RIGOL

Programming Guide

DS1000B Series Digital Oscilloscope

DS1204B/DS1104B/DS1064B

Dec. 2009

RIGOL Technologies, Inc.

Copyright

Trademark Information

RIGOL is registered trademark of **RIGOL** Technologies, Inc.

Notices

RIGOL products are protected by patent law in and outside of P.R.C..

RIGOL Technologies, Inc. reserves the right to modify or change parts of or all the specifications and pricing policies at company's sole decision.

Information in this publication replaces all previously corresponding material.

RIGOL shall not be liable for losses caused by either incidental or consequential in connection with the furnishing, use or performance of this manual as well as any information contained. Any part of this document is forbidden to copy or photocopy or rearrange without prior written approval of **RIGOL**.

Product Certification

RIGOL guarantees this product conforms to the standards of national and industrial. Meanwhile, the related standards conform to other ISO will get further. At present, DS1000B series has passed CE and LXI certification.

Content

Chapter 1 Programming Introduction

This chapter provides guidance to the remote control programming of the DS1000B series digital oscilloscopes and introduction of the commands.

This chapter includes:

- Communication Interface
- Command Introduction
  Command Syntax
  Symbol Description
  *Command Input*Command Input
  Parameter Type

Communication Interface

Computers can communicate with the oscilloscope by sending and receiving messages over USB or LAN interface. Commands in the form of ASCII character strings are embedded in your computer to make control easier.

Operations that you can do with a computer and the oscilloscope include:

- Set up the oscilloscope;
- Relational measurements;
- Acquire data (waveforms or measurement data) from the oscilloscope.

Equipment Connection:

- **USB:** Use an USB data cable and connect the USB Device port on the rear panel of the oscilloscope to PC.
- **LAN:** Use a network cable and connet the oscilloscope to LAN.

Command Introduction

Command Syntax

The commands system of DS1000B series oscilloscope is a multistage tree structure, and each of sub-system is consists of a "Root" keyword and multilayered keywords. The commands are always begin with a "**:**"(except for IEEE commands) and the keywords are also separated by "**:**"; optional parameters are permitted to follow the keywords; "**?**" appeared following a command line denotes to query this function; besides, "**space**" is used to divide command and parameter.

For example:
:TRIGger:EDGE:SLOPe {POSitive|NEGative|ALTernation}
:TRIGger:EDGE:SLOPe?

TRIGger is the root keyword of the command, **EDGE** and **SLOPe** is separately its second and third keyword, all of them are separated by "**:**". Connects enclosed in the "**{}**" denote the parameters permitted to be set by user; "**?**" denotes to query; the command **:TRIGger:EDGE:SLOPe** is set apart from parameter using "**space**". "**,**" is used to compart the parameters existed in some commands, for example:
:TRIGger:PATTern:PATTern <value>,<mask>,<ext source>[,<edge source>,<edge>]

Symbol Description

The following symbols are not "real" parts of the commands, but they are usually used to assist to explain the parameters containd in a command line.

1. Braces { }

The parameters or contents enclosed in "**{}**" must be selected, and only one parameter or content could be selected once. All the options are separated by "**|**".

For example:

{{1|ON}|{0|OFF}} indicateds that 1, ON, 0 or OFF can be selected at a time.

2. Square brackets [ ]

Some keywords or contents are enclosed by square bracket "**[ ]**", which indicates that those parameters are optional and will be execute no matter whether been omitted or not.

For example:

:TIMebase[:MAIN]:OFFSet <offset>

[:MAIN] can be omitted.

3. Triangle Brackets < >

Parameter enclosed in "**< >**" should be replaced by a effective value.

For example:

:DISPlay:BRIGhtness <ncount>

replaced by an effective value:

:DISPlay:BRIGhtness 80

Command Input

All the comands are not sensitive to both capital letter and lowercase, so you can use any kind of them. But if use abbreviation, the capital letters specified in commands must be written completely.
For example:
:TRIGger:ALTernation:SOURce
also can be:
:TRIG:ALT:SOUR or **:trig:alt:sour**

Parameter Type

The commands contains 5 kinds of parameters, different parameters has different setting methods.

1. Boolean

The parameter should be "OFF", "ON", "0" or "1". For example:
:DISPlay:PERSist {{1|ON}|{0|OFF}}
"ON" and "1" denotes trun on (enable) the function, "OFF" and "0" denotes turn off (disable) the fucntion.

2. Consecutive Integer

The parameter should be a consecutive integer. For example:
:DISPlay:BRIGhtness <ncount>
<ncount> could be the integer between 0 and 100 (including 0 and 100).

3. Consecutive Real Number

The parameters can be any value only in effective range precision permitting. For example:
:TRIGger:SENSitivity <count>
<count> could be any value between 0.1 and 1 (including 0.1 and 1).

4. Discrete

The parameters can only be the cited value. For example:
:ACQuire:AVERages <count>
<count> could only be 2, 4, 8, 16, 32, 64, 128, 256.

5. ASCII Character String

The parameter should be composed of ASCII character string. For example:
:TRIGger:MODE <mod>
<mod> could be EDGE, PULSe, VIDEO, PATTern or ALTernation.

Chapter 2 Command Systems

In this chapter, we will introduce every comand in the DS1000B command systems. The introduction inludes command format, function description, query/Returned Format and some other notices that should pay attention to during using the commands.

DS1000B series support the following command subsystems:

- General Command
- SYSTem Command
- ACQuire Command
- DISPlayDISPlay Command
- TIMebase Command
  TRIGger Command
  Trigger Control
  EDGE Trigger
  PULSe Trigger
  VIDEO Trigger
  PATTern Trigger
  ALTernation Trigger
- MATH Command
- CHANnel Command
- MEASure Command
- WAVeform Command
- KEY Command
- SAVe/RECall Command
- MASK Command
- CURSor Command
- Other Command

General Commands

IEEE Standards have defined some general commands which are applied to query basic information of the instrument or perform elementary operations. These commands always have 3 characters and with a "*" marker.

DS1000B series support the following General Commands:

- *IDN?
- *RST
- *LRN?
- *OPC?

We will give detailed introductions for each command in the following parts.

1. *IDN?

Command Format:
*IDN?

Function:
The command queries the manufacturer, the oscilloscope model, the product serial and the software version.

Returned Format:
manufacturer, <model>, <serial>, <version>.

Example:
Rigol Technologies, DS1204B, DS10000000, 00.02.04.

2. *RST

Command Format:
*RST

Function:
The command resets the system.

3. *LRN?

Command Format:
*LRN?

Function:
The command queries the system settings.

Returned Format:
The query returns the data of system settings in the form of a self-defined character string which could be downloaded to do the same settings in the future.

4. *OPC?

Command Format:
*OPC?

Function:
The command queries whether the command operation has been completed.

Returned Format:
The query returns 0 or 1. 1 means operation has been completed, 0 means not.

SYSTem Commands

SYSTem Commands are used for the basic operations of an oscilloscope: RUN/STOP control, operation of the error queue and system setup data.

SYSTem Commands include:

- :RUN
- :STOP
- :AUTO
- :SYSTem:ERRor
- :SYSTem:SETup

We will give detailed introductions for each command in the following parts.

1. :RUN

Command Format:
:RUN

Function:
Execute this command, the oscilloscope will start waveform sampling working. To stop working, execute **:STOP** command again.

2. :STOP

Command Format:
:STOP

Function:
Execute this command, the oscilloscope will stop waveform sampling working. To restart working, execute **:RUN** command again.

3. :AUTO

Command Format:
:AUTO

Function:
The command makes the oscilloscope tests all input waveforms and set the waveforms automatically to get the optimum conditions to display.

4. :SYSTem:ERRor

Command Format:
:SYSTem:ERRor
:SYSTem:ERRor?

Function:
The command clears the queue of error information.

Returned Format:

The query returns the last error, such as "Undefined header". If there is no error, return "0, No error".

For details about system error codes, please refer to page 2-8:

System Error Code

5. :SYSTem:SETup

Command Format:

:SYSTem:SETup <setup_data>

:SYSTem:SETup?

Function:

The command downloads the system setup data. <setup data> is a binary data that meets IEEE 488.2 # format.

Returned Format:

The query returns the value of system setup data.

System Error Code

Up to 10 errors can be recorded in the system error queue. If not enough, the system will adopt FIFO manner to cover the original error record.

The SYST:ERR? Command is used to read the first error code in the form of "error code, error description", so as to reduce the error number of error queue. For instance, if no error appears, the system will return: 0, No error.

Besides, the :SYST:ERR Command is able to clear error queue.

Error Code	Mnemonic Symbol	Error Description
0	ERR_NONE,	No error
1	ERR_SAME_SETTING,	Same setting
2	ERR_INVALID_INPUT,	Invalid input
3	ERR_LIMIT_SETTING,	Setting limit
4	ERR_CH_OFFSET_LIMIT,	Channel offset limit
5	ERR_CH_SCALE_LIMIT,	Channel scale limit
6	ERR_CH_PROBE_LIMIT,	Channel probe limit
7	ERR_CH_FILTER_LIMIT,	Channel filter limit
8	ERR_TIME_OFFSET_LIMIT,	Timebase offset limit
9	ERR_TIME_SCALE_LIMIT,	Timebase scale limit
10	ERR_TIME_DELAYED_OFFSET_LIMIT,	Timebase of timedelay offset limit
11	ERR_TIME_DELAYED_SCALE_LIMIT,	Timebase of timedelay scale limit
12	ERR_TRIG_LEVEL_LIMIT,	Trigger level limit
13	ERR_MATH_VERT_OFFSET_LIMIT,	Math vertical offset limit
14	ERR_MATH_VERT_SCALE_LIMIT,	Math vertical scale limit
15	ERR_FFT_VERT_SCALE_LIMIT,	FFT vertical offset limit
16	ERR_FFT_VERT_OFFSET_LIMIT,	FFT vertical offset limit
17	ERR_FFT_HORIZ_SCALE_LIMIT,	FFT horizontal scale limit
18	ERR_FFT_HORIZ_OFFSET_LIMIT,	FFT horizontal offset limit
19	ERR_CUR_A_X_LIMIT,	CursorA X-Axial limit
20	ERR_CUR_B_X_LIMIT,	CursorB X-Axial limit
21	ERR_CUR_A_Y_LIMIT,	CursorA Y-Axial limit
22	ERR_CUR_B_Y_LIMIT,	CursorB Y-Axial limit
23	ERR_HOLDOFF_TIME_LIMIT,	Holdoff time limit
24	ERR_INTENSITY_LIMIT,	Intensity limit
25	ERR_PULSE_WIDTH_LIMIT,	Pulse width limit

26	ERR_VIDEO_LINE_LIMIT,	Video line limit
27	ERR_REC_INTERVAL_LIMIT,	Record interval limit
28	ERR_REC_END_FRAME_LIMIT,	Record end frame limit
29	ERR_PLAY_INTERVAL_LIMIT,	Play interval limit
30	ERR_PLAY_START_FRAME_LIMIT,	Play start frame limit
31	ERR_PLAY_CUR_FRAME_LIMIT,	Play current frame limit
32	ERR_PLAY_END_FRAME_LIMIT,	Play end frame limit
33	ERR_STOARAGE_START_FRAME_LIMIT,	Storage start frame limit
34	ERR_STOARGE_END_FRAME_LIMIT,	Storage end frame limit
35	ERR_REF_VERT_OFFSET_LIMIT,	Ref vertical offset limit
36	ERR_REF_VERT_SCALE_LIMIT,	Ref vertical scale limit
37	ERR_PF_MASK_LIMIT,	Passfail mask limit
38	ERR_SAMPLING_RATE_LIMIT,	Sampling rate limit
39	ERR_GRID_INTENSITY_LIMIT,	Grid intensity limit
40	ERR_TRIG_SENSITIVITY_LIMIT,	Trigger sensitivity limit
41	ERR_TRIG_SLOPE_TIME_LIMIT,	Trigger slop time limit
42	ERR_MEM_DEPTH_LIMIT,	Memory depth limit
43	ERR_FUNCTION_NOT_AVAILABLE,	Function not available
44	ERR_LOCATION_EMPTY,	Location empty
45	ERR_MEAS_ALREADY_SELECTED,	Measure already selected
46	ERR_NO_SIGNAL_FOUND,	No signal found
47	ERR_WAVEFORM_RECORD_FINISHED,	Waveform record finished
48	ERR_FILE_UTILITY_FAIL,	File utility fail
49	ERR_CHANNEL_INVALID,	Channel invalid
50	ERR_AUTO_KEY_LIMITED,	Auto key limited
51	ERR_NOT_ENOUGH_MEMORY,	Not enough memory
52	ERR_WAVE_SAVE_FAILED,	Waveform save failed
53	ERR_WAVE_LOAD_FAILED,	Waveform load failed
54	ERR_FILE_IS_COVERED,	File is covered
55	ERR_FILTER_IS_CLOSED,	Filter is closed
56	ERR_WAVE_TYPE_NONE,	No signal detected
57	ERR_WAVE_TYPE_DC,	DC signal detected
58	ERR_WAVE_TYPE_SINE,	Sine signal detected
59	ERR_WAVE_TYPE_RAMP,	Triangle signal detected
60	ERR_WAVE_TYPE_RECT,	Square signal detected
61	ERR_WAVE_TYPE_UNKNOWN,	Unknown signal detected
62	CMD_ERR,	Error header
63	CMD_NOT_PARSE,	Undefined header

64	ERR_PF_OUTPUT,	PassFail Out
65	ERR_MISSING_HW,	Missing Hardware
66	ERR_OUT_OF_RANGE	Out of range
67	ERR_CANNOT_EXECURE	Can’t execute

ACQuire Commands

ACQuire Commands are used to set the acquisition mode for oscilloscope.

ACQuire Commands include:

- :ACQuire:TYPE
- :ACQuire:MODE
- :ACQuire:AVERages
- :ACQuire:SRATe?

We will give detailed introductions for each command in the following parts.

1. :ACQuire:TYPE

Command Format:
:ACQuire:TYPE <type>
:ACQuire:TYPE?

Function:
The command sets the acquisition type. The <type>may be NORMal(common sample), AVERage(average sample) or PEAKdetect(peak detection).

Returned Format:
The query returns Normal or AVERAGE, PEAKDETECT.

Example:

:ACQ:TYPE AVERage	Set the acquisition type as average acquisition.
:ACQ:TYPE?	Return AVERAGE.

2. :ACQuire:MODE

Command Format:
:ACQuire:MODE <mode>
:ACQuire:MODE?

Function:
The command sets the acquisition mode. The <mode> may be RTIMe (real time sample) or ETIMe (equal time sample).

Returned Format:
The query returns RTIME or ETIME.

Example:

:ACQ:MODE ETIM	Set the acquisition mode as equal time acquisition.
:ACQ:MODE?	Return ETIME.

3. :ACQuire:AVERages

Command Format:

:ACQuire:AVERages <count>
:ACQuire:AVERages?

Function:
The command sets the average acquisition time. The <count> range is 2~256, and the count increases by the power operation of 2.

Returned Format:
The query returns 2 or 4, 8, 16, 32, 64, 128, 256.

Example:
:ACQ:AVER 16 Set the average acquisition time as 16.
:ACQ:AVER? Return 16.

4. :ACQuire:SRATe?

Command Format:
:ACQuire:SRATe? [{<CHANnel<n>}]

Function:
To query sample rate of CHANnel <n>, <n> may be 1, 2, 3, 4.

Returned Format:
The query returns 5.000e005, the unit is Sa/s.

DISPlay Commands

DISPlay Commands are used to set the display system.

DISPlay Commands include:

- :DISPlay:TYPE
- :DISPlay:GRID
- :DISPlay:PERSist
- :DISPlay:MNUDisplay
- :DISPlay:MNUStatus
- :DISPlay:SCReen
- :DISPlay:CLEar
- :DISPlay:BRIGhtness
- :DISPlay:INTensity
- :DISPlay:DATA?

We will give detailed introductions for each command in the following parts.

1. :DISPlay:TYPE

Command Format:
:DISPlay:TYPE <type>
:DISPlay:TYPE?

Function:
The command sets the display type of acquisition points. The <type> may be VECTors (acquisition points are connected by lines) or DOTS (acquisition points are displayed by dots).

Returned Format:
The query returns VECTORS or DOTS.

Example:
:DISP:TYPE VECT Set the display type as vectors.
:DISP:TYPE? Return VECTORS.

2. :DISPlay:GRID

Command Format:
:DISPlay:GRID <grid>
:DISPlay:GRID?

Function:
The command sets the display type of screen grid. The <grid> may be FULL (grid and coordinate are shown), HALF (grid is not shown) or NONE (grid and coordinate are not shown).

Returned Format:
FULL or HALF, NONE.

Example:
:DISP:GRID FULL Make grid and coordinate shown.
:DISP:GRID? Return FULL.

3. :DISPlay:PERSist

Command Format:
:DISPlay:PERSist {{1|ON}|{0|OFF}}
:DISPlay:PERSist?

Function:
The command sets waveform persist function ON (The waveform is shown until waveform persist function is off or relevant settings are changed.) or OFF (The waveform is updated as high refresh rate).

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:DISP:PERS ON	Set waveform persist function on.
:DISP:PERS?	Return 1.

4. :DISPlay:MNUDisplay

Command Format:
:DISPlay:MNUDisplay <time>
:DISPlay:MNUDisplay?

Function:
The command sets the display time of menu. The menu will hide after the display time. The <time> may be 1s, 2s, 5s, 10s, 20s or INFinite (display all the time).

Returned Format:
The query returns 1s or 2s, 5s, 10s, 20s, Infinite.

Example:

:DISP:MNUD 10s	Set the display time as 10s.
:DISP:MNUD?	Return 10s.

5. :DISPlay:MNUStatus

Command Format:

:DISPlay:MNUStatus {{1|ON}|{0|OFF}}
:DISPlay:MNUStatus?

