RIGOL

Programming Guide

DS1000CA Series Digital Oscilloscope

DS1302CA, DS1202CA, DS1102CA, DS1602CA

Sept. 2010

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.

Contents

Chapter 1 Programming Introduction

This chapter provides the introduction about the interfces and commands so as to control DS1000CA series digital oscilloscopes via remote commands.

The chapter contains following topics:

- Communication Interfaces
- Command Introduction
  Command Syntax
  Symbol Instruction
  Command Abbreviation
  Parameter Type

Communication Interfaces

Computers communicate with the oscilloscope by sending and receiving messages over an I/O port such as an USB, RS-232 port or GPIB. Commands appear as ASCII character strings embedded inside the output statements of a "host" language available on your computer.

Basic operations that you can do with a computer and an oscilloscope include:

- Set up the oscilloscope.
- Make measurements.
- Retrieve data (waveforms or measurements) from the oscilloscope.

Command Introduction

Command Syntax

The command system of DS1000CA series digital oscilloscopes presents a multistate tree structure, each of the subsystem consists of a "Root" keyword and one or multilayered keywords. Generally, a command line starts with ":" (except for IEEE commands) and ":" is also used for separating different keywords, meanwhile, parameters are permitted to follow the keywords; in addition, "?" after a command line denotes to query its function and "space" is used to divide command and parameter.

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

TRIGger is the root keyword of this command, **EDGE** and **SLOPe** are second and third keyword respectively, all of these keywords are separated by ":". Connects enclosed in the "{}" denote the parameters permitted to be set by user; queries are formed by adding a question mark (?) to the end of the commands and "space" is used to divide the command :TRIGger:EDGE:SLOPe and the parameter.

Symbol Instruction

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 enclosed in a command line must be selected. When several elements separated by a vertical line (|) are enclosed by braces { }, only one element may be selected, for example:

:MEASure:TOTal {ON|OFF},

Thereinto, {ON|OFF} indicates that only ON or OFF may be selected, not both.

2. Square Brackets [ ]

Items enclosed in square brackets [ ] are optional, for example:

:TIMebase[:DELayed]:OFFSet <offset>

Thereinto, [:DELayed] could be omitted.

3. Triangle Brackets < >

Items enclosed in < > should be replaced by an effective value, for example:

:DISPlay:BRIGhtness <ncount>

Thereinto, < ncount > must be a numerical value such as 25.

Command Abbreviation

All the commands of DS1000CA series are case-sensitive, so you can use any kind of them. As some commands are too long to memory and spell, we set the abbreviation for the commands in order to reduce your work and memory difficulty. Note that if use abbreviation, the capital letters specified in commands must be written completely.

For example:

:TRIGger:EDGE:SLOPe

can be entered as:

:TRIG:EDGE:SLOP

Parameter Type

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

1. Boolean

The parameter should be "OFF" or "ON", for example:
:MEASure:TOTal {ON|OFF}
Thereinto, "ON" denotes to trun on (enable) this function and "OFF" denoets to turn off (disable).

2. Consecutive Integer

The parameter should be a consecutive integer, for example:
:DISPlay:BRIGhtness <ncount>
Thereinto,<ncount> could be an integer between 0 and 32 (including 0 and 32).

3. Consecutive Real Number

The parameters could be any value within effective range and precision permiting, for example:
:TRIGger:EDGE:SENSitivity <count>
Thereinto, <count> could be a real number 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>
Thereinto, <count> could be 2, 4, 8, 16, 32, 64, 128, 256.

5. ASCII Character String

The parameter should a combination of ASCII character string, for example:
:TRIGger:MODE <mod>
Thereinto, <mod> could be "EDGE", "PULSe", "VIDEO", "SLOPe" or "ALTernation".

Chapter 2 Command Systems

This chapter gives detailed information on each command supported by DS1000CA series oscilloscopes, including command format, function explanation, using methods and considerations.

DS1000CA series include the following commands:

- General Commands
- SYSTem Commands
- ACQuire Commands
- DISPlay Commands
- TIMebase Commonds
- TRIGger Commands
  Trigger Control
  EDGE Trigger
  PULSe Trigger
  VIDEO Trigger
  SLOPe Trigger
  ALTernation Trigger
- STORage Commands
- MATH Commands
- CHANnel Commands
- MEASure Commands
- WAVeform Commands
- KEY Commands
- CURSor Commands
- Other Commands

General Commands

IEEE standard defineds common commands for querying or executing some basic information about instrument, which usually begins with "*" and holds 3-character long command keyword.

DS1000CA series digital oscilloscopes support the following general commands:

- *IDN?
- *RST

The detailed information of each command are given as follows:

1. *IDN?

Command Format:
*IDN?

Function Explanation:
This command is used to query the manufacturer, model, sequence number and the version number composed of numbers and separated by ".".

