVeform:FORMat　{ WORD | BYTE | ASCII }

  :WAVeform:FORMat?

- Functional description：

  The default format of waveform data is AD waveform point

  BYTE：return AD data, one waveform point takes one byte（8 bits）

  WORD：return AD data, one waveform point takes two bytes（16 bits），low 8 bits is valid, high 8 bits is 0

  ASCII：return the actual voltage value of each waveform point by scientific notation method, each voltage value separate by comma mark.

  Conform to binary data IEEE488.2 fomat

For example: #412342.00000E+01,2.20000E+01, 2.30000E+01.....\n

- Return format:

  Query return{ WORD | BYTE | ASCII }.

- For example:

  :WAVeform:FORMat BYTE return format of AD waveform data is single byte mode

  :WAVeform:FORMat? query return BYT

:WAVeform:STARt

- Command format:

  :WAVeform:STARt <start>

  :WAVeform:STARt?

- Functional description:

  Set or query the initial position of waveform data read, < start> integer data type.

  NORMal：1 ~1400

  RAW：1 ~ the maximum current storage depth point

- Return format:

  Query return the initial position.

- For example:

  :WAVeform:STARt 200 set the initial position as 200

  :WAVeform:STARt? query return 200

:WAVeform:STOP

- Command format:

  :WAVeform:STOP <stop>

  :WAVeform:STOP?

- Functional description:

  Set or query the stop position of waveform data read, < start> integer data type.

  NORMal：< stop> range1 ~1400

  RAW：< stop> range: the maximum current storage depth point

- Return format:

  Query return stop position.

- For example:

  :WAVeform:STOP 400 set the stop position as 400

  :WAVeform:STOP? query return 400

:WAVeform:SOURce

- Command format:

  :WAVeform:SOURce {CHANnel<n>| MATH}

:WAVeform:SOURce?

- Functional description:

  Set or query signal source of the current waveform data, if not send this command, it presents query waveform data of the current channel.

- Return format:

  Query return

  { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH }.

- For example:

  :WAVeform:SOURce CHAN1

  set signal source of the current waveform data as channel 1

  :WAVeform:SOURce? query return CHANnel1

:WAVeform:POINts

- Command format:

  :WAVeform:POINts <points>

  :WAVeform:POINts?

- Functional description:

  Set return waveform data, the default value is 0.

- Return format:

  Query return waveform point.

- For example:

  :WAVeform:POINts 120 set return waveform point as 120

  :WAVeform:POINts? query return 120

:WAVeform:DATA?

- Command format:

  :WAVeform:DATA?

- Functional description:

  Read waveform data of the specified data source。

- Return format:

  WAVeform:POINts the specified quantity of waveform data, waveform data source is related with :WAVeform:SOURc, data format is related with WAVeform:FORMat, returns data is conform to binary data IEEE 488.2 # format.

- For example:

  The execute order to obtain the specified data source the waveform data as following:

  - The flow to obtain the waveform data of screen

    :WAVeform:SOURce CHAN1

    Set channel 1 as singal source of query waveform data

    :WAVeform:MODE NORMal

    Set to read the display waveform data

:WAVeform:FORMat BYTE

The return format of waveform data is AD single byte mode

:WAVeform:DATA?

Obtain the waveform data

◆ The flow to obtain the internal waveform data, this flow is only valid when the oscilloscope stops

:WAVeform:SOURce CHAN1

Set channel 1 as singal source of query waveform data

Set to read the internal waveform data

:WAVeform:FORMat BYTE

The return format of waveform data is AD single byte mode

:WAVeform:POINts 5000

Internal waveform point are 5000

:WAVeform:DATA?

Obtain an area of internal waveform data

Description: when read internal data in batch, returns data is only an area of internal data, waveform data is continuous between in two adjacent area and each area data is conform to binary data IEEE 488.2 # format.

:WAVeform:PREamble?

- Command format:

  :WAVeform:PREamble?

- Functional description:

  Query return the current system waveform setup parameter.

- Return format:

  Query return use comma mark to separate.

  Return data format: Format,Type,Points,Count,Xinc,Xor,Xref,Yinc,Yor,Yref.

  Format：BYTE (0), WORD (1), ASCII(2).

  Type：NORMAL(0), PEAK(1), AVER(2), ENVelope(3), HRESolution(4).