Function:
The command sets menu display function ON (Performing menu operation) or OFF (viewing the waveform).

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:DISP:MNUS ON	Set menu display function on.
:DISP:MNUS?	Return 1.

6. :DISPlay:SCReen

Command Format:
:DISPlay:SCReen <scr>
:DISPlay:SCReen?

Function:
The command sets the display mode of screen. The <scr>may be NORMal (normal display mode) or INVerted (inverted display mode).

Returned Format:
The query returns NORMAL or INVERTED.

Example:

:DISP:SCR NORM	Set the screen as normal display mode.
:DISP:SCR?	Return NORMAL.

7. :DISPlay:CLEar

Command Format:
:DISPlay:CLEar

Function:

The command clears the out of date waveforms on the screen during waveform persist.

8. :DISPlay:BRIGhtness

Command Format:
:DISPlay:BRIGhtness <count>
:DISPlay:BRIGhtness?

Function:
The command sets the brightness of grid. The <count> range is 0~100, and the bigger the count is, the brighter the grid becomes.

Returned Format:
The query returns 0 or 1, 2 ……100.

Example:
:DISP:BRIG 10 Set the grid brightness as 10.
:DISP:BRIG? Return 10.

9. :DISPlay:INTensity

Command Format:
:DISPlay:INTensity <count>
:DISPlay:INTensity?

Function:
The command sets the brightness of the waveform. The <count> range is 0~100, and the bigger the count is, the brighter the waveform becomes.

Returned Format:
The query returns 0 or 1, 2 ……100.

Example:
:DISP:INT 12 Set the waveform brightness as 12.
:DISP:IN? Return 12.

10. :DISPlay:DATA?

Command Format:

:DISPlay:DATA?

Function:

The command queries image data on the current screen. The data format accords with IEEE 488.2 standard. The data structure is: #800078788+the data of 8 bit bitmap.

TIMebase Commands

TIMebase Commands are used to set horizontal scale and horizontal offset. Changing horizontal scale makes the waveform enlarge or shrink; and changing horizontal position will lead the waveform offset relative to center screen.

TIMebase Commands include:

- :TIMebase:MODE
- :TIMebase[:MAIN]:OFFSet
- :TIMebase:DELayed:OFFSet
- :TIMebase[:MAIN]:SCALe
- :TIMebase:DELayed:SCALe
- :TIMebase:FORMat

We will give detailed introductions for each command in the following parts.

1. :TIMebase:MODE

Command Format:
:TIMebase:MODE <mode>
:TIMebase:MODE?

Function:
The command sets the scan mode of horizontal timebase as MAIN (main time base) or DELayed (zoomed scan time base).

Returned Format:
The query returns MAIN or DELAYED.

Example:
:TIM:MODE MAIN Set the scan mode as main time base.
:TIM:MODE? Return MAIN.

2. :TIMebase[:MAIN]:OFFSet

Command Format:
:TIMebase[:MAIN]:OFFSet <offset>
:TIMebase:MAIN:OFFSet?

Function:
The command sets the timebase offset of main mode, that is the offset of the waveform position relative to center screen.
In NORMAL mode, <offset>: 1s ~ memory capacitance;
In STOP mode, <offset>: -500s ~ +500s;
In SCAN mode, <offset>:
(-6*MainScale + 6*DelayedScale) ~ (6*MainScale – 6*DelayedScale)
Scale is the current horizontal scale, and the unit is s/div.

Returned Format:
The query returns the value of timebase offset, and the unit is s.

Example:
:TIM:MODE MAIN Set the scan mode as main.
:TIM:OFFS 1 Set the timebase offset as 1s.

:TIM:OFFS? Return 1.000e000.

3. :TIMebase:DELayed:OFFSet

Command Format:

:TIMebase:DELayed:OFFSet <offset>

:TIMebase:DELayed:OFFSet?

Function:

The command sets the timebase offset of delayed scan, that is the offset of the waveform position relative to center screen.

In NORMAL mode, <offset>: 1s ~ memory capacitance;

In STOP mode, <offset>: -500s ~ +500s;

In SCAN mode, <offset>:

(-6*MainScale + 6*DelayedScale①) ~(6*MainScale – 6*DelayedScale)

Scale is the current horizontal scale, and the unit is s/div.

NOTE①: In Delayed mode, only Delayed offset can be changed but for Main offset.

Thereinto:

the time range: +/- 6*MainScale;

The length of time is 12*DelayedScale;

So Delayed Offset range is:

(-6*MainScale + 6*DelayedScale) ~ (6*MainScale-6*DelayedScale).

For example: When Main 5ms, Delay 2ms, EMS memory time is +/- 6*5=30ms,

Delay time is 6*2 = 12ms. Delay Offset range: (-30+6) ~ (30-6) ms.

Returned Format:

The query returns the value of offset, and the unit is s.

Example:

:TIM:MODE DEL Set the scan mode as delayed scan.

:TIM:DEL:OFFS 1 Set the timebase offset as 1s.

:TIM:DEL:OFFS? Return 1.000e000.

4. :TIMebase[:MAIN]:SCALe

Command Format:

:TIMebase[:MAIN]:SCALe <scale_val>
:TIMebase[:MAIN]:SCALe?

Function:

The command sets the timebase scale of main mode, and the unit is s/div.

- In NORMAL mode, different types of instruments have different sweep ranges:
  DS1204B , <scale_val> range: 1ns/div~50s/div.
  DS1104B, <scale_val>range: 2ns/div~50s/div.
  DS1064B, <scale_val>range: 5ns/div~50s/div.
- In SCAN mode, <scale_val>range: 50ms ~ 50s.

Returned Format:

The query returns the value of timebase scale, and the unit is s.

Example:

:TIM:MODE MAIN	Set the scan mode as main.
:TIM:SCAL 2	Set the timebase scale as 2s.
:TIM:SCAL?	Return 2.000e000.

5. :TIMebase:DELayed:SCALe

Command Format:

:TIMebase:DELayed:SCALe <scale_val>
:TIMebase:DELayed:SCALe?

Function:

The command sets the timebase scale of delayed scan, and the unit is s/div. When the "Delayed" is "ON", for view waveform details, the waveform may be amplified under the width of window vary with the delayed timebase scale.

- In NORMAL mode, different types of instruments have different sweep ranges:
  DS1204B, <scale_val> range: 1ns/div~50s/div.
  DS1104B, <scale_val>range: 2ns/div~50s/div.
  DS1064B, <scale_val>range: 5ns/div~50s/div.
- In SCAN mode, <scale_val>range: 50ms ~ 50s.

Returned Format:

The query returns the value of timebase scale, and the unit is s.

Example:

:TIM:MODE DEL	Set the scan mode as delayed scan.
:TIM:DEL:SCAL 2	Set the timebase scale as 2s.
:TIM:DEL:SCAL?	Return 2.000e000.

6. :TIMebase:FORMat

Command Format:

:TIMebase:FORMat <vlaue>

:TIMebase:FORMat?

Function:

The command sets the timebase format as XY (the amplitude of channel 1 is shown in X axis, and the amplitude of channel 2 is shown in Y axis), YT (the relationship between the voltage and the time is shown) or ROLL (the acquisition points on screen are updated from left to right).

Returned Format:

The query returns X-Y or Y-T, ROLL.

Example:

:TIM:FORM YT	Set the timebase format as Y-T.
:TIM:FORM?	Return Y-T.

TRIGger Commands

Trigger system makes the meaningful waveform shown steadily. Trigger determines when the oscilloscope starts to acquire data and to display a waveform. When trigger is set up properly, it can convert unstable displays into meaningful waveforms.

When the oscilloscope starts to acquire data, firstly enough data are needed to be collected so as to shape into a waveform on the left of the trigger point. The oscilloscope continues acquiring data while waiting for the trigger condition to occur. After it detects a trigger, the oscilloscope continues to acquire enough data so that it can display the waveform on the right of the trigger point.

Trigger Mode includes: Edge, Pulse, Video, Pattern and Alternation trigger.

TRIGger Commands include:

Trigger Control Command

- :TRIGger:MODE
- :TRIGger<mode>:SOURce
- :TRIGger<mode>:LEVel
- :TRIGger<mode>:SWEep
- :TRIGger:SENSitivity
- :TRIGger:COUPling
- :TRIGger:HFREject
- :TRIGger:HOLDoff
- :TRIGger:STATus?
- :Trig%50
- :FORCetrig
- :SINGLE

EDGE Command

- :TRIGger:EDGE:SLOPe

PULSe Command

- :TRIGger:PULSe:MODE
- :TRIGger:PULSe:WIDTh

VIDEO Command

- :TRIGger:VIDEO:MODE
- :TRIGger:VIDEO:POLarity
- :TRIGger:VIDEO:STANdard
- :TRIGger:VIDEO:LINE

PATTern Command

- :TRIGger:PATTern:PATTern

ALTernation Command

- :TRIGger:ALTernation:SOURce
- :TRIGger:ALTernation:CURRentSOURce
- :TRIGger:ALTernation:TYPE
- :TRIGger:ALTernation:TimeSCALe
- :TRIGger:ALTernation:TimeOFFSet
- :TRIGger:ALTernation:LEVel
- :TRIGger:ALTernation:EDGE:SLOPe
- :TRIGger:ALTernation:PULSe:MODE
- :TRIGger:ALTernation:PULSe:TIME
- :TRIGger:ALTernation:VIDEO:POLarity
- :TRIGger:ALTernation:VIDEO:STANdard
- :TRIGger:ALTernation:VIDEO:MODE
- :TRIGger:ALTernation:VIDEO:LINE
- :TRIGger:ALTernation:COUPling
- :TRIGger:ALTernation:HFREject
- :TRIGger:ALTernation:HOLDoff
- :TRIGger:ALTernation:SENSitivity

We will give detailed introductions for each command in the following parts.

Trigger Control

1. :TRIGger:MODE

Command Format:
:TRIGger:MODE <mode>
:TRIGger:MODE?

Function:
The command sets the trigger mode as EDGE, PULSe, VIDEO, ALTernation or PATTern trigger.

Returned Format:
The query returns EDGE or PULSE, VIDEO, ALTERNATION, PATTERN.

Example:

:TRIG:MODE EDGE	Set the trigger mode as edge trigger.
:TRIG:MODE?	Return EDGE.

2. :TRIGger<mode>:SOURce

Command Format:
:TRIGger<mode>:SOURce <source>
:TRIGger<mode>:SOURce?

Function:
The command sets the trigger source as channel (CH1, CH2, CH3, CH4), external trigger (EXT, EXT5) or AC Line.
The< mode> is :EDGE, the <source> may be CHANnel<n>, EXT, EXT5 or ACLine;
The< mode> is :PULSE, the <source> may be CHANnel<n>, EXT or EXT5;
The< mode> is :VIDEO, the <source> may be CHANnel<n>, EXT or EXT5;
The <mode> is :PATTern, <source> may be CHANnel<n>, EXT or EXT5.
The <n> may be 1, 2, 3 or 4.

Returned Format:
The query returns CH1 or CH2, CH3, CH4, EXT, EXT5, ACLINE.

Example:

:TRIG:EDGE:SOUR CHAN1	Set the edge trigger source as channel 1.
:TRIG:EDGE:SOUR?	Return CH1.

3. :TRIGger<mode>:LEVel

Command Format:

:TRIGger<mode>:LEVel <level>[,<src>]
:TRIGger<mode>:LEVel? [,<src>]

Function:

The command sets the voltage level of Edge, Pulse or Video trigger.

- <mode> may be :EDGE, :PULSe or :VIDEO or :PATTern.
- <level> range: (- 6*Scale-Offset①) ~(+6*Scale+Offset①). Scale is the current vertical scale, and the unit is V/div.
- In PATTern mode, <src> should be set to CHANnel<n> or EXT.

NOTE①: Trigger Level range is up to+/-6 Scale, when channel has offset, it needs to detract offset ,such as 1V tap position, 1V offset, the trigger range is -7V~5V.

Returned Format:

The query returns the value of voltage level, and the unit is V.

Example:

:TRIG:EDGE:LEV 2	Set the trigger level as 2V.
:TRIG:EDGE:LEV?	Return 2.000e000.

4. :TRIGger<mode>:SWEep

Command Format:

:TRIGger<mode>:SWEep {AUTO|NORMal|SINGle}
:TRIGger<mode>:SWEep?

Function:

The command sets the trigger mode. The <mode> may be :EDGE, :PULSe or :PATTern.

- AUTO: When no trigger is existing, the system will generate a trigger signal to force trigger;

- NORMal: Acquire waveform when trigger occurred;
- SINGle: Execute once trigger when all the condition are marched and stop.

Returned Format:
The query returns AUTO or NORMAL, SINGLE.

Example:
:TRIG:EDGE :SWE AUTO Set the trigger type as AUTO.
:TRIG:EDGE :SWE? Return AUTO.

5. :TRIGger:SENSitivity

Command Format:
:TRIGger:SENSitivity <count>
:TRIGger:SENSitivity?

Function:
The command sets the trigger sensitivity. The <count> range is 0.1div~1div.

Returned Format:
The query returns the value of trigger sensitivity, and the unit is div.

Example:
:TRIG:SENS 0.2 Set the trigger sensitivity as 0.2div.
:TRIG:SENS? Return 2.000e-001.

6. :TRIGger:COUPling

Command Format:
:TRIGger:COUPling {DC|AC|LF}
:TRIGger:COUPling?

Function:
The command sets the coupling mode.
- DC: Allow all signals pass;
- AC: Reject DC signals and attenuate the signal below 10Hz;
- LF: Reject DC signals and attenuate the signals below 8kHz.

Returned Format:
The query returns DC, AC or LF.

Example:
:TRIG:COUP DC Set the coupling mode as DC.
:TRIG:COUP? Return DC.

7. :TRIGger:HFREject

Command Format:
:TRIGger:HFREject {{1|ON}|{0|OFF}}
:TRIGger:HFREject?

Function:
The command sets high frequency reject function on or off.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:
:TRIG:HFRE ON Set HFR on.
:TRIG:HFRE? Return 1.

8. :TRIGger:HOLDoff

Command Format:
:TRIGger:HOLDoff <count>
:TRIGger:HOLDoff?

Function:
The command sets the holdoff time of a trigger. Holdoff time indicates the waiting time before oscilloscope starts a new trigger. During Holdoff, the oscilloscope will not trigger until Holdoff ends.
The <count> range is 100ns~1.5s.

Returned Format:
The query returns the value of holdoff time, and the unit is s.

Example:

:TRIG:HOLD 0.0001	Set the holdoff time as 100μs.
:TRIG:HOLD?	Return 1.000e-004.

9. :TRIGger:STATus?

Query Format:

:TRIGger:STATus?

Function:

The command queries the current status of the oscilloscope. The status may be RUN, STOP, T' D, WAIT, SCAN or AUTO.

Returned Format:

The query returns RUN or STOP, T'D, WAIT, AUTO.

10. :Trig%50

Command Format:

:Trig%50

Function:

The command sets the trigger level at the vertical midpoint of the signal amplitude.

11. :FORCetrig

Command Format:

:FORCetrig

Function:

The command will produce a trigger signal to force the oscilloscope trigger and to display a waveform when there is no suitable trigger condition.

NOTE: It is mainly applicable to the "Normal" and "Single" trigger modes.

12. :SINGLE

Command Format:

:SINGLE

Function:

The command sets the trigger mode as Single trigger, means that collect a waveform when detect a trigger signal, then stop running.

EDGE Trigger

1. :TRIGger:EDGE:SLOPe

Command Format:

:TRIGger:EDGE:SLOPe {POSitive|NEGative|ALTernation}
:TRIGger:EDGE:SLOPe?

Function:

The command sets the trigger edge as POSitive (rising edge), NEGative (falling edge) or ALTernation (rising and falling edge).

Returned Format:

The query returns POSITIVE or NEGATIVE, ALTERNATION.

Example:

:TRIG:EDGE:SLOP POS	Set the trigger edge as rise edge.
:TRIG:EDGE:SLOP?	Return POSITIVE.

PULSe Trigger

1. :TRIGger:PULSe:MODE

Command Format:

:TRIGger:PULSe:MODE <mod>
:TRIGger:PULSe:MODE?

Function:

The command sets the trigger condition. <mod> can be: +GREaterthan (positive pulse width greater than), +LESSthan (positive pulse width less than), + EQUal (positive pulse width equal), -GREaterthan (negative pulse width greater than), -LESSthan (negative pulse width less than) or –EQUal (negative pulse width equal).

Returned Format:

The query returns +GREATER THAN or +LESS THAN, +EQUAL, -GREATER THAN, -LESS THAN, –EQUAL.

Example:

:TRIG:PULS:MODE +GRE	Set the trigger condition as +GREaterthan.
:TRIG:PULS:MODE?	Return +GREATER THAN.

2. :TRIGger:PULSe:WIDTh

Command Format:

:TRIGger:PULSe:WIDTh <wid>
:TRIGger:PULSe:WIDTh?

Function:

The command sets the pulse width. The <wid> range is 20ns ~10s.

Returned Format:

The query returns the value of pulse width, and the unit is s.

Example:

:TRIGr:PULS:WIDT 0.001	Set the pulse width as 1ms.
:TRIG:PULS:WIDT?	Return 1.000e-003.

VIDEO Trigger

1. :TRIGger:VIDEO:MODE

Command Format:
:TRIGger:VIDEO:MODE <mode>
:TRIGger:VIDEO:MODE?

Function:
The command sets the trigger sync mode as ODDfield, EVENfield, LINE or ALLlines.

Returned Format:
The query returns ODD FIELD, EVEN FIELD, LINE or ALL LINES.

Example:

:TRIG:VIDEO:MODE EVEN	Set the trigger sync mode as even field.
:TRIG:VIDEO:MODE?	Return EVEN FIELD.

2. :TRIGger:VIDEO:POLarity

Command Format:
:TRIGger:VIDEO:POLarity {POSitive|NEGative}
:TRIGger:VIDEO:POLarity?