Returned Format:
RIGOL TECHNOLOGIES,<model>,<sequence_number>,<rev_number>

Example:
RIGOL TECHNOLOGIES,DS1302CA,DS1AA093200284,03.08.01

2. *RST

Command Format:
*RST

Function Explanation:
This command is used to reset the system parameters.

SYSTem Commands

SYSTem Commands are the fundamental commands for the operation of an oscilloscope. They can either be used for operational control or screen data interception and more.

System Commands include:

- :RUN
- :STOP
- :AUTO
- :HARDcopy

The detailed information of each command are given as follows:

1. :RUN

Command Format:
:RUN

Function Explanation:
This command initiates the oscilloscope to acquire waveform data. To stop acquisition, please execute **:STOP** command.

2. :STOP

Command Format:
:STOP

Function Explanation:
This command controls the oscilloscope to stop acquiring data. To restart the acquisition, please execute **:RUN** command.

3. :AUTO

Command Format:
:AUTO

Function Explanation:
This command is used to control the oscilloscope to evaluate all waveform characteristics and set the optimum conditions to display the waveforms,

4. :HARDcopy

Command Format:
:HARDcopy

Function Explanation:
This command is to extract the current information on the screen, and then copy the bitmap into the U disc in the form of "HardCopyxxx.bmp". (Note:This command is unavailable in file systems).

ACQuire Commands

ACQuire Commands are used to adjust the acquisition modes of the oscilloscope.

ACQuire Commands include:

- :ACQuire:TYPE
- :ACQuire:MODE
- :ACQuire:AVERages
- :ACQuire:SAMPlingrate?

The detailed information of each command are given as follows:

1. :ACQuire:TYPE

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

Function Explanation:
The commands are used to set and query the current acquisition type. The options of <type> are: "NORMal", "AVERage" and "PEAKdetec".

Returned Format:
Query returns "NORMAL", "AVERAGE" or "PEAKDETECT".

Example:

:ACQ:TYPE AVER	Set the acquisition type as "AVERage"
:ACQ:TYPE?	Query returns "AVERAGE"

2. :ACQuire:MODE

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

Function Explanation:
The commands are used to set and query the current acquisition mode. The options of <mode> are "RTIMe" and "ETIMe".

Returned Format:
Query returns "REAL_TIME" or "EQUAL_TIME".

Example:

:ACQ:MODE ETIM	Set the acquisition mode as "ETIMe"
:ACQ:MODE?	Query returns "EQUAL_TIME"

3. :ACQuire:AVERages

Command Format:

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

Function Explanation:
The commands are used to set and query the average acquisition time. The range of <count> is form 2 to 256 at 2 times the power of N.

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

Example:

:ACQ:AVER 16	Set average acquisition time as 16
:ACQ:AVER?	Query returns 16

4. :ACQuire:SAMPlingrate?

Query Format:
:ACQuire:SAMPlingrate? CHANnel<n>

Function Explanation:
The commands are used to query the sampling rate of the current channel. <n> could be 1 or 2 which means channel 1 or channel 2.

Returned Format:
Query returns the value of the sampling rate.

Example:

:ACQ:SAMP? CHAN2	Query the smpling rate of channel2
1.250e+08	Query returns 125Msa/s

DISPlay Commands

DISPlay Commands are used to adjust the display mode of oscilloscopes.

Display Commands include:

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

The detailed information of each command are given as follows:

1. :DISPlay:TYPE

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

Function Explanation:
The commands are used to set and query the display type between sampling points. The options of <type> are "VECTors" (vectir display) or "DOTS" (point display).

Returned Format:
Query returns "VECTORS" or "DOTS".

Example:

:DISP:TYPE VECT	Set the display type as "VECTors"
:DISP:TYPE?	Query returns "VECTORS"

2. :DISPlay:GRID

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

Function Explanation:
The commands are used to set and query the display grid. The options of <grid> are "FULL" (set the grid and coordinate on), "HALF" (set the grid on) or "NONE" (set the grid and coordinate off).

Returned Format:
Query returns "FULL", "HALF" or "NONE".

Example:

:DISP:GRID FULL	Set the grid state as "FULL"
:DISP:GRID?	Query returns "FULL"

3. :DISPlay:PERSist

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

Function Explanation:
The commands are used to set the waveform persist "ON" or "OFF".

Returned Format:
Query returns "ON" or "OFF".

Example:

:DISP:PERS ON	Set the waveform persist "ON"
:DISP:PERS?	Query returns "ON"

4. :DISPlay:MNUDisplay

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

Function Explanation:
The commands are used to set and query the display time of the menu. The options of <time> are: 1s, 2s, 5s, 10s, 20s or Infinite.

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

Example:

:DISP:MNUD 10	Set the display time of the menu as 10s
:DISP:MNUD?	Query returns 10s

5. :DISPlay:MNUStatus

Command Format:
:DISPlay:MNUStatus {ON|OFF}

:DISPlay:MNUStaus?

Function Explanation:
The commands are used to set and query the menu "ON" or "OFF".

Returned Format:
Query returns "ON" or "OFF".