  Points：return the waveform data points.

  Count：in average sampling mode as average time, other mode as 1.

  Xinc：the time difference between in two point of X direction of waveform data source.

  Xor：the relative time of trigger point.

  Xref：X reference

  Yinc：unit voltage of Y direction

  Yor：the relative position of Y direction and YREF zero position

  Yref：the reference value of Y direction, the center point of screen

- For example:

:WAVeform:PREamble?

return 1,0,0,1,8.000e-009,-6.000e-006,0,4.000e-002,0.000e000,100

:WAVeform:XINCrement?

- Command format:

  :WAVeform:XINCrement?

- Functional description:

  Query the time interval between two adjacent points in X direction of the selected source.

  The returns value is related to data read mode:

  NORMal mode，XINCrement=TimeScale/ waveform points in the horizontal direction of time base

  RAW mode，XINCrement=1/SampleRate

- Return format:

  Query return time base count，unit is s.

- For example:

  :WAV:XINC? query return 3.000e-003

:WAVeform:XORigin?

- Command format:

  :WAVeform:XORigin?

- Functional description:

  Query the selected channel initial time of waveform data in the X direction, trigger point time is zero, before trigger point is negative number.

  Return value is related with the current data read mode:

  NORMal mode，return initial time of waveform data display on the screen.

  RAW mode，return initial time of waveform data storage in memory.

- Return format:

  Query return time value，unit is s.

- For example:

  :WAV:XOR? query return 3.000e-002

:WAVeform:XREFerence?

- Command format:

  :WAVeform:XREFerence?

- Functional description:

  Query the time reference of waveform points in X direction of the selected channel source, always be zero.

- Return format:

  Query time reference，query return 0.

- For example:

:WAV:XREF? query return 0

:WAVeform:YINCrement?

- Command format:

  :WAVeform:YINCrement?

- Functional description:

  Query unit voltage value of the current channel source in Y direction, unit is accord with amplitude value.

  Return value is related with the current data read mode:

  NORMal mode，YINCrement = VerticalScale/waveform point of voltage base in vertical position

  RAW mode，YINCrement is related with memory waveform VerticalScale and the current selected VerticalScale.

- Return format:

  Query return unit voltage value of Y direction.

- For example:

  :WAV:YINC? query return 2.000e000

:WAVeform:YORigin?

- Command format:

  :WAVeform:YORigin?

- Functional description:

  Query the vertical shift in the Y direction of the current selected channel source relative to the vertical reference position.

  Return value is related with the current data read mode:

  NORMal mode， YORigin = VerticalOffset/YINCrement

  RAW mode， YORigin is related with memory waveform VerticalScale and the current selected VerticalScale.

- Return format:

  Query return vertical offset, integer type.

- For example:

  :WAV:YOR? query return 0

:WAVeform:YREFerence?

- Command format:

  :WAVeform:YREFerence?

- Functional description:

  Query the reference position in the Y direction of the current selected channel source, ADC of channel zero level channel.

Return value is related with the current data read mode:

NORMal mode， YREFerence is fixed as 128（the bottom of screen as 0, top as 255）

RAW mode，YREFerence is related with memory waveform VerticalScale and the current selected VerticalScale.

- Return format：

  Query return referece position, integer type.

- For example：

  :WAV:YREF? query return 100

FILE Command

This command is for reference waveform and storage function.

:FILE:LOAD

- Command format：

  :FILE:LOAD <filename>[,<source>][,<disk>]

- Functional description：

  For loading waveform to the related channel or data setting.

  <filename> file name must be character string data with double quotation mark, for example “test.bsv”

  - The file name of *.bsv or *.csv presents load waveform data of the file into the reference channel.
  - The file name of *.set presents load the setting data of the file into oscilloscope.

  <source > presents reference channel{REFA | REFB | REFC | REFD}，**optional parameter only valid when load waveform data.**

  - REFA presents reference channel A
  - REFB presents reference channel B
  - REFC presents reference channel C

  REFD presents reference channel D

  <disk> presents memory medium { FLASH | UDISK **}，optional parameter，ignore presents internal data of FLASH.**

  - FLASH presents internal data
  - UDISK presents U disk data

- **For example：**

  FILE:LOAD "test.bsv",REFA,UISK

  Load test.bsv waveform data into reference channel A from U disk

  FILE:LOAD "system-set-up01.set"

  Load No .1 configuration data into a oscilloscope from internal medium .