Function:
The command sets the video polarity as POSitive (it is applicable for the video signal that the black level is low) or NEGative (the black level is high).

Returned Format:
The query returns POSITIVE or NEGATIVE.

Example:

:TRIG:VIDEO:POL POS	Set the video polarity as positive.
:TRIG:VIDEO:POL?	Return POSITIVE.

3. :TRIGger:VIDEO:STANdard

Command Format:

:TRIGger:VIDEO:STANdard {NTSC|PALSecam}
:TRIGger:VIDEO:STANdard?

Function:

The command sets the video standard as NTSC or PAL/SECAM.

Returned Format:

The query returns NTSC or PAL/SECAM.

Example:

:TRIG:VIDEO:STAN PALS	Set the video standard as PAL/SECAM.
:TRIG:VIDEO:STAN?	Return PAL/SECAM.

4. :TRIGger:VIDEO:LINE

Command Format:

:TRIGger:VIDEO:LINE <value>
:TRIGger:VIDEO:LINE?

Function:

The command sets the number of sync specified line. In NTSC standard, the <value> range is 1~525; in PAL standard, the <value> range is 1~625.

Returned Format:

The query returns the number of specified line.

Example:

:TRIG:VIDEO:LINE 25	Set the number of sync specified line as 25.
:TRIG:VIDEO:LINE?	Return 25.

PATTern Trigger

1. :TRIGger:PATTern:PATTern

Command Format:

:TRIGger:PATTern:PATTern <value>,<mask>,<ext source>[,<edge source>,<edge>]
:TRIGger:PATTern:PATTern?

Function:

The command sets the code pattern of signals.

- <value>:
  Code pattern values of the channels. It is a 16 bit unsigned integer (High is 1, Low is 0).
- <mask>:
  Mask code of the channels. It is a 16 bit unsigned integer (enable is 1, X is 0) which indicates whether the mask code is 1 or 0. The relationship between <mask> and <value> is "And" , if the mask of a channel is 0, which denotes this channel is ineffective and the corresponding setting of oscilloscope is "X"; if the mask is 1, <value> will decide whether the channel is H or L.
- <ext source>:
  It is external trigger signal, and EXT5 is 1, EXT is 0;
- <edge source>:
  It is the current channel, its range: 0(CH1), 1(CH2), 2(CH3), 3(CH4), 4(EXT5);
- <edge>:
  It is the code pattern of current channel. The rising <edge> is 1and the falling<edge> is 0.

NOTE: The priority of <edge> is higher than <mask>.

Returned Format:

The query returns the value, the mask, the ext source, the edge source and the edge. The value and the mask are expressed in decimal.

Example:

:TRIG:PATT:PATT 31,31,1,2,1	Set the code pattern.
:TRIG:PATT:PATT?	Return 27, 31, EXT5, Channel3, Positive.

ALTernation Trigger

1. :TRIGger:ALTernation:SOURce

Command Format:

:TRIGger:ALTernation:SOURce <source>
:TRIGger:ALTernation:SOURce?

Function:

The command selects the alternation trigger channel. The <source> may be CH1CH2, CH1CH3, CH1CH4, CH2CH3, CH2CH4 or CH3CH4.

Returned Format:

The query returns CH1CH2 or CH1CH3, CH1CH4, CH2CH3, CH2CH4, CH3CH4.

Example:

:TRIG:ALT:SOUR CH1CH2	Set the alternation channel as CH1CH2.
:TRIG:ALT:SOUR?	Return CH1CH2.

2. :TRIGger:ALTernation:CURRentSOURce

Command Format:

:TRIGger:ALTernation:CURRentSOURce <source>
:TRIGger:ALTernation:CURRentSOURce?

Function:

The command sets the current channel. The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Returned Format:

The query returns SOURceA or SOURceB.

Example:

:TRIG:ALT:SOUR CH1CH2	Set the alternation channel as CH1CH2.
:TRIG:ALT:CURRSOUR SOURB	Set the current channel as source B.
:TRIG:ALT:CURRSOUR?	Return SOURceB.

3. :TRIGger:ALTernation:TYPE

Command Format:

:TRIGger:ALTernation:TYPE <type>[,<source>]
:TRIGger:ALTernation:TYPE? [<source>]

Function:

The command sets the trigger type. The <type> may be EDGE, PULSe or VIDEO, and the <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Returned Format:

The query returns EDGE or PULSE, VIDEO.

Example:

:TRIG:ALT:TYPE EDGE,SOURB Set the trigger type as edge trigger.
:TRIG:ALT:TYPE? SOURB Return EDGE.

4. :TRIGger:ALTernation:TimeSCALe

Command Format:

:TRIGger:ALTernation:TimeSCALe <value>[,<source>]
:TRIGger:ALTernation:TimeSCALe? [<source>]

Function:

The command sets the time scale of current channel. The <value> range is 2ns~20ms, and the <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.
In NORMAL mode, different types of instruments have different sweep ranges:
DS1204B, <scale_val> range：1ns/div~50s/div.
DS1104B, <scale_val>range：2ns/div~50s/div.
DS1064B, <scale_val>range：5ns/div~50s/div.

Returned Format:

The query returns the value of time scale, and the unit is s.

Example:

:TRIG:ALT:TSCAL 0.001,SOURB Set the time scale as 1ms.

:TRIG:ALT:TSCAL? SOURB Return 1.000e-003.

5. :TRIGger:ALTernation:TimeOFFSet

Command Format:

:TRIGger:ALTernation:TimeOFFSet <value>[,<source>]
:TRIGger:ALTernation:TimeOFFSet? [<source>]

Function:

The command sets the timebase offset.
In NORMAL mode, <value>: 1s ~ memory capacitance;
In STOP mode, <value: -500s ~ +500s.

Returned Format:

The query returns the value of timebase offset, and the unit is s.

Example:

:TRIG:ALT:TOFFS 0.0002,SOURB Set the timebase offset as 200μs.
:TRIG:ALT:TOFFS? SOURB Return 2.000e-004.

6. :TRIGger:ALTernation:LEVel

Command Format:

:TRIGger:ALTernation:LEVel <value>[,<source>]
:TRIGger:ALTernation:LEVel? [<source>]

Function:

The command sets the trigger level of current channel. The <value> range <value> range: (- 6*Scale-Offset①) ~(+6*Scale+Offset①).Scale is the current vertical scale, and the unit is V/div. the <source> may be SOURceA or SOURceB, and the source A and B is different according to the current alternation channel.

NOTE①: Trigger Level range is up to+/-6 Scale, when channel has offset, it needs to detract offset ,such as 1V tap position, 1V offset, the trigger range is -7V~5V.

Returned Format:

The query returns the value of trigger voltage level, and the unit is V.

Example:

:TRIG:ALT:LEV 2, SOURB Set the trigger voltage level as 2V.
:TRIG:ALT:LEV? SOURB Return 2.000e000.

7. :TRIGger:ALTernation:EDGE:SLOPe

Command Format:

:TRIGger:ALTernation:EDGE:SLOPe <value>[,<source>]
:TRIGger:ALTernation:EDGE:SLOPe? [<source>]

Function:

The command sets the edge type of edge trigger in current channel as POSitive (rising edge) or NEGative (falling edge). The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Returned Format:

The query returns POSITIVE or NEGATIVE.

Example:

:TRIG:ALT:EDGE:SLOP POS, SOURB Set the edge type as rising edge.
:TRIG:ALT:EDGE:SLOP? SOURB Return POSITIVE.

8. :TRIGger:ALTernation:PULSe:MODE

Command Format:

:TRIGger:ALTernation:PULSE:MODE <value>[,<source>]
:TRIGger:ALTernation:PULSe:MODE? [<source>]

Function:

The command sets the trigger condition of pulse trigger. The <value> may be +GREaterthan, +LESSthan, + EQUal, -GREaterthan, -LESSthan or –EQUal. The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Returned Format:

The query returns +GREATER THAN or +LESS THAN, +EQUAL, -GREATER THAN,

-LESS THAN, –EQUAL.

Example:

:TRIG:ALT:PULS:MODE +GRE, SOURB Set the trigger condition.
:TRIG:ALT:PULS:MODE? SOURB Return +GREATER THAN.

9. :TRIGger:ALTernation:PULSe:TIME

Command Format:

:TRIGger:ALTernation:PULSe:TIME <value>[,<source>]
:TRIGger:ALTernation:PULSe:TIME? [<source>]

Function:

The command sets the pulse width, the value range is 20ns~10s. The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Returned Format:

The query returns the value of pulse width, and the unit is s.

Example:

:TRIG:ALT:PULS:TIME 0.002, SOURB Set the pulse width as 2ms.
:TRIG:ALT:PULS:TIME? SOURB Return 2.000e-003.

10. :TRIGger:ALTernation:VIDEO:POLarity

Command Format:

:TRIGger:ALTernation:VIDEO:POLarity {POSitive|NEGative }[,<source>]
:TRIGger:ALTernation:VIDEO:POLarity? [<source>]

Function:

The command sets the video polarity as POSitive or NEGative. The <source> may be SOURceA or SOURceB, and the source A and B are ivarying with the current alternation channel.

Returned Format:

The query returns POSITIVE or NEGATIVE.

Example:

:TRIG:ALT:VIDEO:POL POS,SOURB Set the video polarity as positive.
:TRIG:ALT:VIDEO:POL? SOURB Return POSITIVE.

11. :TRIGger:ALTernation:VIDEO:STANdard

Command Format:

:TRIGger:ALTernation:VIDEO:STANdard {NTSC|PALSecam}[,<source>]
:TRIGger:ALTernation:VIDEO:STANdard? [<source>]

Function:

The command sets the video standard as NTSC or PAL/SECAM. The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Returned Format:

The query returns NTSC or PAL/SECAM.

Example:

:TRIG:ALT:VIDEO:STAN NTSC,SOURB Set the video standard as NTSC.
:TRIG:ALT:VIDEO:STAN? SOURB Return NTSC.

12. :TRIGger:ALTernation:VIDEO:MODE

Command Format:

:TRIGger:ALTernation:VIDEO:MODE
{ALLLins|ODDField|EVENfield|LINE}[,<source>]
:TRIGger:ALTernation:VIDEO:MODE? [<source>]

Function:

The command sets the sync mode of alternation trigger and video trigger as ALLLINES, ODDFIELD, EVENFIELD or LINE. The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Returned Format:

The query returns ALL LINES or ODD FIELD, EVEN FIELD, LINE

Example:

:TRIG:ALT:VIDEO:MODE ALLLINES,SOURB	Set the sync mode as all lines.
:TRIG:ALT:VIDEO:MODE? SOURB	Return ALL LINES.

13. :TRIGger:ALTernation:VIDEO:LINE

Command Format:

:TRIGger:ALTernation:VIDEO:LINE <value>[,<source>]
:TRIGger:ALTernation:VIDEO:LINE? [<source>]

Function:

The command sets the number of sync specified line. In NTSC standard, the <value> range is 1~525; in PAL/SECAN standard, the <value> range is 1~625. The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Returned Format:

The query returns the number of specified line.

Example:

:TRIG:ALT:VIDEO:LINE 100, SOURB	Set the specified line number as 100.
:TRIG:ALT:VIDEO:LINE? SOURB	Return 100.

14. :TRIGger:ALTernation:COUPling

Command Format:

:TRIGger:ALTernation:COUPling {DC|AC|LF}[,<source>]
:TRIGger:ALTernation:COUPling? [<source>]

Function:

The command sets the coupling mode.
DC: Allow all signals pass;
AC: Reject DC signals and attenuate AC signals below 10Hz.
LF: Reject DC and attenuate low frequency signals below 8kHz.
The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Returned Format:
The query returns DC, AC or LF.

Example:
:TRIG:ALT:COUP DC, SOURB Set the coupling mode as DC.
:TRIG:ALT:COUP? SOURB Return DC.

15. :TRIGger:ALTernation:HFREject

Command Format:
:TRIGger:ALTernation:HFREject {{1|ON}|{0|OFF}}
:TRIGger:ALTernation:HFREject?

Function:
The command sets high frequency reject function of ALTernation Trigger on or off.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:
:TRIG:ALT:HFRE ON Set HFR on.
:TRIG:ALT:HFRE? Return 1.

16. :TRIGger:ALTernation:HOLDoff

Command Format:
:TRIGger:ALTernation:HOLDoff <count>[,<source>]
:TRIGger:ALTernation:HOLDoff? [<source>]

Function:
The command sets the holdoff time to trigger the specified source alternately. Holdoff time is the waiting time of oscilloscope before starting a new trigger. During Holdoff, oscilloscope will not trigger until Holdoff ends. The <count> range is 100ns~1.5s. The <source> may be SOURceA or SOURceB.

Returned Format:

The query returns the value of holdoff time, and the unit is s.

Example:

:TRIG:HOLD 0.0001, SOURA	Set the holdoff time of source A as 100us.
:TRIG:HOLD? SOURA	Return 1.000e-004.

17. :TRIGger:ALTernation:SENSitivity

Command Format:

:TRIGger:ALTernation:SENSitivity <count>[,<source>]
:TRIGger:ALTernation:SENSitivity? [<source>]

Function:

The command sets the trigger sensitivity of alternation trigger, the count range is 0.1div~1div. The <source> may be SOURceA or SOURceB, and the source A and B are varying with to the current alternation channel.

Returned Format:

The query returns the value of trigger sensitivity, and the unit is div.

Example:

:TRIG:ALT:SENS 0.1, SOURceB	Set the trigger sensitivity as 01.
:TRIG:ALT:SENS? SOURceB	Return 1.000e-001.

MATH Commands

MATH Commands are used to display the result of adding, subtracting, multiplying and FFT operation for the signals from CH1, CH2, CH3 and CH4. The results can be measured by the grid and the cursor.

MATH Commands include:

- :MATH:DISPlay

We will give detailed introductions for each command in the following parts.

1. :MATH:DISPlay

Command Format:

:MATH:DISPlay {{1|ON}|{0|OFF}}
:MATH:DISPlay?

Function:

The command sets Math waveform on or off.

Returned Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:MATH:DISP ON	Set Math waveform on.
:MATH:DISP?	Return 1.

CHANnel Commands

CHANnel Commands are used to set the vertical system from every channel separately.

CHANnel Commands include:

- :CHANnel<n>:BWLimit
- :CHANnel<n>:COUPling
- :CHANnel<n>:DISPlay
- :CHANnel<n>:INVert
- :CHANnel<n>:OFFSet
- :CHANnel<n>:PROBe
- :CHANnel<n>:SCALe
- :CHANnel<n>:FILTer
- :CHANnel<n>:MEMoryDepth?
- :CHANnel<n>:VERNier
- :CHANnel<n>:UNITs

We will give detailed introductions for each command in the following parts.

1. :CHANnel<n>:BWLimit

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

Function:
The command sets bandwidth limit function ON (limit band width to 20MHz to reduce noise) or OFF (full band width). The <n> may be 1, 2, 3 or 4.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:CHAN2:BWL OFF	Set BW limit function of channel 2 off.
:CHAN2:BWL?	Return 0.

2. :CHANnel<n>:COUPling

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

Function:
The command sets coupling mode as DC (both AC and DC components of the input signal can pass), AC (the DC component of the input signal can not pass) or GND (disconnect the input signal). The <n> may be 1, 2, 3 or 4.

Returned Format:
The query returns AC or DC, GND.

Example:

:CHAN2:COUP DC	Set the coupling mode of channel 2 as DC.
:CHAN2:COUP?	Return DC.

3. :CHANnel<n>:DISPlay

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

Function:
The command sets the channel ON or OFF. The <n> may be 1, 2, 3 or 4.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Examples:
:CHAN2:DISP ON Set channel 2 on.
:CHAN2:DISP? Return 1.

4. :CHANnel<n>:INVert

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

Function:
The command sets waveform invert function ON (the inverted waveform is display) or OFF (the normal waveform is display). The <n> may be 1, 2, 3 or 4.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:
:CHAN2:INV OFF Set the invert function of channel 2 off.
:CHAN2:INV? Return 0.

5. :CHANnel<n>:OFFSet

Command Format:
:CHANnel<n>:OFFSet <offset>
:CHANnel<n>:OFFSet?

Function:

The command sets the vertical offset. The <n> may be 1, 2, 3 or 4.
Scale≥250mV, <offset>: -40V~ +40V;
Scale<250mV, <offset>: -2V ~ +2V.

Returned Format:

The query returns the value of offset, and the unit is V.

Example:

:CHAN2:OFFS 20	Set the vertical offset of channel 2 as 20V.
:CHAN2:OFFS?	Return 2.000e001.

6. :CHANnel<n>:PROBe

Command Format:

:CHANnel<n>:PROBe <attn>
:CHANnel<n>:PROBe?

Function:

The command sets the attenuation factor of probe. The <n> may be 1, 2, 3 or 4, and the <attn> may be 0.001 X, 0.01 X, 0.1 X, 1X, 2 X, 5X, 10X, 20 X, 50X, 100X, 200 X, 500X or 1000X.

Returned Format:

The query returns the value of attenuation factor.

Example:

:CHAN2:PROB 10X	Set the annenuation factor of channel 2 as 10X.
:CHAN2:PROB?	Return 10X.

7. :CHANnel<n>:SCALe

Command Format:

:CHANnel<n>:SCALe <range>
:CHANnel<n>:SCALe?

Function:

The command sets the vertical scale for magnifying waveform. The <n> may be 1, 2, 3 or 4.
Probe 0.001X, <range>: 2μV ~ 10mV;
Probe 0.01X, <range>: 20μV ~ 100mV;
Probe 0.1X, <range>: 200μV ~ 1V;
Probe 1X, <range>: 2mV ~ 10V;
Probe 2X, <range>: 4mV ~ 20V;
Probe 5X, <range>: 10mV ~ 50V;
Probe 10X, <range>: 20mV ~ 100V;
Probe 20X, <range>: 40mV ~ 200V;
Probe 50X, <range>: 100mV ~ 500V;
Probe 100X, <range>: 200mV ~ 1kV;
Probe 200X, <range>: 400mV ~ 2kV;
Probe 500X,<range>: 1V ~ 5kV;
Probe 1000X, <range>: 2V ~ 10kV.