Example:

:DISP:MNUS ON	Set the menu "ON"
:DISP:MNUS?	Query returns "ON"

6. :DISPlay:CLEar

Command Format:
:DISPlay:CLEar

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

7. :DISPlay:BRIGhtness

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

Function Explanation:
The commands are used to set and query the grid brightness of the display from 0 and 32.

Returned Format:
Query returns an integer between 0 and 32.

Example:

:DISP:BRIG 10	Set the grid brightness of the display as 10

:DISP:BRIG? Query returns 10

8. :DISPlay:INTensity

Command Format:

:DISPlay:INTensity <count>
:DISPlay:INTensity?

Function Explanation:

The commands are used to set and query the brightness of the waveform from 0~32.

Returned Format:

Query returns an integer between 0 and 32.

Example:

:DISP:INT 12 Set the brightness of the waveform as 12
:DISP:INT? Query returns 12

9. :DISPlay:SCReen

Command Format:

:DISPlay:SCReen <scr>
:DISPlay:SCReen?

Function Explanation:

The commands are used to set and query the screen display mode. The options of <scr> are "NORmal" or "INVerted".

Returned Format:

Query returns "NORMAL" or "INVERTED".

Example:

:DISP:SCR NORM Set the display mode as "NORmal"
:DISP:SCR? Query returns "NORMAL"

TIMebase Commonds

TIMebase Commands set the Horizontal Scale (time base) and the waveform horizontal position in the memory (trigger offset). The waveform will enlarge or shrink when the horizontal Scale changed.

TIMebase Commands include:

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

The detailed information of each command are given as follows:

1. :TIMebase:MODE

Command Format:

:TIMebase:MODE <mode>
:TIMebase:MODE?

Function Explanation:

The commands are used to set and query the time base mode. The options of <mode> are "MAIN" (main timebase) and "DELayed" (delayed scan).

Returned Format:

Query returns "MAIN" or "DELAYED".

Example:

:TIM:MODE MAIN	Set the horizontal timebase mode as "MAIN"
:TIM:MODE?	Query returns "MAIN"

2. :TIMebase[:DELayed]:OFFSet

Command Format:

:TIMebase[:DELayed]:OFFSet <offset>
:TIMebase[:DELayed]:OFFSet?

Function Explanation:

The commands are used to adjust the timebase offset in "MAIN" or "Delayed" mode.

In **Normal** mode, the range of <offset> is 1s~ momery depth/(2* sample rate);

In **STOP** mode, the range of <offset> is -500s~+500s;

In **SCAN** mode, the range of <offset> is -6*Scale~+6*Scale,"Scale" indicates the current horizontal scale, the default unit is s/div.

When "MAIN" is "ON", [: DELayed] should be omitted, the range of <offset> is: MainOffset+6*(DelayScale-MainScale)
~ MainOffset+6*(MainScale-DelayScale)

Returned Format:

Query returns the set value of <offset> in scientific numeric notation in s.

Example:

:TIM:MODE MAIN	Set the timebase as "MAIN"
:TIM:OFFS 1	Set the timebase offset as 1s
:TIM:OFFS?	Query returns 1.000e+00

3. :TIMebase[:DELayed]:SCALe

Command Format:

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

Function Explanation:

The commands are used to set and query the time base scale (s/div) in "MAIN" or "DELAYE" mode.
In **YT** mode, the range of <scale_val> is 1ns~50s;
In **ROLL** mode, the range of <scale_val> is 500ms~50s.
When "MAIN" is "ON", [: DELayed] needs to be omitted.

Returned Format:

Query returns the set value of < scale_val> in scientific numeric notation in s.

Example:

:TIM:MODE MAIN	Set the timebase as "MAIN"
:TIM:SCAL 2	Set the timebase scale as 2s
:TIM:SCAL?	Query returns 2.000e+00

4. :TIMebase:FORMat

Command Format:

:TIMebase:FORMat <value>
:TIMebase: FORMat?

Function Explanation:

The commands are used to set and query the timebase. The options of <value> are "XY", "YT" or "SCANning".

Returned Format:

Query returns "X-Y", "Y-T' or "SCANNING".

Example:

:TIM:FORM YT	Set the format of the timebase as "YT" mode
:TIM:FORM?	Query returns "Y-T"

TRIGger Commands

To make the waveform display more stable, trigger system needs to be set. The trigger determines when the oscilloscope starts to acquire data and display a waveform. When a trigger is set up properly, it converts unstable displays into meaningful waveforms.

When the oscilloscope starts to acquire a waveform, it collects enough data so that it can display the waveform to the left of the trigger point. The oscilloscope continues to acquire 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 to the right of the trigger point.

DS1000CA series oscilloscopes provide five trigger modes: Edge, Pulse, Video, Slope and Alternation.