Notes：

- The file name of internal storage setting must be"system-set-up01.set"~ "system-set-up255.set",

maximum limit 255 files.

- The file name of internal storage bsv must be"wave01.bsv"~" wave255.bsv", maximum limit 255 files.

:FILE:SAVE

➢ Command format:

:FILE:SAVE <filename>[,<source>][,<disk>]

➢ Functional description:

For save channel waveform or setting data into file.

<filename> file name must be character string data with double quotation mark, for example "test.bsv"

- The file name of *.bsv or *.csv presents save channel waveform with the suffix name format into file.
- The file name of *.set presents save the setting data into file.

<source > presents physical channel {CHANnel1 | CHANnel2| CHANnel3| CHANnel4}**, optional parameter only valid when load waveform data.**

- CHANnel1 presents Channel 1
- CHANnel2 presents Channel 2
- CHANnel3 presents Channel 3
- CHANnel4 presents Channel 4

<disk> presents memory medium { FLASH | UDISK **}, optional parameter, ignore presents internal data of FLASH.**

- FLASH presents internal data
- UDISK presents U disk data

➢ **For example:**

FILE:SAVE "test.bsv",CHANnel1,UDISK

Save waveform data of channel 1 as test.bsv format into U disk.

FILE:SAVE "system-set-up01.set"

The configuration data of oscilloscope save as No.1 position of internal medium.

FILE:SAVE "wave01.bsv",CHANnel1,FLASH

Save waveform data of channel 1 into internal medium.

FILE:SAVE "wave01.bsv",CHANnel1

Save waveform data of channel 1 into internal medium.

FILE:SAVE "system-set-up01.set",FLASH

The configuration data of oscilloscope save as internal medium.

FILE:SAVE "system-set-up01.set"

The configuration data of oscilloscope save as No.1 position of internal medium.

Notes：

- The file name of internal storage setting must be"system-set-up01.set"~ "system-set-up255.set", maximum limit 255 files.
- The file name of internal storage bsv must be"wave01.bsv"~" wave255.bsv", maximum limit 255 files.

Appendix 1：Key Table

Key	Functional Description	LED
CH1	Channel 1 switch	√
CH2	Channel 2 switch	√
MATH	Mathematical operation and menu	√
AUTO	Set control parameter automatically to display waveform	
RS	Control oscilloscope operating status, send this command continuously, oscilloscope will step through stop and running status	√
TMENu	Trigger menu	
DEFault	Recovery default setting	
HELP	Help system	
HMENu	Horizontal system meue	
DISPlay	Display menu	
F1	Select the first item of the current menu	
F2	Select the second item of the current menu	
F3	Select the third item of the current menu	
F4	Select the fourth item of the current menu	
F5	Select the fifth item of the current menu	
MENu	Menu display switch	
PSCReen	One key to print or one key save screen image	
MEASure	Measurement	
CURSor	Cursor measurement and menu	
ACQuire	Sampling menu	
STORage	Storage menu	
UTILity	Subsystem menu	
FKNob	Multifunction knob	
FKNLeft	Multifunction left knob	
FKNRight	Multifunction right knob	
VPKNob	Vertical position knob	
VPKNLeft	Vertical position left knob	
VPKNRight	Vertical position right knob	

HPKNob	Horizontal position knob	
HPKNLeft	Horizontal position left knob	
HPKNRight	Horizontal position right knob	
TPKNob	Trigger position knob	
TPKNLeft	Trigger position left knob	
TPKNRight	Trigger position right knob	
VBKNob	Voltage reference knob	
VBKNLeft	Voltage reference left knob	
VBKNRight	Voltage reference right knob	
TBKNob	Time reference knob	
TBKNLeft	Time reference left knob	
TBKNRight	Time reference right knob	
SETZERO	Zero setting	

Appendix 2: Binary Data Format IEEE 488.2

DATA is data flow, other is ASCII character, as shown in the table below: <#812345678 + DATA + \n>

Start mark (1Byte)	**Length of bit wide (1Byte)**	**Data total length (bit wide Byte)**	**DATA (n Byte)**	**End mark (1Byte)**
#	x	x x x x x x x x	……	\n