Returned Format:
The query returns the value of vertical scale, and the unit is V.

Example:

:CHAN2:PROB 10X	Set the attenuation factor of channel 2 as 10X.
:CHAN2:SCAL 20	Set the vertical scale of channel 2 as 20V.
:CHAN2:SCAL?	Return 2.000e001.

8. :CHANnel<n>:FILTer

Command Format:
:CHANnel<n>:FILTer {{1|ON}|{0|OFF}}
:CHANnel<n>:FILTer?

Function:
The command sets digital filter function ON or OFF. The <n> may be 1, 2, 3 or 4.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:CHAN2:FILT OFF Set the digital filter of channel 2 off.
:CHAN2:FILT? Return 0.

9. :CHANnel<n>:MEMoryDepth?

Command Format:
:CHANnel<n>:MEMoryDepth?

Function:
This command is query the memory depth on channel x.
There are three instances:
1) Alternate trigger: 8192
2) Slow scan or ROLL: 0~8192
3) Others: 8192

NOTE: In Slow scan mode: In the event of the horizontal timebase is set as 50ms/div or more slowly, the instrument will turn into Slow scan mode. Under this circumstance, the oscilloscope will gather the datum form the left side of the trigger point and then continue gathering the waves form the right side after triggering. If use Slow scan mode to observe the low frequency signal, you are suggested to set the couple mode of channel as DC.

Returned Format:
The query returns the value such as: 8192e003.

10. :CHANnel<n>:VERNier

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

Function:
The command sets the adjustment mode of vertical scale as ON (Fine) or OFF (Coarse). The vertical scale steps by 1-2-5 in Coarse mode; and by equality in Fine mode. The <n> may be 1, 2, 3, 4.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:CHAN2:VERN ON	Set the fine adjustment function of channel 2 on.
:CHAN2:VERN?	Return 1.

11. :CHANnel<n>:UNITs

Command Format:

:CHANnel<n>:UNITs <units>

:CHANnel<n>:UNITs?

Function:

The command sets the unit as VOLTs (V), AMPeres (A), WATTs (W) or UNKNown. The <n> may be 1, 2, 3 or 4.

Returned Format:

The query returns VOLTs or AMPeres, WATTs, UNKNown.

Example:

:CHAN1:UNIT VOLT	Set the unit of channel 1 as V.
:CHAN1:UNIT?	Return VOLTs.

MEASure Commands

MEASure Commands are used for the fundamental measurement operations, and the measurement results are expressed by scientific notation.

MEASure Commands include:

- :MEASure:CLEar
- :MEASure:VPP?
- :MEASure:VMAX?
- :MEASure:VMIN?
- :MEASure:VAMPlitude?
- :MEASure:VTOP?
- :MEASure:VBASe?
- :MEASure:VAVerage?
- :MEASure:VRMS?
- :MEASure:OVERshoot?
- :MEASure:PREShoot?
- :MEASure:FREQuency?
- :MEASure:RISetime?
- :MEASure:FALLtime?
- :MEASure:PERiod?
- :MEASure:PWIDth?
- :MEASure:NWIDth?
- :MEASure:PDUTycycle?
- :MEASure:NDUTycycle?
- :MEASure:PDELay?
- :MEASure:NDELay?
- :MEASure:PPHAse?
- :MEASure:NPHAse?
- :MEASure:TOTal
- :MEASure:SOURce
- :MEASure:DELAySOURce
- :MEASure:PHAseSOURce
- :MEASure:ENABle
- :MEASure:DISable
- :MEASure?

We will give detailed introductions for each command in the following parts.

1. :MEASure:CLEar

Command Format:
:MEASure:CLEar

Function:
The command clears the current measurement parameters.

2. :MEASure:VPP?

Command Format:
:MEASure:VPP? [<source>]

Function:
The command measures the Peak-Peak value of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

NOTE: "<>" in <source> indicates the source is the parameter that must be set in the commond; and "[ ]" indicates the parameter can be set or not set according to your demand. The followings are the same, don't repeat. Please refer to chapter 1 **Symbol Description** about the related explanations.

Returned Format:
The query returns as 5.280e000, and the unit is V.

3. :MEASure:VMAX?

Command Format:
:MEASure:VMAX? [<source>]

Function:
The command measures the maximum of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 2.640e000, and the unit is V.

4. :MEASure:VMIN?

Command Format:
:MEASure:VMIN? [<source>]

Function:
The command measures the minimum of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as -2.640e000, and the unit is V.

5. :MEASure:VAMPlitude?

Command Format:
:MEASure:VAMPlitude? [<source>]

Function:
The command measures the amplitude of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 5.280e000, and the unit is V.

6. :MEASure:VTOP?

Command Format:
:MEASure:VTOP? [<source>]

Function:
The command measures the top value of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 2.640e000, and the unit is V.

7. :MEASure:VBASe?

Command Format:
:MEASure:VBASe? [<source>]

Function:
The command measures the base value of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as -2.640e000, and the unit is V.

8. :MEASure:VAVerage?

Command Format:
:MEASure:VAVerage? [<source>]

Function:
The command measures the average value of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as -4.200e-003, and the unit is V.

9. :MEASure:VRMS?

Command Format:
:MEASure:VRMS? [<source>]

Function:
The command measures the root mean square of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 2.460e000, and the unit is V.

10. :MEASure:OVERshoot?

Command Format:
:MEASure:OVERshoot? [<source>]

Function:
The command measures the overshoot value of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 8.000e003, and the unit is V.

11. :MEASure:PREShoot?

Command Format:
:MEASure:PREShoot? [<source>]

Function:
The command measures the preshoot value of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 8.000e-003, and the unit is V.

12. :MEASure:FREQuency?

Command Format:
:MEASure:FREQuency? [<source>]

Function:
The command measures the frequency of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 1.000e003, and the unit is Hz.

13. :MEASure:RISetime?

Command Format:
:MEASure:RISetime? [<source>]

Function:
The command measures the rise time of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 4.000e-005, and the unit is s.

14. :MEASure:FALLtime?

Command Format:
:MEASure:FALLtime? [<source>]

Function:
The command measures the fall time of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 4.000e-005, and the unit is s.

15. :MEASure:PERiod?

Command Format:
:MEASure:PERiod? [<source>]

Function:
The command measures the period of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 1.000e-003, and the unit is s.

16. :MEASure:PWIDth?

Command Format:
:MEASure:PWIDth? [<source>]

Function:
The command measures the positive pulse width of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 5.000e-004, and the unit is s.

17. :MEASure:NWIDth?

Command Format:
:MEASure:NWIDth? [<source>]

Function:
The command measures the negative pulse width of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 5.000e-004, and the unit is s.

18. :MEASure:PDUTycycle?

Command Format:
:MEASure:PDUTycycle? [<source>]

Function:
The command measures the positive duty cycle of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 5.000e001, and the unit is %.

19. :MEASure:NDUTycycle?

Command Format:
:MEASure:NDUTycycle? [<source>]

Function:
The command measures the negative duty cycle of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as 5.000e001, and the unit is %.

20. :MEASure:PDELay?

Command Format:
:MEASure:PDELay? [<source A>,<source B>]

Function:
The command measures the delay between <sourceA> and <sourceB> relative to the rising edge. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as <-1.000 e-004, and the unit is s.

21. :MEASure:NDELay?

Command Format:
:MEASure:NDELay? [<source A>,<source B>]

Function:
The command measures the delay between <sourceA> and <sourceB> relative to the falling edge. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as <-1.000 e-004, and the unit is s.

22. :MEASure:PPHAse?

Command Format:
:MEASure:PPHAse? [<source A>,<source B>]

Function:
The command measures the phase difference between <sourceA> and <sourceB> relative to the rising edge. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as <-1.000 e-004, and the unit is s.

23. :MEASure:NPHAse?

Command Format:
:MEASure:NPHAse? [<source A>,<source B>]

Function:
The command measures the phase difference between <sourceA> and <sourceB> relative to the falling edge. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns as <-1.000 e-004, and the unit is s.

24. :MEASure:TOTal

Command Format:
:MEASure:TOTal {{1|ON}|{0|OFF}}
:MEASure:TOTal?

Function:
The command sets all the measurement functions on or off.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:MEAS:TOT ON Set the total measurement function on.
:MEAS:TOT? Return 1.

25. :MEASure:SOURce

Command Format:

:MEASure:SOURce <source>
:MEASure:SOURce?

Functions:

The command selects the measurement channel. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:

The query returns CH1 or CH2, CH3, CH4.

Example:

:MEAS:SOUR CHAN1 Measure the signal from CH1.
:MEAS:SOUR? Return CH1.

26. :MEASure:DELAySOURce

Command Format:

:MEASure:DELAySOURce <source>,<source>
:MEASure:DELAySOURce?

Functions:

The command selects the channel for measuring the time delay. The< source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:

The query returns CH1, CH2 or CH1, CH3, CH1, CH4, CH2, CH3, CH2, CH4, CH3, CH4.

Example:

:MEAS:DELASOUR CHAN1 CHAN2 Measure the delay source.

:MEAS:DELASOUR? Return CH1, CH2.

27. :MEASure:PHAseSOURce

Command Format:

:MEASure:PHAseSOURce <source>,<source>

:MEASure:PHAseSOURce?

Functions:

The command selects the channel for measuring the phase delay. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:

The query returns CH1, CH2 or CH1, CH3, CH1, CH4, CH2, CH3, CH2, CH4, CH3, CH4.

Example:

:MEAS:PHASOUR CHAN1 CHAN2 Measure the phase delay.

:MEAS:PHASOUR? Return CH1, CH2.

28. :MEASure:ENABle

Command Format:

:MEASure:ENAble

Functions:

This command is open the locked MEASURE button and allow user to open Auto Measure.

NOTE: Only corresponding unlock command can open AUTO and Auto Measure once they are locked, neither restart nor update is available.

29. :MEASure:DISable

Command Format:

:MEASure:DISable

Function:

This command is lock MEASURE button and forbid user opening Auto Measure.

30. :MEASure?

Command Format:
:MEASure?

Function:
This command is query the state of keylock.

Returned Format:
The query returns Locked or UnLocked.

WAVeform Commands

WAVeform Commands are used to read the data and parameters of waveform on the screen.

WAVeform Commands include:

- :WAVeform:FORMat
- :WAVeform:DATA?
- :WAVeform:POINts
- :WAVeform:POINts:MODE
- :WAVeform:SOURce
- :WAVeform:PREamble?
- :WAVeform:YINCrement?
- :WAVeform:YORigin?
- :WAVeform:XINCrement?
- :WAVeform:XORigin?
- :WAVeform:XREFerence?
- :WAVeform:YREFerence?

We will give detailed introductions for each command in the following parts.

1. :WAVeform:FORMat

Command Format:
:WAVeform:FORMat <value>
:WAVeform:FORMat?

Function:
The command sets the format of waveform data. The <value> may be WORD, BYTE or ASCii.
Difference of WORD, BYTE and ASCii:
ASCii: Returns ASCII values when data ware transformed into character.
Eg: Waveform data is 1000, returns '1','0','0','0', a point correspond many bytes.
BYTE and WORD: They will returns 8 bit and 16 bit values to use datum.
Eg: Waveform data is 1000, returns 1000 in decimal system, a point correspond one byte.

Returned Format:
The query returns WORD or BYTE, ASCii.

Example:
:WAV:FORM ASC Set the data format as ASCII.
:WAV:FORM? Return ASCii.

2. :WAVeform:DATA?

Command Format:
:WAVeform:DATA? [<source>]

Function:
The command reads waveform data from the specified source. <source> may be : CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Returned Format:
The query returns a certain amount of waveform data that specifed by **:WAVeform:POINts**.

NOTE:

- The command returns the data on the screen while waveform playback, at this moment,

only **NORMal** and **MAXimum** mode is available and the system is in **STOP** state.

- 600 points are returned in common operation（+, —, ×）while 500 points are returned in FFT operation in all modes (**NORMal**, **RAW**, **MAXimum**).
- The waveform data read in **NORMal** mode is fixed as 600 points while the system is in **STOP** state, if increase the time base until all the waveforms are displayed on the screen, some invalid data may be contained in data returned under the circumstances. So, you are recommended to read the data in **RAW** mode while in **STOP** state.

Example:

:WAV:DATA? CHAN1

Read the data from CH1.

3. :WAVeform:POINts

Command Format:

:WAVeform:POINts <points>

:WAVeform:POINts?

Function:

This command sets the waveform points need to be returned, the default is 0. <points> has different Value ranges in different modes.

NORMal: 0~600

RAW: 0~8192 or 0~16384 (in half channel state)

NOTE: Half channel indicates selecting one of the channels in CH1 and CH2, or in CH3 and CH4.

Returned Format:

The query returns an integer, for example: 10.

NOTE:

- If you set the waveform points to be 0, the query will return the maximum points in current mode (NORMal: return 600 points, RAW: return current memory depth);
- In **MATH** operation, 600 points are returned no matter what mode it is;
- In **FFT,** the maximum points will always be 500.

Example:

:WAV:POIN 20	Set the waveform points as 20.
:WAV:POIN?	Return 20.

For details about storage format of waveform points, please refer to Page 2-76: **Peak Detect**

4. :WAVeform:POINts:MODE

Command Format:

:WAVeform:POINts:MODE <points_mode>
:WAVeform:POINts:MODE?

Function:

This command sets the mode of waveform points. <points_mode> can be: NORMal, MAXimum or RAW.

NOTE: What will be returned by **:WAVeform:POINts?** in different modes:

- **NORMal:** Return data points currently display on the screen (600 points).
- **RAW:** Return the data points of the memory data (in **STOP** state). In **RUN** state, no data are returned, the system error code is 67 which indicate system condition is not met and the execution failed.
- **MAXimum:** Return the maximum valid data points in current state. In **RUN** state, the screen data points are returned, while in **STOP** state, the memory data points are returned.

	NORmal		**RAW**	**MAX**
	Normal /Average	Peak Detect		In **RUN** state, MAX is the same with NORMal; in **STOP** state, MAX is the same with RAW.
MATH	600	1200	600	
FFT	500	500	500	
CHx	600	1200	8192	
Half-Channel CHx	600	1200	16384	

Returned Format:

The query returns NORMal, MAXimum or RAW.

Example:

:WAV:POIN:MODE NORM	Set the mode as NORMal.
:WAV:POIN:MODE?	Return NORMal.

5. :WAVeform:SOURce

Command Format:
:WAVeform:SOURce <source>
:WAVeform:SOURce?

Function:
The command sets the source of waveform data which is going to check. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.
Returned Format:
The query returns Channel1 or Channel2, Channel3, Channel4, MATH.

Example:

:WAV:SOUR CHAN2	Set the data source as channel 2.
:WAV:SOUR?	Return Channel2.

6. :WAVeform:PREamble?

Command Format:
:WAVeform:PREamble?

Function:
This command queries the current waveform settings.

Returned Format:
The query returns 10 data which are separated by comma ",", they are:
Format,Type,Points,Count,Xinc,Xor,Xref,Yinc,Yor,Yref

Parameter Value:
Format: BYTE – 0; WORD – 1; ASCII – 2.
Type: NORMAL – 0; PEAK_DETECT – 1; AVERAGE – 2.
Points: specified by **:WAVeform:POINts** command.
Count: the "average acquisition time" (in average mode) or "1" (other mode);
Xinc: 1/SaRate (RAW) or TimeScale/50 (NORMAL);
Xor: relative time of the trigger points;
Xref: X reference;
Yinc: Y unit voltage;

Yor: vertical offset relative to YREF;
Yref: Y reference, the middle point of the screen.

Example:
+1,+0,0,+1,8.000e-009,-6.000e-006,+0,4.000e-002,0.000e000,+100

7. :WAVeform:YINCrement?

Command Format:
:WAVeform:YINCrement? [<source>]

Function:
This command queries the Y unit voltage of the specified source. <source> can be: CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.
NOTE: returned value= VoltScale /25

Returned Format:
The query returns Y unit voltage, and the unit is V.

Example:
:WAV:YINC? CHAN2 Return 4.000e000.

8. :WAVeform:YORigin?

Command Format:
:WAVeform:YORigin? [<source>]

Function:
The command queries the vertical offset of specified source. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Returned Format:
The query returns the value of vertical offset, and the unit is V.

Example:
:WAV:YOR? CHAN2 Return -1.600e001.

9. :WAVeform:XINCrement?

Command Format:
:WAVeform:XINCrement? [<source>]

Function:
The command queries the interval time between two points of the specified source. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Returned Format:
The query returns the value of interval, and the unit is s.
Example:
:WAV:XINC? CHAN2 Return 1.000e-003.

10. :WAVeform:XORigin?

Command Format:
:WAVeform:XORigin? [<source>]

Function:
The command queries the time from trigger point to XREF of the specified source. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Returned Format:
The query returns the value of time and the unit is s.

Example:
:WAV:XOR? CHAN2 Return 2.000e-002.

11. :WAVeform:XREFerence?

Command Format:
:WAVeform:XREFerence?

Function:

The command queries the horizontal reference axis.

Returned Format:

The query returns the value of reference axis.

Example:

:WAV:XREF? Return 0.

12. :WAVeform:YREFerence?

Command Format:

:WAVeform:YREFrence?

Function:

The command queries the vertical reference axis. YREFrence is fixed at the verticel middle of the screen (100).

Returned Format:

The query returns the value of reference axis.

Example:

:WAV:YREF? Return 100.