TRIGger Commands include:

Trigger Control

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

EDGE Trigger

- :TRIGger:EDGE:SLOPe
- :TRIGger:EDGE:SENSitivity

PULSe Trigger

- :TRIGger:PULSe:MODE
- :TRIGger:PULSe:SENSitivity

- :TRIGger:PULSe:WIDTh

VIDEO Trigger

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

SLOPe Trigger

- :TRIGger:SLOPe:TIME
- :TRIGger:SLOPe:SENSitivity
- :TRIGger:SLOPe:MODE
- :TRIGger:SLOPe:WINDow
- :TRIGger:SLOPe:LEVelA
- :TRIGger:SLOPe:LEVelB

ALTernation Trigger

- :TRIGger:ALTernation:SOURce
- :TRIGger:ALTernation:TYPE
- :TRIGger:ALTernation:TimeSCALe
- :TRIGger:ALTernation:TimeOFFSet
- :TRIGger:ALTernation<mode>:LEVel
- :TRIGger:ALTernation:EDGE:SLOPe
- :TRIGger:ALTernation<mode>:MODE
- :TRIGger:ALTernation<mode>:TIME
- :TRIGger:ALTernation:VIDEO:POLarity
- :TRIGger:ALTernation:VIDEO:STANdard
- :TRIGger:ALTernation:VIDEO:LINE
- :TRIGger:ALTernation:SLOPe:WINDow
- :TRIGger:ALTernation:SLOPe:LEVelA
- :TRIGger:ALTernation:SLOPe:LEVelB
- :TRIGger:ALTernation<mode>:COUPling
- :TRIGger:ALTernation<mode>:HOLDoff
- :TRIGger:ALTernation<mode>:SENSitivity

The detailed information of each command are given as follows:

Trigger Control

1. :TRIGger:MODE

Command Format:

:TRIGger:MODE <mode>
:TRIGger:MODE?

Function Explanation:

The commands are used to set and query the trigger mode. The options of <mode> are "EDGE", "PULSe", "SLOPe", "VIDEO" or "ALTernation".

Returned Format:

Query returns "EDGE", "PULSE", "SLOPE", "VIDEO" or "ALTERNATION".

Example:

:TRIG:MODE EDGE	Set the trigger mode as "EDGE"
:TRIG:MODE?	Query returns "EDGE"

2. :TRIGger<mode>:SOURce

Command Format:

:TRIGger<mod>:SOURce <src>
:TRIGger<mod>:SOURce?

Function Explanation:

The commands are used to set and query the trigger source. The options of <src> are "CHANnel1", "CHANnel2", "EXT", "EXT/5" or "AC Line".
When <mode> is ":EDGE", <src> can be CHANnel<n>, EXT, EXT5 or ACLine;
When <mode> is ":PULSE", <src> can be CHANnel<n>, EXT or EXT5;
When <mode> is ":SLOPE", <src> can be CHANnel<n>, EXT or EXT5;
When <mode> is ":VIDEO", <src> can be CHANnel<n>, EXT or EXT5.
<n> is 1 or 2.

Returned Format:

Query returns "CH1", "CH2", "EXT", "EXT5" or "ACLINE".

Example:

:TRIG:EDGE:SOUR CHAN1	Set the source of edge trigger as "CHANnel1"
:TRIG:EDGE:SOUR?	Query returns "CH1"

3. :TRIGger<mode>:LEVel

Command Format:

:TRIGger<mode>:LEVel <level>
:TRIGger<mode>:LEVel?

Function Explanation:

The commands are used to set and query the voltage level. The options of <mode> are ":EDGE", ":PULSe" or ":VIDEO"; the range of <level> is -6*Scale~+6*Scale (away from screen center), "Scale" indicates the current vertical scale, the default unit of "level" is V/div.

Returned Format:

Query returns the set value of <level> in scientific numeric notation in V.

Example:

:TRIG:EDGE:LEV 2	Set the edge trigger level as 2V
:TRIG:EDGE:LEV?	Query returns 2.00e+00

4. :TRIGger<mode>:SWEep

Command Format:

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

Function Explanation:

The commands are used to set and query the trigger type as "AUTO", "NORMal" or "SIMGle". The options of <mode> are ":EDGE", ":PULSe", "VIDEO" or ":SLOPe".

Returned Format:

Query returns "AUTO", "NORMAL" or "SINGLE".

Example:

:TRIG:EDGE:SWE AUTO Set the edge trigger mode as "AUTO"

:TRIG:EDGE:SWE? Query returns "AUTO"

5. :TRIGger<mode>:COUPling

Command Format:

:TRIGger<mode>:COUPling {DC|AC|HF|LF}

:TRIGger<mode>:COUPling?

Function Explanation:

The commands are used to set and query the coupling mode as "DC", "AC", 'HF" or "LF". The options of <mode> are ":EDGE", ":PULSe" or ":SLOPe".

DC: Allow all signals pass

AC: Block DC signals

HF: Reject high frequency signals (Higher than 150kHz)

LF: Reject DC and low frequency signals (Lower than 8kHz)

Returned Format:

Query returns "DC", "AC", "HF" or "LF".