Peak Detect

1. Conditions

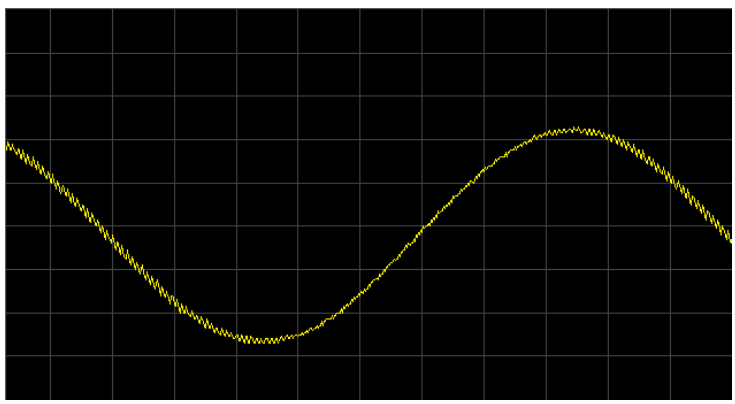
(1) The Peak Detect acquisition mode is open

(2) Time base is greater than 1 us

The acquisition is not Peak Detect in other conditions.

2. Data storage format in memory under Peak Detect

The waveform data in Peak Detect mode are stored in the form of max1-min1, max2-min2, max3-min3. That is, a waveform point has a couple of data: a max and a min, which are stored alternately. As the following figure shown, the waveform display will be serrated.



3. Pick up waveform point of specific time:

(1) CntSpan = (-Time+TrigOffset)*SaRate //The sample points of specific time

(2) CntSpan = CntSpan*2 // Every time contains two data

(3) DstPtr = MidPt – CntSpan // Index of waveform point of specific time

Example: To calculate the index position at -7.69ms point when sample rate is 250k Sa/s and trigger offset is 500 us.

CntSpan = (7.69+0.5)*250=1922+125=2047 //The sample points of specific time

CntSpan = 2047*2 = 4094 // Every time contains two data

DstPtr = 4096 – 4094=2 // Index of waveform point of specific time

Then, you will get two waveform points of this time: Memory(2) and Memory(3).

4. Time calculation of specific index point:

Known condition: memory index point "ind"

(1) TimeSpan = (ind – MidPt)/(SaRate*2) //Take half of the result since every time

contains two waveform points

(2) Time(ind) = TimeSpan + TimeOffset

Example: When sample rate is 250k Sa/s and trigger offset is 500 us.

To calculate time of index point 2:

TimeSpan = (2 – 4096)/(250*2) = -8.188ms

Time(2) = (-8.188+0.5) = -7.688ms

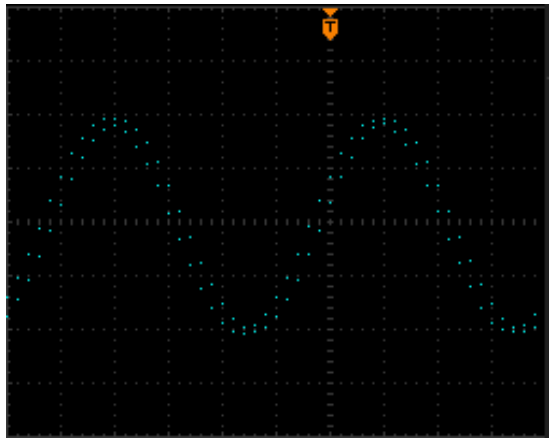
To calculate time of index point 3:

TimeSpan =((3 – 4096)/2)/(250) = (-2047)/250 = -7.686ms

Time(3)=(-8.186+0.5) = -7.686ms

Time(3) ＝ Time(2).

5. Data displays on the screen:



6. Data format of returned screen data:

The screen data obtained by command :WAV:DATA? in Peak Detect mode are stored in the form of T1max – T1min,T2max-T2min, the format of which is the same as memory data storage. Because one time contains two points, the waveform data extends to 1200 ＝ 600*2.

KEY Commands

KEY Commands are used to control the keys and knobs on the operation panel of DS1000B.

KEY Commands include:

- :KEY:LOCK
- :KEY:STORage
- :KEY:UTILity
- :KEY:MEASure
- :KEY:CURSor
- :KEY:ACQuire
- :KEY:DISPlay
- :KEY:HELP
- :KEY:QUICKMEASure
- :KEY:QUICKPRINt
- :KEY:AUTO
- :KEY:RUN
- :KEY:SINGLe
- :KEY:MNUTIME
- :KEY:MNUoff
- :KEY:F1
- :KEY:F2
- :KEY:F3
- :KEY:F4
- :KEY:F5
- :KEY:CH1
- :KEY:CH2
- :KEY:CH3
- :KEY:CH4
- :KEY:MATH
- :KEY:REF
- :KEY:TrigMODE
- :KEY:TrigMENU
- :KEY:TrigFORCe
- :KEY:Trig%50
- :KEY:CH1_VOLT_INC
- :KEY:CH1_VOLT_DEC
- :KEY:CH1_VOLT_Z
- :KEY:CH1_POS_INC
- :KEY:CH1_POS_DEC
- :KEY:CH1_POS_Z
- :KEY:CH2_VOLT_INC
- :KEY:CH2_VOLT_DEC
- :KEY:CH2_VOLT_Z
- :KEY:CH2_POS_INC
- :KEY:CH2_POS_DEC
- :KEY:CH2_POS_Z
- :KEY:CH3_VOLT_INC
- :KEY:CH3_VOLT_DEC
- :KEY:CH3_VOLT_Z
- :KEY:CH3_POS_INC
- :KEY:CH3_POS_DEC
- :KEY:CH3_POS_Z
- :KEY:CH4_VOLT_INC
- :KEY:CH4_VOLT_DEC
- :KEY:CH4_VLOT_Z
- :KEY:CH4_POS_INC
- :KEY:CH4_POS_DEC
- :KEY:CH4_POS_Z
- :KEY:TIME_INC
- :KEY:TIME_DEC
- :KEY:TIME_Z
- :KEY:TIME_POS_INC
- :KEY:TIME_POS_DEC
- :KEY:TIME_POS_Z

- :KEY:FUNC_Z
- :KEY:FUNC_INC
- :KEY:FUNC_DEC
- :KEY:TRIG_LEVEL_INC
- :KEY:TRIG_LEVEL_DEC
- :KEY:TRIG_LEVEL_Z

We will give detailed introductions for each command in the following parts.

1. :KEY:LOCK

Command Format:
:KEY:LOCK { ENABle | DISable }
:KEY:LOCK?

Function:
The command enables and disables the function of Remote control on the keys of front panel.

Returned Format:
The query returns ENABLE or DISABLE.

Example:

:KEY:LOCK ENAB	Enable remote control on the keys of front panel .
:KEY:LOCK?	Return ENABLE.

2. :KEY:STORage

Command Format:
:KEY:STORage

Function:
The command sets storage menu on or off.

3. :KEY:UTILity

Command Format:
:KEY:UTILITY

Function:
The command sets utility menu on or off.

4. :KEY:MEASure

Command Format:
:KEY:MEASure

Function:
The command sets measurement function and its menu on or off.

5. :KEY:CURSor

Command Format:
:KEY:CURSor

Function:
The command enables cursor measurement function and its menu. The cursor mode can be set by sending the command continually.

6. :KEY:ACQuire

Command Format:
:KEY:ACQuire

Function:
The command sets acquire menu on or off.

7. :KEY:DISPlay

Command Format:
:KEY:DISPLAY

Function:
The command sets display menu on or off.

8. :KEY:HELP

Command Format:
:KEY:HELP

Function:
The command sets the built-in help system on or off.

9. :KEY:QUICKMEASure

Command Format:
:KEY:QUICKMEASure

Function:
The command sets quick-measurement function on or off. And it could be set in Measurement menu.

10. :KEY:QUICKPRINt

Command Format:
:KEY:QUICKPRINt

Function:
The command prints and saves the screen.

11. :KEY:AUTO

Command Format:
:KEY:AUTO

Function:
The command sets oscilloscope automatically to display the waveform to be optimal condition.

12. :KEY:RUN

Command Format:
:KEY:RUN

Function:
This command controls the running state of the oscilloscope. The oscilloscope will cutover between RUN and STOP when sending this command continually.

13. :KEY:SINGLe

Command Format:
:KEY:SINGLe

Function:
The command sets the trigger mode as Single trigger.

14. :KEY:MNUTIME

Command Format:
:KEY:MNUTIME

Function:
The command sets horizontal system and its menu on or off.

15. :KEY:MNUoff

Command Format:
:KEY:MNUoff

Function:
The command sets menu display function on or off.

16. :KEY:F1

Command Format:
:KEY:F1

Function:
The command selects the first option in current menu. If the subordinate menu existing and its options are available, then these options could be selected circularly when send repeatly the command.

17. :KEY:F2

Command Format:
:KEY:F2

Function:

The command selects the second option in current menu. If the subordinate menu existing and its options are available, then these options could be selected circularly when send repeatly the command.

18. :KEY:F3

Command Format:

:KEY:F3

Function:

The command selects the third option in current menu. If the subordinate menu existing and its options are available, then these options could be selected circularly when send repeatly the command.

19. :KEY:F4

Command Format:

:KEY:F4

Function:

The command selects the fourth option in current menu. If the subordinate menu existing and its options are available, then these options could be selected circularly when send repeatly the command.

20. :KEY:F5

Command Format:

:KEY:F5

Function:

The command selects the fifth option in current menu. If the subordinate menu existing and its options are available, then these options could be selected circularly when send repeatly the command.

21. :KEY:CH1

Command Format:
:KEY:CH1

Function:
The command sets channel 1 and its menu on or off.

22. :KEY:CH2

Command Format:
:KEY:CH2

Function:
The command sets channel 2 and its menu on or off.

23. :KEY:CH3

Command Format:
:KEY:CH3

Function:
The command sets channel 3 and its menu on or off.

24. :KEY:CH4

Command Format:
:KEY:CH4

Function:
The command sets channel 4 and its menu on or off.

25. :KEY:MATH

Command Format:
:KEY:MATH

Function:

The command sets Math function and its menu on or off.

26. :KEY:REF

Command Format:

:KEY:REF

Function:

The command sets reference waveform function and its menu on or off.

27. :KEY:TrigMODE

Command Format:

:KEY:TrigMODE

Function:

The command shifts the trigger mode among AUTO, NORMAL and SINGLE.

28. :KEY:TrigMENU

Command Format:

:KEY:TrigMENU

Function:

The command sets trigger menu on or off.

29. :KEY:TrigFORCe

Command Format:

:KEY:TrigFORCe

Function:

The command is used for forcing trigger.

30. :KEY:Trig%50

Command Format:
:KEY:Trig%50

Function:
This command sets the trigger level at the vertical midpoint of the amplitude of trigger signal.

31. :KEY:FUNC_Z

Command Format:
:KEY:FUNC_Z

Function:
The command selects the multifunction knob.

32. :KEY:FUNC_INC

Command Format:
:KEY:FUNC_INC

Function:
The command increases the offset of multifunction knob.

33. :KEY:FUNC_DEC

Command Format:
:KEY:FUNC_DEC

Function:
The command decreases the offset of multifunction knob.

34. :KEY:CH1_VOLT_INC

Command Format:

:KEY:CH1_VOLT_INC

Function:

The command decreases the vertical scale of channel 1.

35. :KEY:CH1_VOLT_DEC

Command Format:

:KEY:CH1_VOLT_DEC

Function:

The command increases the vertical scale of channel 1.

36. :KEY:CH1_VOLT_Z

Command Format:

:KEY:CH1_VOLT_Z

Function:

The command sets the adjustment mode of vertical scale of channel 1 as Fine or Coarse. The vertical scale steps by 1-2-5 in Coarse; and by equality in Fine.

37. :KEY:CH1_POS_INC

Command Format:

:KEY:CH1_POS_INC

Function:

The command increases the vertical offset of channel 1 evenly.

38. :KEY:CH1_POS_DEC

Command Format:

:KEY:CH1_POS_DEC

Function:

The command decreases the vertical offset of channel 1 evenly.

39. :KEY:CH1_POS_Z

Command Format:
:KEY: CH1_POS_Z

Function:
The command adjusts the vertical offset of channel 1 to zero.

40. :KEY:CH2_VOLT_INC

Command Format:
:KEY:CH2_VOLT_INC

Function:
The command decreases the vertical scale of channel 2.

41. :KEY:CH2_VOLT_DEC

Command Format:
:KEY:CH2_VOLT_DEC

Function:
The command increases the vertical scale of channel 2.

42. :KEY:CH2_VOLT_Z

Command Format:
:KEY:CH2_VOLT_Z

Function:
The command sets the adjustment mode of vertical scale of channel 2 as Fine or Coarse. The vertical scale steps by 1-2-5 in Coarse; and by equality in Fine.

43. :KEY:CH2_POS_INC

Command Format:
:KEY:CH2_POS_INC

Function:
The command increases the vertical position of channel 2 evenly.

44. :KEY:CH2_POS_DEC

Command Format:
:KEY:CH2_POS_DEC

Function:
The command decreases the vertical position of channel 2 evenly.

45. :KEY:CH2_POS_Z

Command Format:
:KEY:CH2_POS_Z

Function:
The command adjusts the vertical offset of channel 2 to zero.

46. :KEY:CH3_VOLT_INC

Command Format:
:KEY:CH3_VOLT_INC

Function:
The command decreases the vertical scale of channel 3.

47. :KEY:CH3_VOLT_DEC

Command Format:
:KEY:CH3_VOLT_DEC

Function:
The command increases the vertical scale of channel 3.

48. :KEY:CH3_VOLT_Z

Command Format:
:KEY:CH3_VOLT_Z

Function:
The command sets the adjustment mode of vertical scaling of channel 3 as Fine or Coarse. The vertical scale steps by 1-2-5 in Coarse; and by equality in Fine.

49. :KEY:CH3_POS_INC

Command Format:
:KEY:CH3_POS_INC

Function:
The command increases the vertical offset of channel 3 evenly.

50. :KEY:CH3_POS_DEC

Command Format:
:KEY:CH3_POS_DEC

Function:
The command decreases the vertical offset of channel 3 evenly.

51. :KEY:CH3_POS_Z

Command Format:
:KEY:CH3_POS_Z

Function:
The command adjusts the vertical offset of channel 3 to zero.

52. :KEY:CH4_VOLT_INC

Command Format:
:KEY:CH4_VOLT_INC

Function:
The command decreases the vertical scale of channel 4.

53. :KEY:CH4_VOLT_DEC

Command Format:
:KEY:CH4_VOLT_DEC

Function:
The command increases the vertical scale of channel 4.

54. :KEY:CH4_VLOT_Z

Command Format:
:KEY:CH4_VOLT_Z

Function:
The command sets the adjustment mode of vertical scale of channel 4 as Fine or Coarse. The vertical scale steps by 1-2-5 in Coarse; and by equality in Fine.

55. :KEY:CH4_POS_INC

Command Format:
:KEY:CH4_POS_INC

Function:
The command increases the vertical offset of channel 4 evenly.

56. :KEY:CH4_POS_DEC

Command Format:

:KEY:CH4_POS_DEC

Function:
The command decreases the vertical offset of channel 4 evenly.

57. :KEY:CH4_POS_Z

Command Format:
:KEY:CH4_POS_Z

Function:
The command adjusts the vertical offset of channel 4 to zero.

58. :KEY:TIME_INC

Command Format:
:KEY:TIME_INC

Function:
The command decreases the time base by 1-2-5 step.

59. :KEY:TIME_DEC

Command Format:
:KEY:TIME_DEC

Function:
The command increases time base by 1-2-5 step.

60. :KEY:TIME_Z

Command Format:
:KEY:TIME_Z

Function:
The command sets delayed scan function on or off.

61. :KEY:TIME_POS_INC

Command Format:
:KEY:TIME_POS_INC

Function:
The command decreases the trigger offset to the horizontal zero point evenly.

62. :KEY:TIME_POS_DEC

Command Format:
:KEY:TIME_POS_DEC

Function:
The command increases the trigger offset to the horizontal zero point evenly.

63. :KEY:TIME_POS_Z

Command Format:
:KEY:TIME_POS_Z

Function:
The command adjusts the trigger offset to the horizontal zero point evenly.

64. :KEY:TRIG_LEVEL_INC

Command Format:
:KEY:TRIG_LEVEL_INC

Function:
The command increases the trigger level evenly.

65. :KEY:TRIG_LEVEL_DEC

Command Format:
:KEY:TRIG_LEVEL_DEC

Function:

The command decreases the trigger level evenly.

66. :KEY:TRIG_LEVEL_Z

Command Format:

:KEY:TRIG_LEVEL_Z

Function:

The command adjusts the trigger level to zero.

SAVe/RECall Commands

SAVe/RECall Commands are used to save and recall the waveform data and image on the screen.

SAVe/RECall Commands include:

- :SAVERECALL:TYPE
- :SAVERECALL:LOCation
- :SAVERECALL:LOAD
- :SAVERECALL:SAVe
- :SAVe:IMAGe:STARt
- :SAVe:IMAGe:FACTors
- :SAVe:IMAGe:FORMat
- :SAVe:WAVeform:STARt
- :SAVe:SETup:STARt
- :SAVe:CSV:STARt
- :RECall:WAVeform:STARt
- :RECall:SETup:STARt

We will give detailed introductions for each command in the following parts.

1. :SAVERECALL:TYPE

Command Format:

:SAVERECALL:TYPE <type>
:SAVERECALL:TYPE?

Function:

The command sets the data type for storage. The <type> may be WAVeform (waveform data) or SETups (data settings).

Returned Format:

The query returns WAVEFORMS or SETUPS.

Example:

:SAVERECALL:TYPE WAV	Set the storage type as waveform data.
:SAVERECALL:TYPE?	Return WAVEFORM.

2. :SAVERECALL:LOCation

Command Format:

:SAVERECALL:LOCation <location>
:SAVERECALL:LOCation?

Function:

The command sets the storage location. The <location> may be 0~9.

Returned Format:

The query returns 0 or 1 ……9.

Example:

:SAVERECALL:LOC 1	Set the storage location as the second.
:SAVERECALL:LOC?	Return 1.

3. :SAVERECALL:LOAD

Command Format:

:SAVERECALL:LOAD

Function:
The command recalls the waveform or setup data from internal flash according to storage type.

4. :SAVERECALL:SAVe

Command Format:
:SAVERECALL:SAVE

Function:
The command saves the waveform or setup to internal flash according to storage type.

5. :SAVe:IMAGe:STARt

Command Format:
:SAVE:IMAGe:STARt <file_spec>

Function:
The command saves the image. The <file_spec> is the file name, which is composed of double quotation marks and ASCII characters, also, the file name length within the double quotation marks must less than 26 characters.
NOTE: The system will add the file format suffix with 4 characters automatically, which are not included in 26 characters.

6. :SAVe:IMAGe:FACTors

Command Format:
:SAVE:IMAGe:FACTors {{1|ON}|{0|OFF}}
:SAVE:IMAGe:FACTors?

Function:
The command sets the saving function of system parameters on or off. The function indicates to save a file which records all system parameters while saving image.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:
:SAVE:IMAG:FACT ON Save the system parameters.
:SAVE:IMAG:FACT? Return 1.

7. :SAVe:IMAGe:FORMat

Command Format:
:SAVE:IMAGe:FORMat <format>
:SAVE:IMAGe:FORMat?

Function:
The command sets the format of saved image. The <format> may be 24bit real color (BMP|BMP24bit), 8bit bitmap (BMP8bit) or PNG (PNG).

Returned Format:
The query returns BMP24bit, BMP8bit or PNG.

Example:
:SAVE:IMAG:FORM BMP Set the format as 24 bit real color.
:SAVE:IMAG:FORM? Return BMP24bit.

8. :SAVe:WAVeform:STARt

Command Format:
:SAVE:WAVeform:STARt <file_spec>

Function:
The command starts the saving waveform function. If the waveforms are in internal flash, the <file_spec> is composed of integers among 0~9; if in external storage medium, the <file_spec> will be the file name, which is composed of double quotation marks and ASCII characters, also, the file name length within the double quotation marks must less than 26 characters.
NOTE: The system will add the file format suffix with 4 characters automatically, which are not included in 26 characters.

9. :SAVe:SETup:STARt

Command Format:
:SAVE:SETup:STARt <file_spec>

Function:
The command starts the saving setup function. If the waveforms are in internal flash, the <file_spec> is composed of integers among 0~9; if in external storage medium, the <file_spec> will be the file name, which is composed of double quotation marks and ASCII characters, also, the file name length within the double quotation marks must less than 26 characters.
NOTE: The system will add the file format suffix with 4 characters automatically, which are not included in 26 characters.

10. :SAVe:CSV:STARt

Command Format:
:SAVE:CSV:STARt <file_spec>]

Function:
The command sets the saving function of CSV file on. CSV file can be saved in external storage medium. The <file_spec> will be the file name, which is composed of double quotation marks and ASCII characters, also, the file name length within the double quotation marks must less than 26 characters.
NOTE: The system will add the file format suffix with 4 characters automatically, which are not included in 26 characters.

11. :RECall:WAVeform:STARt

Command Format:
:RECAll:WAVeform:STARt <file_spec>

Function:
The command sets the recalling waveform function on. If the waveforms are in internal flash, the <file_spec> is composed of integers among 0~9; if in external storage medium, the <file_spec> will be the file name, which is composed of double quotation marks and ASCII characters, also, the file name

length within the double quotation marks must less than 26 characters.

NOTE: The system will add the file format suffix with 4 characters automatically, which are not included in 26 characters.

12. :RECall:SETup:STARt

Command Format:

:RECAll:SETup:STARt <file_spec>

Function:

The command sets the recalling setup function on. If the waveforms are in internal flash, the <file_spec> is composed of integers among 0~9; if in external storage medium, the <file_spec> will be the file name, which is composed of double quotation marks and ASCII characters, also, the file name length within the double quotation marks must less than 26 characters.

NOTE: The system will add the file format suffix with 4 characters automatically, which are not included in 26 characters.

MASK Commands

MASK Commands are used to create and modify the rules for pass/fail test function.

MASK Commands include:

- :MASK:CREate
- :MASK:ENABle
- :MASK:X
- :MASK:Y
- :MASK:SOURce
- :MASK:OPERate
- :MASK:OUTPut
- :MASK:STOPonoutput
- :MASK:SAVE
- :MASK:LOAD
- :MASK:DOWNload
- :MASK:Upload
- :MASK:MSG

We will give detailed introductions for each command in the following parts.

1. :MASK:CREate

Command Format:
:MASK:CREate

Function:
The command creates the rule of passing test.

2. :MASK:ENABle

Command Format:
:MASK:ENABle {{1|ON}|{0|OFF}}
:MASK:ENABle?

Function:
The command sets the state of passing test as ON or OFF.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:
:MASK:ENAB ON Set passing test on.
:MASK:ENAB? Return 1.

3. :MASK:X

Command Format:
:MASK:X <x>
:MASK:X?

Function:
The command sets the rule of testing X direction. The <x> is 0.04div~4div.

Returned Format:
The query returns the x value, and the unit is div.

Example:

:MASK:X 1 Set the X direction rule as 1div.
:MASK:X? Return 1.000e000.

4. :MASK:Y

Command Format:
:MASK:Y <y>
:MASK:Y?

Function:
The command sets the rule of testing Y direction. The <y> is 0.04div~4div.

Returned Format:
The query returns the y value, and the unit is div.

Example:
:MASK:Y 1 Set the Y direction rule as 1div.
:MASK:Y? Return 1.000e000.

5. :MASK:SOURce

Command Format:
:MASK:SOURce <source>
:MASK:SOURce?

Function:
The command sets the passing test source. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Returned Format:
The query returns CHAN1 or CHAN2, CHAN3, CHAN4.

Example:
:MASK:SOUR CHAN1 Set the passing test source as channel 1.
:MASK:SOUR? Return CHAN1.

6. :MASK:OPERate

Command Format:
:MASK:OPERate <opt>
:MASK:OPERate?

Function:
The command sets the function of passing test run or stop. The <opt> may be RUN or STOP.

Returned Format:
The query returns RUN or STOP.

Example:
:MASK:OPER RUN Set the operation of passing test run.
:MASK:OPER? Return RUN.

7. :MASK:OUTPut

Command Format:
:MASK:OUTPut <output>
:MASK:OUTPut?

Function:
The command sets the output mode of passing test. The <output> may be FAIL, PASS, FAIL_SOUND or PASS_SOUND.
NOTE: PASS SOUND is effective when sound setting is on.

Returned Format:
The query returns FAIL or PASS, FAIL_SOUND, PASS_SOUND.

Example:
:MASK:OUTP PASS Set the output mode of passing test as pass.
:MASK:OUTP? Return PASS.

8. :MASK:STOPonoutput

Command Format:
:MASK:STOPonoutput {{1|ON}|{0|OFF}}
:MASK:STOPonoutput?

Function:
The command sets the output stop mode of passing test ON or OFF.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:
:MASK:STOP ON Set the output stop mode of passing test on.
:MASK:STOP? Return 1.

9. :MASK:SAVE

Command Format:
:MASK:SAVE

Function:
The command saves the rule of passing test.

10. :MASK:LOAD

Command Format:
:MASK:LOAD

Function:
The command loads the rule of passing test.

11. :MASK:DOWNload

Command Format:
:MASK:DOWNload <filename>

Function:

The command download the test rule to the external storage equipment, and the <filename> is the file name, which is composed of double quotation marks and ASCII characters, also, the file name length within the double quotation marks must less than 26 characters.

NOTE: The system will add the file format suffix with 4 characters automatically, which are not included in 26 characters.

12. :MASK:Upload

Command Format:

:MASK:Upload <filename>

Function:

The command uploads the test rule from the external storage equipment, and the <filename> is the file name, which is composed of double quotation marks and ASCII characters, also, the file name length within the double quotation marks must less than 26 characters.

NOTE: The system will add the file format suffix with 4 characters automatically, which are not included in 26 characters.

13. :MASK:MSG

Command Format:

:MASK:MSG {{1|ON}|{0|OFF}}
:MASK:MSG?

Function:

The command sets the prompt information function of passing test ON or OFF.

Returned Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:MASK:MSG ON Set the prompt information function on.
:MASK:MSG? Return 1.

CURSor Commands

CURSor Commands are used to set cursor parameters to measure manually and automatically and track the waveform data.

CURSor Commands include:

- :CURSor:MODE
- :CURSor:MANUal:TYPE
- :CURSor:MANUal:SOURce
- :CURSor:MANUal:CURAX
- :CURSor:MANUal:CURAY
- :CURSor:MANUal:CURBX
- :CURSor:MANUal:CURBY
- :CURSor:TRACk:SOURceA
- :CURSor:TRACk:SOURceB
- :CURSor:TRACk:CURA
- :CURSor:TRACk:CURB

We will give detailed introductions for each command in the following parts.

1. :CURSor:MODE

Command Format:
:CURSor:MODE <mode>
:CURSor:MODE?

Function:
The command sets the cursor mode. The <mode> may be CLOSe, MANUal, TRACk or MEASure (measure automatically).

Returned Format:
The query returns CLOSE or MANUAL, TRACK, MEASURE.

Example:
:CURS:MODE TRAC Set the cursor mode as track.
:CURS:MODE? Return TRACK.

2. :CURSor:MANUal:TYPE

Command Format
:CURSor:MANUal:TYPE <type>
:CURSor:MANUal:TYPE?

Function:
The command sets the cursor type of manual cursor. The <type> may be TIME or AMPlitude.

Returned Format:
The query returns Time or Amplitude.

Example:
:CURS:MANU:TYPE TIME Set the cursor type of manual cursor as time.
:CURS:MANU:TYPE? Return Time.

3. :CURSor:MANUal:SOURce

Command Format:

:CURSor:MANUal:SOURce <source>
:CURSor:MANUal:SOURce?

Function:
The command sets the cursor source of manual cursor. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Returned Format:
The query returns Channel1 or Channel2, Channel3, Channel4, Math.

Example:

:CURS:MANU:SOUR CHAN1	Set the cursor source of manual cursor.
:CURS:MANU:SOUR?	Return Channel1.

4. :CURSor:MANUal:CURAX

Command Format:
:CURSor:MANUal:CURAX <value>
:CURSor:MANUal:CURAX?

Function:
The command sets the AX position of manual cursor. The <value> range is 4~297.

Returned Format:
The query returns the value of AX position.

Example:

:CURS:MANU:CURAX 100	Set the AX position of manual cursor as 100.
:CURS:MANU:CURAX?	Return 100.

5. :CURSor:MANUal:CURAY

Command Format:
:CURSor:MANUal:CURAY <value>
:CURSor:MANUal:CURAY?

Function:

The command sets the AY position of manual cursor. The <value> range is 4~194.

Returned Format:

The query returns the value of AY position.

Example:

:CURS:MANU:CURAY 100 Set the AY position of manual cursor as 100.
:CURS:MANU:CURAY? Return 100.

6. :CURSor:MANUal:CURBX

Command Format:

:CURSor:MANUal:CURBX <value>
:CURSor:MANUal:CURBX?

Function:

The command sets the BX position of manual cursor. The <value> range is 4~297.

Returned Format:

The query returns the value of BX position.

Example:

:CURS:MANU:CURBX 100 Set the BX position of manual cursor as 100.
:CURS:MANU:CURBX? Return 100.

7. :CURSor:MANUal:CURBY

Command Format:

:CURSor:MANUal:CURBY <value>

Function:

The command sets the BY position of manual cursor. The <value> range is 4~194.

Query Format:
:CURSor:MANUal:CURBY?

Returned Format:
The query returns the value of BY position.

Example:
:CURS:MANU:CURBY 100 Set the BY position of manual cursor as 100.
:CURS:MANU:CURBY? Return 100.

8. :CURSor:TRACk:SOURceA

Command Format:
:CURSor:TRACk:SOURceA <source>
:CURSor:TRACk:SOURceA?

Function:
The command sets the signal source A of track cursor. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4, MATH or NONE.

Returned Format:
The query returns Channel1 or Channel2, Channel3, Channel4, Math, None.

Example:
:CURS:TRAC:SOURA CHAN1 Set the signal source A of track cursor.
:CURS:TRAC:SOURA? Return Channel1.

9. :CURSor:TRACk:SOURceB

Command Format:
:CURSor:TRACk:SOURceB <source>
:CURSor:TRACk:SOURceB?

Function:
The command sets the signal source B of track cursor. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4, MATH or NONE.

Returned Format:

The query returns Channel1 or Channel2, Channel3, Channel4, Math, None.

Example:

:CURS:TRAC:SOURB CHAN1 Set the signal source B of track cursor.
:CURS:TRAC:SOURB? Return Channel1.

10. :CURSor:TRACk:CURA

Command Format:

:CURSor:TRACk:CURA <value>
:CURSor:TRACk:CURA?

Function:

The command sets the position of track cursor A. The <value> range is 4~297.

Returned Format:

The query returns the position of cursor A.

Example:

:CURS:TRAC:CURA 100 Set the position of track cursor A as 100.
:CURS:TRAC:CURA? Return 100.

11. :CURSor:TRACk:CURB

Command Format:

:CURSor:TRACk:CURB <value>
:CURSor:TRACk:CURB?

Function:

The command sets the position of track cursor B. The <value> range is 4~297.

Returned Format:

The query returns the position of cursor B.

Example:

:CURS:TRAC:CURB 100 Set the position of track cursor B as 100.

:CURS:TRAC:CURB? Return 100.

Other Commands

The following commands are used to set some additional functions: counter, beeper, system language, real-time clock and the state of AUTO key.

Other Commands include:

- :COUNter:ENABle
- :BEEP:ENABle
- :BEEP:ACTion
- :INFO:LANGuage
- :RTC
- :AUToscale:DISable
- :AUToscale:ENable
- :AUToscale?

We will give detailed introductions for each command in the following parts.

1. :COUNter:ENABle

Command Format:
:COUNter:ENABle {{1|ON}|{0|OFF}}
:COUNter:ENABle?

Function:
The command sets the counter ON or OFF.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:COUN:ENAB ON	Set the counter on.
:COUN:ENAB?	Return 1.

2. :BEEP:ENABle

Command Format:
:BEEP:ENABle {{1|ON}|{0|OFF}}
:BEEP:ENABle?

Function:
The command sets the system beeper ON or OFF.

Returned Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:BEEP:ENAB ON	Set the system beeper on.
:BEEP:ENAB?	Return 1.

3. :BEEP:ACTion

Command Format:
:BEEP:ACTion

Function:

The command tests the system beeper.

4. :INFO:LANGuage

Command Format:

:INFO:LANGuage <cmd_lang>
:INFO:LANGuage?

Function:

The command sets the system language. The <cmd_lang> may be SIMPlifiedchinese, TRADitionalchinese, KORean, JAPanese, ENGLish, FRENch, GERMan, ITALian, RUSSian, PORTuguese or SPANish.

Returned Format:

The query returns Simplified Chinese or Traditional Chinese, Korean, Japanese, English, German, French, Italian, Russian, Portuguese, Spanish.

Example:

:INFO:LANG SIMP	Set the system language as SIMPlifiedchinese.
:INFO:LANG?	Return Simplified Chinese.

5. :RTC

Command Format:

:RTC <year>,<month>,<day>,<hour>,<minute>,<second>
:RTC?

Function:

The command sets the system time. The ranges of each parameter are:
<year> : 2000~2099
<month>: 1~12
<day>: 1~31
<hour>: 0~23
<minute>: 0~59
<second>: 0~59

Returned Format:

The query returns the Year, the Month, the day, the hour, the minutes, and the second.

Example:

:RTC 2008,8,8,20,08,08	Set the system time as 08, 08, 08, 08, 08, 08pm.
:RTC?	Return 2008, 8, 8, 20, 8, 8.

6. :AUToscale:DISable

Command Format:

:AUToscale:DISable

Function:

The command disables the AUTO key, and forbids users setting automatically.

7. :AUToscale:ENable

Command Format:

:AUToscale:ENable

Function:

The command enables the AUTO key, and allows users to set automatically.

8. :AUToscale?

Command Format:

:AUToscale?

Function:

Return the AUTOSCALE state.

Returned Format:

The query returns UnLocked or Locked.

Chapter 3 Programming Examples

This chapter lists some programming examples in the development environments of Visual C++ 6.0, Visual Basic 6.0 and LabVIEW 8.6. All the examples are based on VISA (Virtual Instrument Software Architecture).

VISA is an API (Application Programming Interface) used for controlling instruments. It is convenient for users to develop testing applications which are independent of the types of instrument and interface. Note that "VISA" here we mention is NI (National Instrument)-VISA. NI-VISA is an API written by NI based on VISA standard. You can use NI-VISA to achieve the communication between the oscilloscope and PC via GPIB, USB, RS232, LAN and such instrument bus. As VISA has defined a set of software commands, users can control the instrument without understanding the working state of the interface bus. For more details, please refer to NI-VISA help.