Example:

:TRIG:EDGE:COUP DC Set the coupling mode as "DC"

:TRIG:EDGE:COUP? Query returns "DC"

6. :TRIGger:HOLDoff

Command Format:

:TRIGger:HOLDoff <count>

:TRIGger:HOLDoff?

Function Explanation:

The commands are used to set and query trigger holdoff. The range of <count> is 500ns~1.5s.

Returned Format:

Query returns the set value of <count> in scientific numeric notation in s.

Example:

:TRIG:HOLD 0.0001	Set the time of holdoff as 100 us
:TRIG:HOLD?	Query returns 1.000e-04

7. :TRIGger:STATus?

Command Format:

:TRIGger:STATus?

Function Explanation:

The commands are used to query the current status of the oscilloscope. The status may be RUN, STOP, T'D, WAIT, SCAN or AUTO.

Returned Format:

Query returns RUN, STOP, T'D, WAIT or AUTO.

8. :Trig%50

Command Format:

:Trig%50

Function Explanation:

The command sets the trigger level to the vertical middle of the signal amplitude.

9. :FORCetrig

Command Format:

:FORCetrig

Function Explanation:

The command forces the oscilloscope to display a steady waveform when there is no suitable trigger condition, which is usually used in "Normal" and "Single" trigger mode.

EDGE Trigger

1. :TRIGger:EDGE:SLOPe

Command Format:

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

Function Explanation:

The commands are used to set and query the trigger edge as "POSitive", "NEGative" or "ALTernation".

Returned Format:

Query returns "POSITIVE", "NEGATIVE" or "ALTERNATION".

Example:

:TRIG:EDGE:SLOP POS	Set the trigger edge as "POSitive"
:TRIG:EDGE:SLOP?	Query returns "POSITIVE"

2. :TRIGger:EDGE:SENSitivity

Command Format:

:TRIGger:EDGE:SENSitivity <count>
:TRIGger:EDGE:SENSitivity?

Function Explanation:

The commands are used to set and query the trigger sensitivity of EDGE trigger. The range of <count> is 0.1~1.

Returned Format:

Query returns the set value of <count> in scientific numeric notation in div.

Example:

:TRIG:EDGE:SENS 0.5	Set the trigger sensitivity as 0.5div
:TRIG:EDGE:SENS?	Query returns 5.00e-01

PULSe Trigger

1. :TRIGger:PULSe:MODE

Command Format:

:TRIGger:PULSe:MODE <mode>

:TRIGger:PULSe:MODE?

Function Explanation:

The commands are used o set and query the pulse condition. The options of <mode> are "+GREaterthan", "+LESSthan", "+EQUal", "-GREaterthan", "-LESSthan" or "-EQUal".

Returned Format:

Query returns "+GREATER THAN", "+LESS THAN", "+EQUAL", "-GREATER THAN", "-LESS THAN" or "-EQUAL".

Example:

:TRIG:PULS:MODE +GRE Set the pulse mode as "+GREaterthan"

:TRIG:PULS:MODE? Query returns "+GREATER THAN"

2. :TRIGger:PULSe:SENSitivity

Command Format:

:TRIGger:PULSe:SENSitivity <count>

:TRIGger:PULSe:SENSitivity?

Function Explanation:

The commands are used to set and query the trigger sensitivity of PULSe tigger. The range of <count> is 0.1~1.

Returned Format:

Query returns the set value of <count> in scientific numeric notation in div.

Example:

:TRIG:PULS:SENS 0.5 Set the trigger sensitivity as 0.5div

:TRIG:PULS:SENS? Query returns 5.00e-01

3. :TRIGger:PULSe:WIDTh

Command Format:

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

Explanation:

The commands are used to set and query the pulse width from 20ns to 10s.

Returned Format:

Query returns the set value of <wid> in scientific numeric notation in s.

Example:

:TRIG:PULS:WIDT 0.001 Set the pulse width as 1ms
:TRIG:PULS:WIDT? Query returns 1.000e-03

VIDEO Trigger

1. :TRIGger:VIDEO:MODE

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

Function Explanation:
The commands are used to set and query the trigger synchronization. The options of <mode> are "ODDfield", "EVENfield", "LINE" or "ALLlines".

Returned Format:
Query returns "ODD FILED", "EVEN FILED", "LINE" or "ALL LINES".

Example:
:TRIG:VIDEO:MODE EVEN Set the trigger synchronization as "EVENfield".
:TRIG:VIDEO:MODE? Query returns "EVEN FIELD".

2. :TRIGger:VIDEO:POLarity

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

Function Explanation:
The commands are used to set and query the Video polarity as "POSitive" or "NEGative".

Returned Format:
Query returns "POSITIVE" or "NEGATIVE".

Example:
:TRIG:VIDEO:POL POS Set the Video polarity as "POSitive"
:TRIG:VIDEO:POL? Query returns "POSITIVE"

3. :TRIGger:VIDEO:STANdard

Command Format:

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

Function Explanation:

The commands are used to set and query the video type as "NTSC"or "PALSecam".