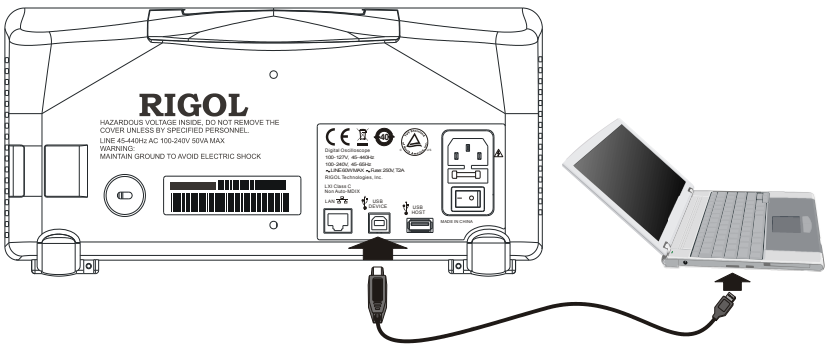
A typical application of VISA contains the fowling parts:

1. Set up the conversation for the existing resource
2. Configure the resource (such as: Baud rate)
3. Close the conversation

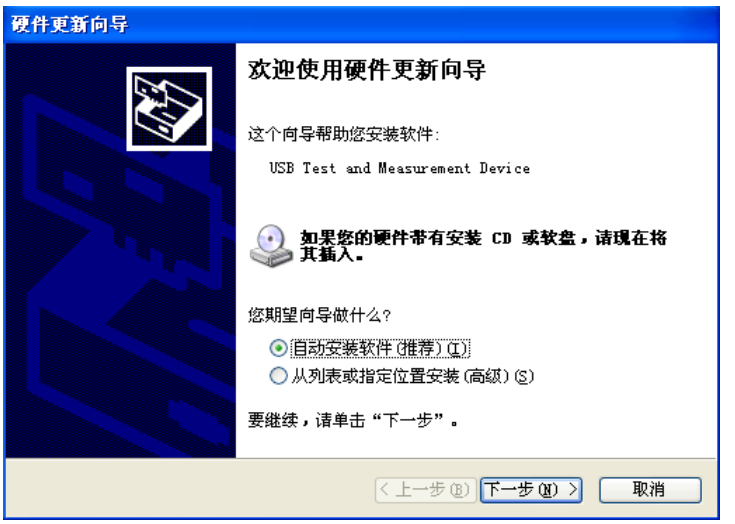
Prepare for Programming

First affirm your computer has installed VISA library of NI (see http://www.ni.com). Here we install it in the default path: C:\Program Files\IVI Foundation\VISA.

In this text, we use USB interface to achieve the communication between the oscilloscope and PC. See the figure below.



After successful connection, turn on the instrument, a dialog will guide you to install the driver of "USB Test and Measurement Device" on the PC. See the figure below:

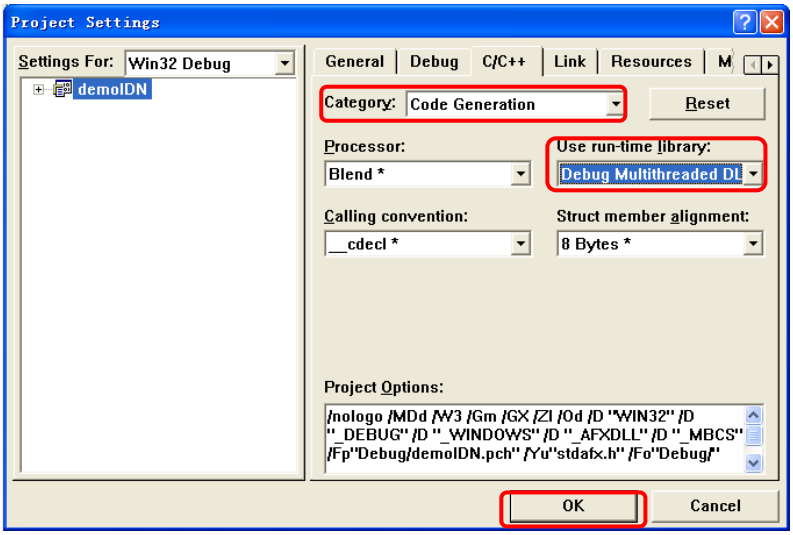


Now, you have finished the preparations. Next, we will give you some programming examples in Visual C++ 6.0, Visual Basic 6.0 and LabVIEW 8.6.

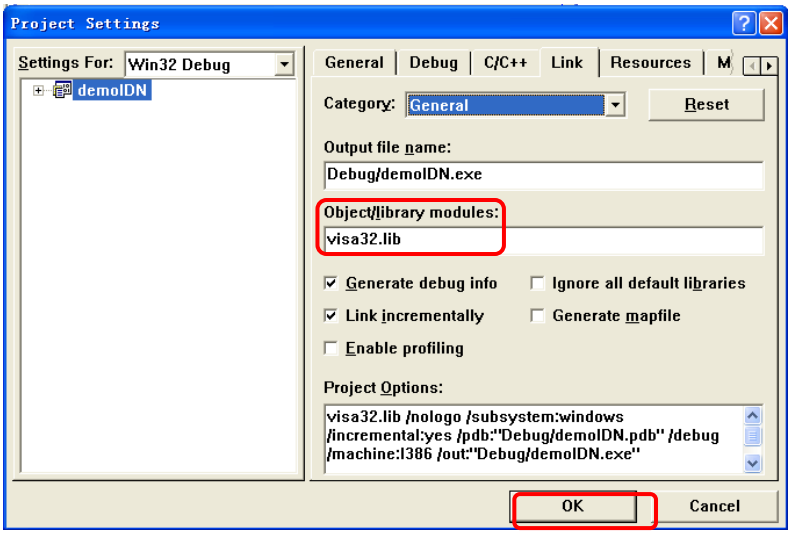
Program in Visual C++ 6.0

Open Visual C++ 6.0, take the following steps:

1. Create a project based on MFC.
2. Choose **Project → Settings → C/C++**; select **"Code Generation"** in **Category** and **"Debug Multithreaded DLL"** in **Use run-time library**; click **OK**.

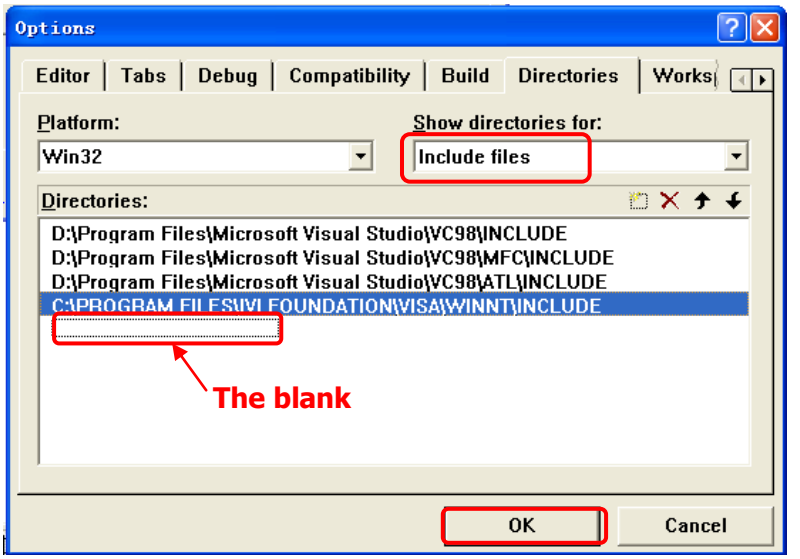


3. Choose **Project→Settings→Link**, add the file **"visa32.lib"** manually in **Object/library modules**.



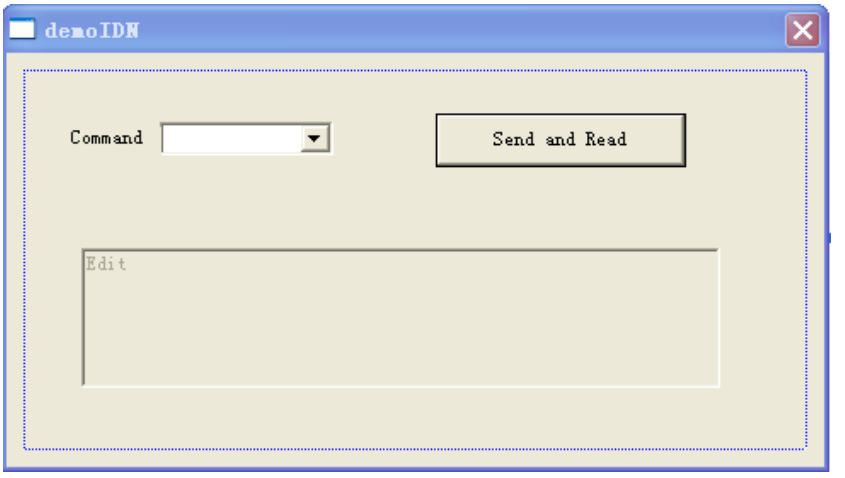
4. Choose **Tools→Options→Directories**; select **"Include files"** in **Show directories for**, and then dblclick the blank in **Directories** to add the path of "**Include**": C:\Program Files\IVI Foundation\VISA\WinNT\include.

Select **"Library files"** in **Show directories for**, and then dblclick the blank in **Directories** to add the path of **"Lib"**:
C:\Program Files\IVI Foundation\VISA\WinNT\lib\msc.



Note: At present, VISA library has been added successfully.

5. Add controls: **Text**, **Com box**, **Button** and **Edit**. See the figure below.



6. Modify the properties of the controls.

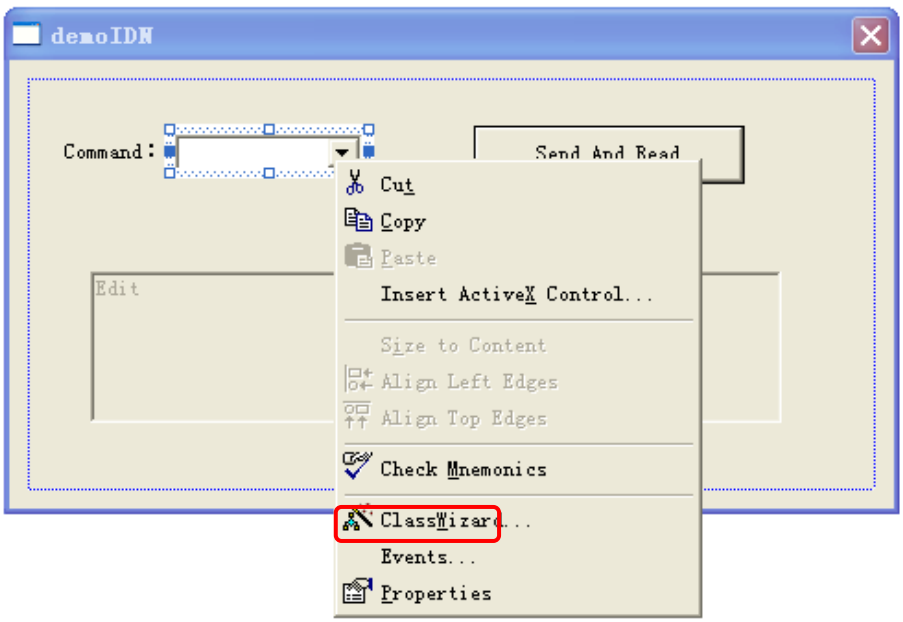
1) Name the Text to be "**Command**".

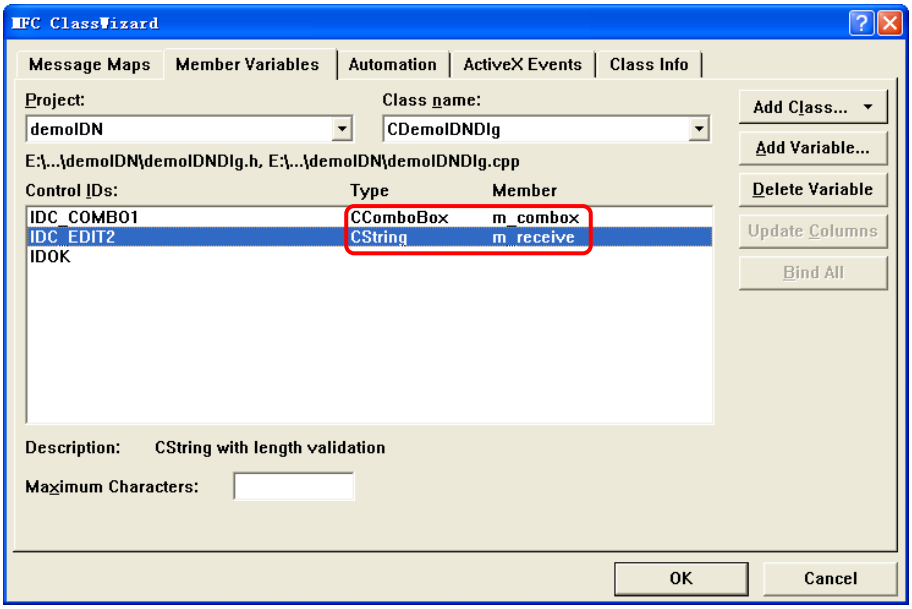
2) Choose **Data** in the property of **Com box**, input three commands manually:
*IDN?
*OPC?
:ACQuire:TYPE?

3) Choose **General** in the property of **Edit** and select **Disable**.

4) Modify the name of **Button** such as: Send and Read.

7. Respectively add two variables **m_combox** and **m_receive** for the controls of **Com box** and **Edit**.





8. Add the codes.

Dblclick the **Button**, enter the programming environment. First of all, declare "#include <visa.h>" in header file, then add the following codes:

```
ViSession defaultRM, vi;
char buf [256] = {0};
CString s,strTemp;
char* stringTemp;

ViChar buffer [VI_FIND_BUFLEN];
ViRsrc matches=buffer;
ViUInt32 nmatches;
ViFindList list;

viOpenDefaultRM (&defaultRM);

// acquire USB resource of visa
viFindRsrc(defaultRM, "USB?*", &list,&nmatches, matches);
viOpen (defaultRM,matches,VI_NULL,VI_NULL,&vi);
viPrintf (vi, "*RST\n");
```

```
// send the receiving commands
m_combox.GetLBText(m_combox.GetCurSel(),strTemp);
strTemp = strTemp + "\n";
stringTemp = (char *)(LPCTSTR)strTemp;
viPrintf (vi,stringTemp);

// read the result
viScanf (vi, "%t\n", &buf);

// display the results
UpdateData (TRUE);
m_receive = buf;
UpdateData (FALSE);
viClose (vi);
viClose (defaultRM);
```

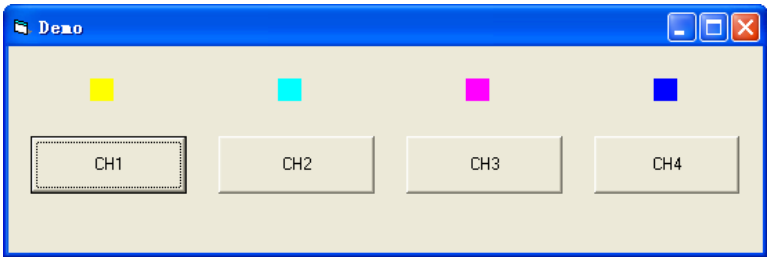
9. Save, build and run the project, you will get an EXE file. When the oscilloscope has been successfully connected with PC, choose a command such as ***IDN?** and click **"Send and Read"**, the oscilloscope will return the result.

Program in Visual Basic 6.0

Open Visual Basic 6.0, take the following steps:

1. Create a **Standard EXE** project.
2. Choose **Project→Add Module→Existing**; find the **"visa.bas"** file in the filefolder of **include** under the path of NI-VISA and add;

3. Add four **Command Buttons** and **Lables** to the demo, each button denotes each channel (CH1~CH4).and each Label denotes different states (yellow, light blue, pink and dark blue which is the channel's color indicates opening, while gray indicates close) of the channels. See the figure below.



4. Choose **Project→Project1 Properties→General**, select **"Form1"** from the drop down box of **Startup Object**.

5. Dblclick **CH1** button to enter the programming environment, add the following codes to achieve the control to it. (for **CH2**, **CH3** and **CH4**, the methods are similar)

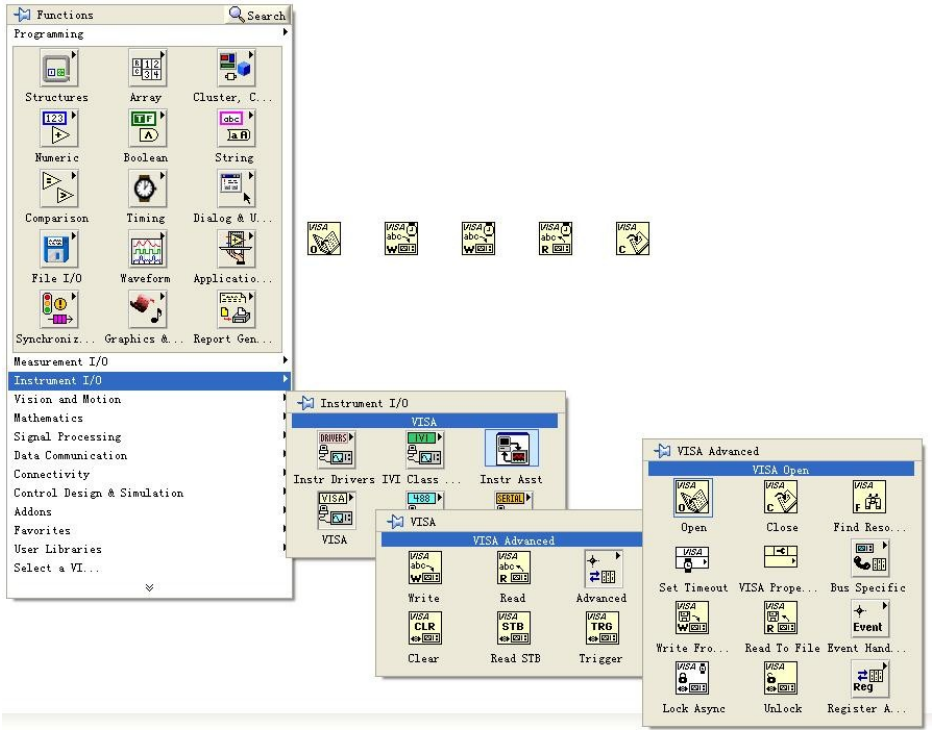
```
Dim defrm As Long
Dim vi As Long
Dim strRes As String * 200
Dim list As Long
Dim nmatches As Long
Dim matches As String * 200   ' reserve to acquire the equipment ID.

' acquire USB resource of visa
Call viOpenDefaultRM(defrm)
Call viFindRsrc(defrm, "USB?*", list, nmatches, matches)

' open the equipment
Call viOpen(defrm, matches, 0, 0, vi)

' send the command to query the state of CH1
Call viVPrintf(vi, ":CHAN1:DISP?" + Chr$(10), 0)

' get the state of CH1
Call viVScanf(vi, "%t", strRes)

If strRes = 1 Then

' send the setting command
Call viVPrintf(vi, ":CHAN1:DISP 0" + Chr$(10), 0)
Label1(0).ForeColor = &H808080   ' gray

Else

Call viVPrintf(vi, ":CHAN1:DISP 1" + Chr$(10), 0)
Label1(0).ForeColor = &HFFFF&   ' yellow

End If

' close the resource
Call viClose(vi)
Call viClose(defrm)
```

6. Save and run the project, you will get a single executable program about demo. When the oscilloscope has been successfully connected with PC, you can open/close each channel conveniently by clicking the button.

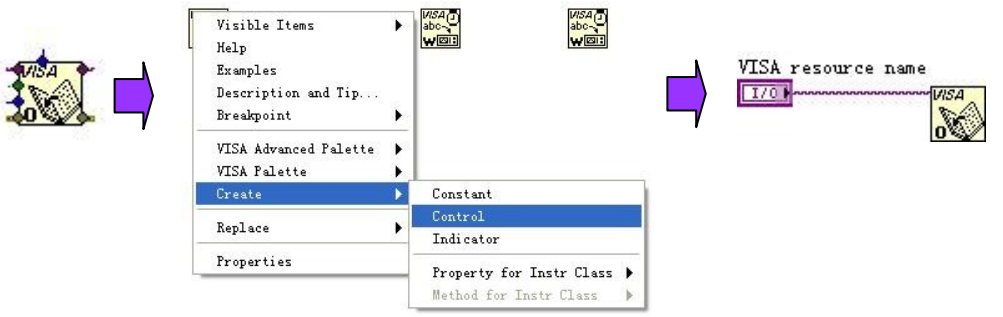
Program in LabVIEW 8.6

Open LabVIEW 8.6, take the following steps:

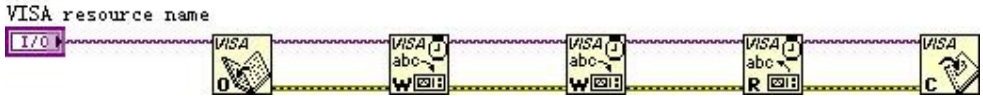
1. Open **Block Diagram**; choose **Instrument I/O→VISA**; then separately add four functions: **"VISA Open"**, **"VISA Read"**, **"VISA Write"** and **"VISA Close"**. See the figure below.



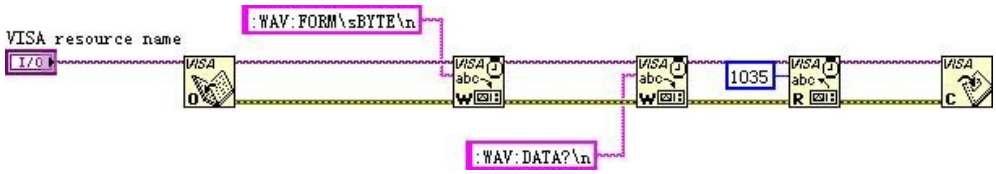
2. Move the mouse to the item of **"VISA resource name"** on the control of **"VISA Open"**; right-click the mouse to choose **Create→Control**. See the figure below.



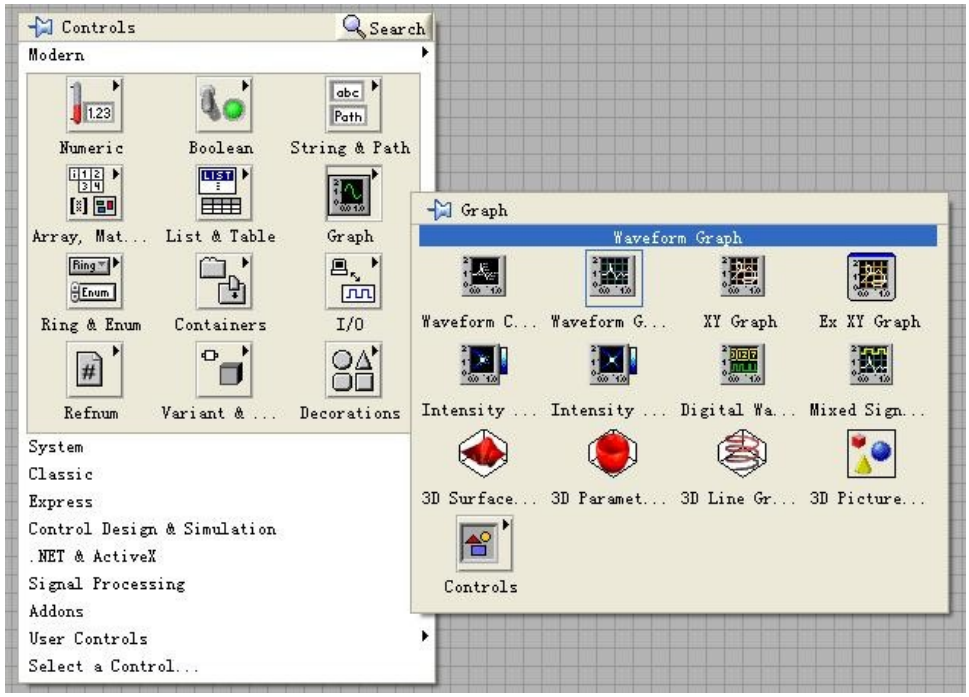
3. Separately connect **"VISA resource name"** with **"VISA resource name out"** and **"error out"** with **"error in"** of all the functions. See the figure below.



4. Add a textbox written with **":WAV:FORM\sBYTE\n"** to **"write buffer"** on one of the **"VISA Write"** control, and **":WAV:DATA?\n"** on the other one. The former is to set the format of waveform reading to be "BYTE", while the latter reads the waveform data shown on the screen.

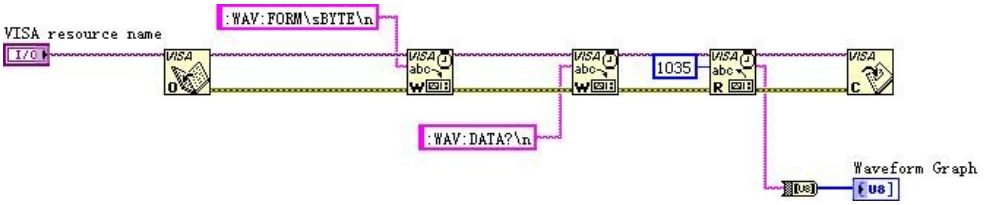


5. Open the **Front Panel**; choose **Modern→Graph→Waveform Graph** to add a **Waveform Graph** control. See the figure below.

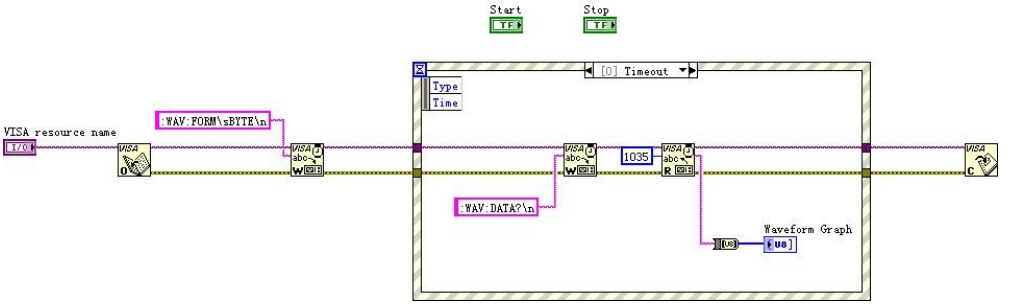


6. open **Block Diagram**; right-click and choose **Programming → String →**

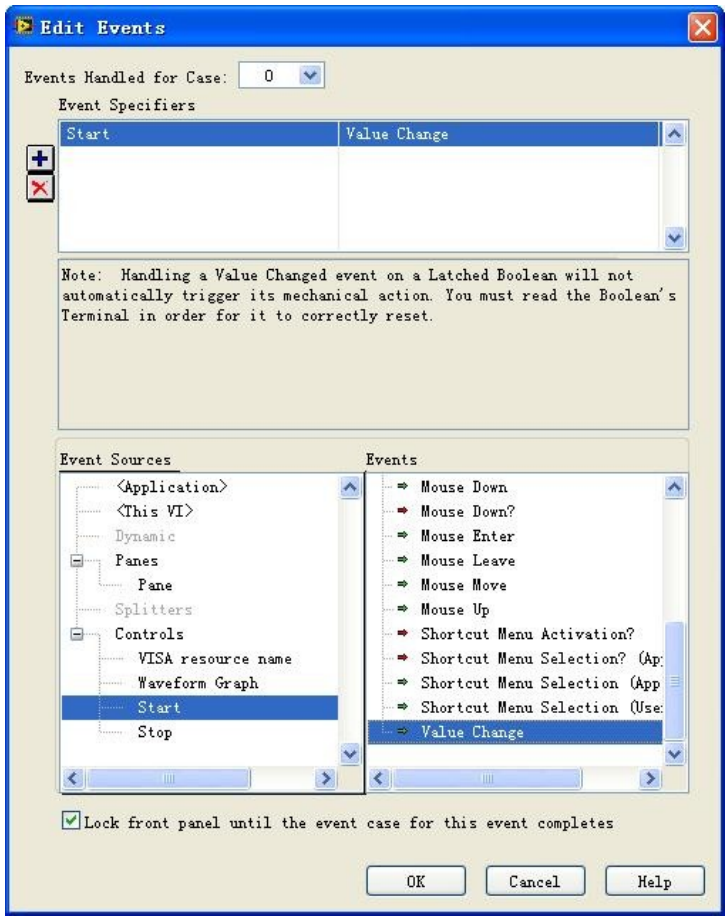
String/Array/Path and select "String To Byte Array"; then, use this function to connect **"read buffer"** on **"VISA Read"** function with the **Waveform Graph**. See the figure below.



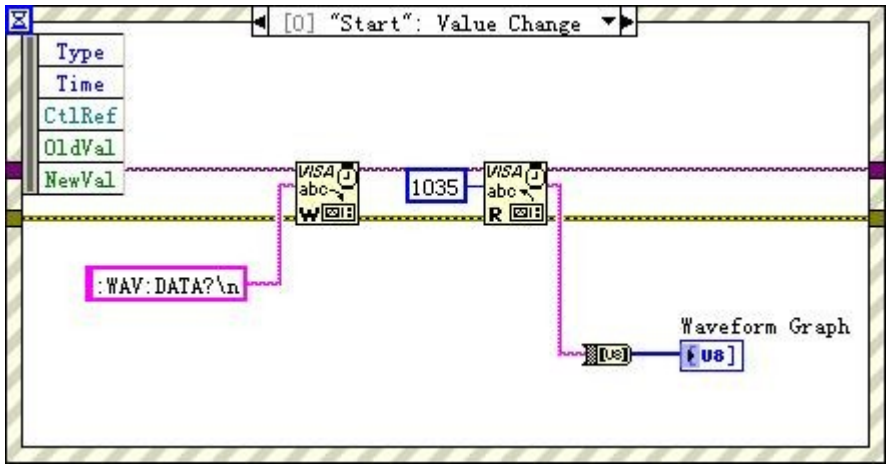
7. Add an **Event Structure** and a **While Loop** as well as two buttons, one of the buttons is used to control the start of waveform fetching, and the other one is to stop capturing. See the figure below.



8. Right-click the **"selector label"** and choose **"Edit Events Handled by This Case"** or **"Add Event case"** to add events respectively for each button. Press **"Start"** to capture waveform and **"Stop"** to exit the program.

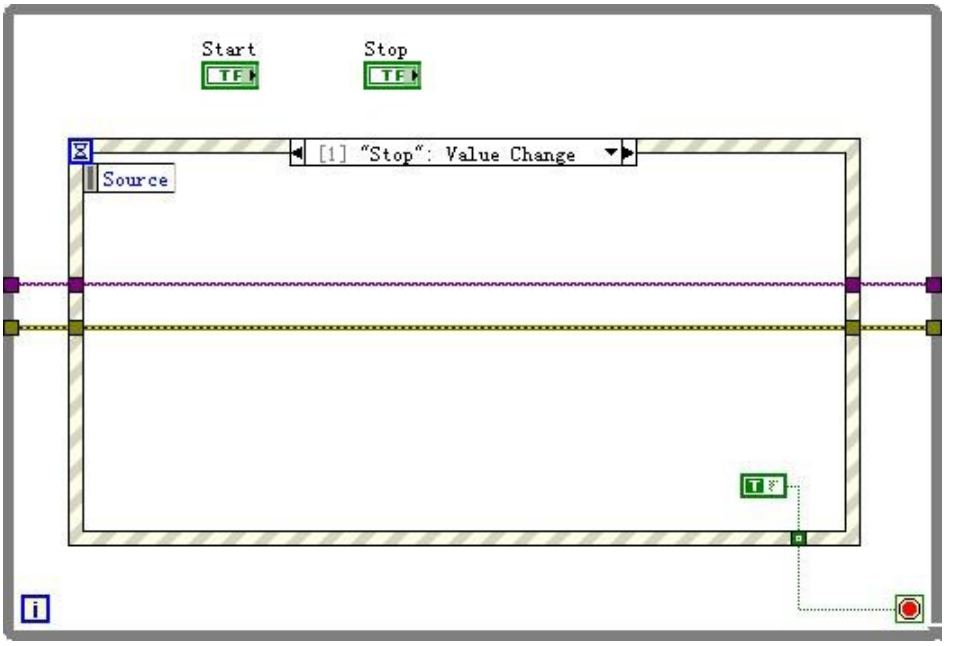


After you set the **"Start"** event, see the result below.

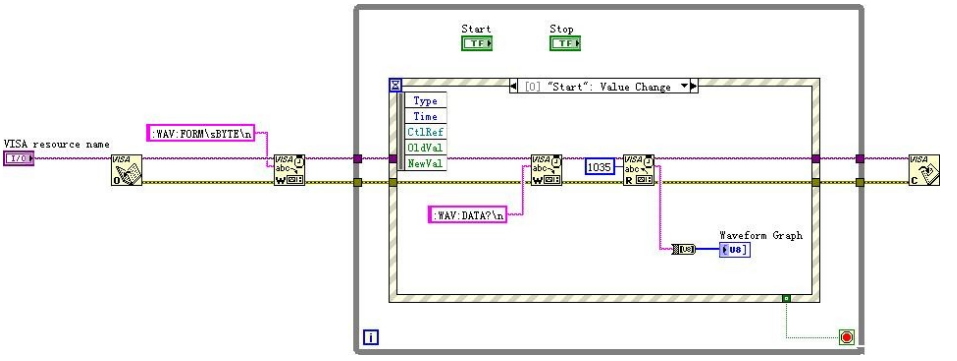


9. Add a **While Loop**; add **"Boolean"→"True Constant"** to point the event of

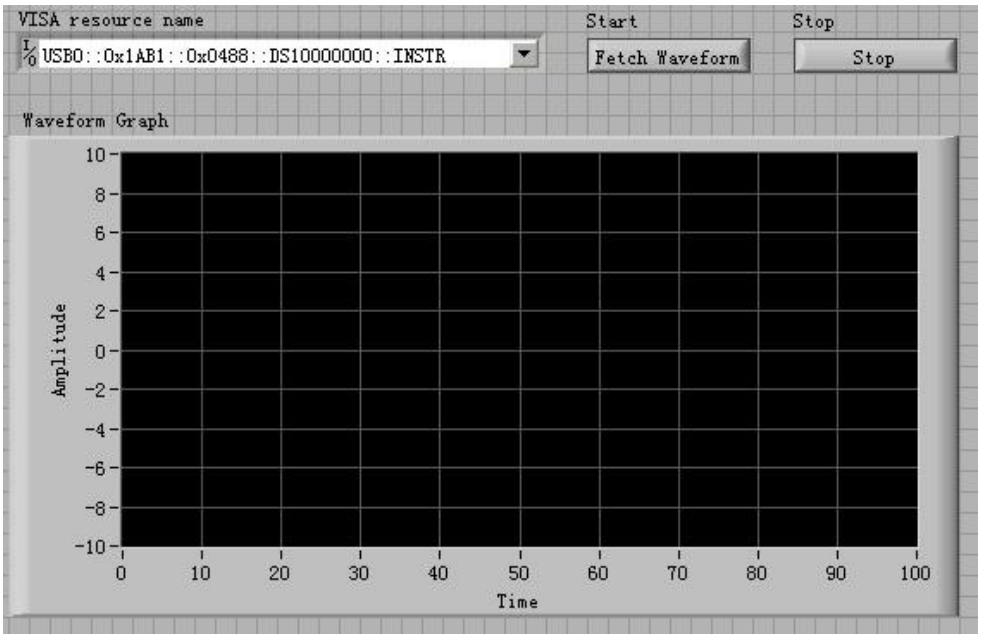
the **"Stop"** button to **While** and exit.



10. Change the input tunnel of VISA resource name and errors into **"Shift Register"** to finish creating program.



11. Adjust the style of **Front Panel** and click **"Fetch Waveform"** to get following interface. (the oscilloscope has been properly connected)


VISA resource name
USB0::0x1AB1::0x0488::DS10000000::INSTR
Start
Fetch Waveform
Stop
Stop
Waveform Graph
Amplitude
10
8
6
4
2
0
-2
-4
-6
-8
-10
0
10
20
30
40
50
60
70
80
90
100
Time

Appendix: Command Quick Reference A-Z

M

W