Returned Format:

Query returns "NTSC" or "PAL/SECAM".

Example:

:TRIG:VIDEO:STAN PALS	Set the video type as "PALSecam"
:TRIG:VIDEO:STAN?	Query returns "PAL/SECAM"

4. :TRIGger:VIDEO:LINE

Command Format:

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

Function Explanation:

The commands are used to set and query the number of synchronous appointed lines.
In NTSC standard, the range of <value> is 1~525;
In PALSecam standard, the range of <value> is 1~625.

Returned Format:

Query returns the set value of lines.

Example:

:TRIG:VIDEO:LINE 25	Set number of synchronous appointed lines as 25
:TRIG:VIDEO:LINE?	Query returns 25

5. :TRIGger:VIDEO:SENSitivity

Command Format:

:TRIGger:VIDEO:SENSitivity <count>

:TRIGger:VIDEO:SENSitivity?

Function Explanation:

The commands are used to set and query the trigger sensitivity of VIDEO trigger. The range of <count> is 0.1~1.

Returned Format:

Query returns the set value of <count> in scientific numeric notation in div.

Example:

:TRIG:VIDEO:SENS 0.5	Set the trigger sensitivity as 0.5div
:TRIG:VIDEO:SENS?	Query returns 5.00e-01

SLOPe Trigger

1. :TRIGger:SLOPe:TIME

Command Format:

:TRIGger:SLOPe:TIME <count>
:TRIGger:SLOPe:TIME?

Function Explanation:

The commands are used to set and query the slope time from 20ns~10s.

Returned Format:

Query returns the set value of <count> in scientific numeric notation in s.

Example:

:TRIG:SLOP:TIME 0.01	Set the slope time as 10ms
:TRIG:SLOP:TIME?	Query returns 1.000e-02

2. :TRIGger:SLOPe:SENSitivity

Command Format:

:TRIGger:SLOPe:SENSitivity <count>
:TRIGger:SLOPe:SENSitivity?

Function Explanation:

The commands are used to set and query the trigger sensitivity of SLOPe trigger. The range of <count> is 0.1~1.

Returned Format:

Query returns the set value of <count> in scientific numeric notation in div.

Example:

:TRIG:SLOP:SENS 0.5	Set the trigger sensitivity as 0.5div
:TRIG:SLOP:SENS?	Query returns 5.00e-01

3. :TRIGger:SLOPe:MODE

Command Format:

:TRIGger:SLOPe:MODE <mode>

:TRIGger: SLOPe: MODE?

Function Explanation:

The commands are used to set and query the slope condition. The options of <mode> are "+GREaterthan", "+LESSthan", "+EQUal", "-GREaterthan", "-LESSthan" or "-EQUal".

Returned Format:

Query returns "+GREATER THAN", "+LESS THAN", "+EQUAL", "-GREATER THAN", "-LESS THAN" or "-EQUAL".

Example:

:TRIG:SLOP:MODE +GRE Set the slope condition as "+Greaterthan"

:TRIG:SLOP:MODE? Query Returns "+GREATER THAN"

4. :TRIGger:SLOPe:WINDow

Command Format:

:TRIGger:SLOPe:WINDow <count>

:TRIGger:SLOPe:WINDow?

Function Explanation:

The commands are used to set and query the trigger level of the vertical window.

The options of <count> are "PA", "PB" or "PAB" when slope condition is "+GREaterthan", "+LESSthan" or "+EQUal";

The options of <count> are "NA", "NB" or "NAB" when slope condition is "-GREaterthan", "-LESSthan" or "-EQUal";

Returned Format:

Query returns "P_WIN_A", "P_WIN_B", "P_WIN_AB", "N_WIN_A", "N_WIN_B" or "N_WIN_AB".

Example:

:TRIG:SLOP:WIND PA	Set the border of the vertical window as "PA"
:TRIG:SLOP:WIND?	Query returns "P_WIN_A"

5. :TRIGger:SLOPe:LEVelA

Command Format:

:TRIGger:SLOPe:LEVelA <value>
:TRIGger:SLOPe:LEVelA?

Function Explanation:

The commands are to set and query the upper trigger level A.The range of <value> is -6*Scale~+6*Scale. "Scale" indicates the current vertical scale, the default unit of "level" is V/div.

Returned Format:

Query returns the set value of <value> in scientific numeric notation in V.

Example:

:TRIG:SLOP:LEVA 2	Set the upper trigger level as 2V
:TRIG:SLOP:LEVA?	Query returns 2.000e+00

6. :TRIGger:SLOPe:LEVelB

Command Format:

:TRIGger:SLOPe:LEVelB <value>
:TRIGger:SLOPe:LEVelB?

Function Explanation:

The commands are used to set and query the lower trigger Level B. The range of <value> is -6*Scale~+6*Scale. "Scale" indicates the current vertical scale, the default unit of "level" is V/div.

Returned Format:

Query returns the set value of <value> in scientific numeric notation in V.
Note: Minimum of Level A couldn't be less than maximum of Level B.

Example:

:TRIG:SLOP:LEVB -1.5	Set the lower trigger level as -1.5V
:TRIG:SLOP:LEVB?	Query returns -1.500e+00

ALTernation Trigger

1. :TRIGger:ALTernation:SOURce

Command Format:

:TRIGger:ALTernation:SOURce <src>
:TRIGger:ALTernation:SOURce?

Function Explanation:

The commands are used to set and query the source to be operated. The options of <src> are "CHANnel1" or "CHANnel2".

Returned Format:

Query returns "CH1" or "CH2".

Example:

:TRIG:ALT:SOUR CHAN2	Set the source as "CHANnel2"
:TRIG:ALT:SOUR?	Query returns "CH2"

2. :TRIGger:ALTernation:TYPE

Command Format:

:TRIGger:ALTernation:TYPE <value>
:TRIGger:ALTernation:TYPE?

Function Explanation:

The commands are used to set and query trigger type. The options of <type> are "EDGE", "PULSe", "SLOPe" or "VIDEO".

Returned Format:

Query returns "EDGE", "PULSE", "SLOPE" or "VIDEO".

Example:

:TRIG:ALT:TYPE EDGE	Set trigger type as "EDGE"
:TRIG:ALT:TYPE?	Query returns "EDGE"

3. :TRIGger:ALTernation:TimeSCALe

Command Format:

:TRIGger:ALTernation:TimeSCALe <value>
:TRIGger:ALTernation:TimeSCALe?

Function Explanation:

The commands are used to set and query time base of the current channel. The range of <value> is 2ns - 50s.

Returned Format:

Query returns the set value of <value> in scientific numeric notation in s.

Example:

:TRIG:ALT:TSCAL 0.001	Set the time base as 1ms
:TRIG:ALT:TSCAL?	Query returns 1.000e-03

4. :TRIGger:ALTernation:TimeOFFSet

Command Format:

:TRIGger:ALTernation:TimeOFFSet <value>
:TRIGgervALTernation:TimeOFFSet?

Function Explanation:

The commands are used to set and query time offset for the current channel.
In **NOMAL** mode, the range of <value> is 1s~ memory terminal;
In **STOP** mode, the range of <value> is -500s~+500s;
In **ROLL** mode, the range of <value> is -6*scale~+6*Scale, "Scale" indicates the current horizontal scale, the default unit is s/div.

Returned Format:

Query returns the set value of <value> in scientific numeric notation in s.

Example:

:TRIG:ALT:TOFFS 0.0002	Set up current horizontal time base as 200us
:TRIG:ALT:TOFFS?	Query returns 2.000e-04

5. :TRIGger:ALTernation<mode>:LEVel

Command Format:

:TRIGger:ALTernation<mode>:LEVel <value>
:TRIGger:ALTernation<mode>:LEVel?

Function Explanation:

The commands are used to set and query the trigger level of current channel. The options of <mode> are ":EDGE", ":PULSe" or ":VIDEO". The range of <value> is -6*Scale~+6*Scale. "Scale" indicates the current vertical scale, the default unit is V/div.

Returned Format:

Query returns the set value of <value> in scientific numeric notation in V.

Example:

:TRIG:ALT:EDGE:LEV 2	Set the trigger level of current channel as 2V
:TRIG:ALT:EDGE:LEV?	Query returns 2.00e+00

6. :TRIGger:ALTernation:EDGE:SLOPe

Command Format:

:TRIGger:ALTernation:EDGE:SLOPe <value>
:TRIGger:ALTernation:EDGE:SLOPe?

Function Explanation:

The commands are used to set and query the edge polarity of EDGE trigger in current channel. The options of <value> are "POSitive", "NEGative" or "ALTernation".

Returned Format:

Query returns "POSITIVE", "NEGATIVE" or "ALTernation".

Example:

:TRIG:ALT:EDGE:SLOP POS	Set the edge polarity as "POSitive"
:TRIG:ALT:EDGE:SLOP?	Query returns "POSITIVE"

7. :TRIGger:ALTernation<mode>:MODE

Command Format:

:TRIGger:ALTernation<mode>:MODE <value>

:TRIGger:ALTernation<mode>:MODE?

Function Explanation:

The commands are used to set and query pulse condition, slope condition or synchronous settings. The options of <mode> are ":PULSe", ":VIDEO" or ":SLOPe".

In ":PULSe" or ":SLOPe" mode, <value> can be "+GREaterthan","+LESSthan", "+EQUal", "-GREaterthan", "-LESSthan" or "-EQUal".

In ":VIDEO" mode, <value> can be "ODDfield", "EVENfield", "LINE" or "ALLlines".

Returned Format:

Query Returns the set value of <value>.

Example:

:TRIG:ALT:PULS:MODE +GRE Set the pulse condition as "+GREaterthan"

:TRIG:ALT:PULS:MODE? Query returns "+GREATER THAN"

8. :TRIGger:ALTernation<mode>:TIME

Command Format:

:TRIGger:ALTernation<mode>:TIME <value>

:TRIGger:ALTernation<mode>:TIME?

Function Explanation:

The commands are used to set and query pulse width or slope time. The options of <mode> are ":SLOPe" or ":PULSe". The range of <value> is 2ns~10s.

Returned Format:

Query returns the set value of <value> in scientific numeric notation in s.

Example:

:TRIG:ALT:SLOP:TIME 0.002 Set the slope time as 2ms
:TRIG:ALT:SLOP:TIME? Query returns 2.000e-03

9. :TRIGger:ALTernation:VIDEO:POLarity

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

Function Explanation:
The commands are used to set the video polarity as "POSitive" or "NEGative".

Returned Format:
Query returns "POSITIVE" or "NEGATIVE".

Example:
:TRIG:ALT:VIDEO:POL POS Set the video polarity as "POSitive"
:TRIG:ALT:VIDEO:POL? Query returns "POSITIVE"

10. :TRIGger:ALTernation:VIDEO:STANdard

Command Format:
:TRIGger:ALTernation:VIDEO:STANdard {NTSC|PALSecam}
:TRIGger:ALTernation:VIDEO:STANdard?

Function Explanation:
The commands are used to set the video standard as "NTSC" or "PALSecam".

Returned Format:
Query returns "NTSC" or "PAL/SECAM".

Example:
:TRIG:ALT:VIDEO:STAN NTSC Set the video standard as "NTSC"
:TRIG:ALT:VIDEO:STAN? Query returns "NTSC"

11. :TRIGger:ALTernation:VIDEO:LINE

Command Format:

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

Function Explanation:

The commands are used to set and query the number of appointed synchronous lines.
In NTSC standard, the range of <value> is 1~525;
In PALSecam standard, the range of <value> is 1~ 625.

Returned Format:

Query returns the set value of <value>.

Example:

:TRIG:ALT:VIDEO:LINE 100 Set the number as 100
:TRIG:ALT:VIDEO:LINE? Query returns 100

12. :TRIGger:ALTernation:SLOPe:WINDow

Command Format:

:TRIGger:ALTernation:SLOPe:WINDow <count>
:TRIGger:ALTernation:SLOPe:WINDow?

Function Explanation:

The commands are used to set and query trigger level of the vertical window.
The options of <count> are "PA", "PB" or "PAB" when slope condition is "+GREaterthan", "+LESSthan" or "+EQUal";
The options of <count> are "NA", "NB" or "NAB" when slope condition is "-GREaterthan", "-LESSthan" or "-EQUal".

Returned Format:

Query returns "P_WIN_A", "P_WIN_B", "P_WIN_AB", "N_WIN_A", "N_WIN_B" or "N_WIN_AB".

Example:

:TRIG:ALT:SLOP:WIND PA Set the border of the vertical window as "PA"
:TRIG:ALT:SLOP:WIND? Query returns "P_WIN_A"

13. :TRIGger:ALTernation:SLOPe:LEVelA

Command Format:
:TRIGger:ALTernation:SLOPe:LEVelA <value>
:TRIGger:ALTernation:SLOPe:LEVelA?

Function Explanation:
The commands are used to set and query the upper trigger level A of Slope trigger. The range of <value> is -6*Scale~+6*Scale. "Scale" indicates the current vertical scale, the default unit is V/div.

Returned Format:
Query returns the set value of <value> in scientific numeric notation in V.

Example:
:TRIG:ALT:SLOP:LEVA 2 Set the trigger level A as 2V
:TRIG:ALT:SLOP:LEVA? Query returns 2.000e+00

14. :TRIGger:ALTernation:SLOPe:LEVelB

Command Format:
:TRIGger:ALTernation:SLOPe:LEVelB <value>
:TRIGger:ALTernation:SLOPe:LEVelB?

Function Explanation:
The commands are used to set and query the lower trigger level B of Slope trigger. The range of <value> is -6*Scale~+6*Scale. "Scale" indicates the current vertical scale, the default unit is V/div.
Note: The minimum of Level A couldn't be less than maximum of Level B.

Returned Format:
Query returns the set value of <value> in scientific numeric notation in V.

Example:

:TRIG:ALT:SLOP:LEVB -1.5	Set Level B as -1.5V
:TRIG:ALT:SLOP:LEVB?	Query returns -1.500e+00

15. :TRIGger:ALTernation<mode>:COUPling

Command Format:

:TRIGger:ALTernation<mode>:COUPling{DC|AC|HF|LF}

:TRIGger<mode>:COUPling?

Function Explanation:

The commands are used to set and query the coupling mode as "DC", "AC", 'HF" or "LF".

DC: Allow all signals pass

AC: Block DC signals

HF: Reject high frequency signals (Higher than 150kHz)

LF: Reject DC and low frequency signals (Lower than 8kHz)

The options of <mode> are ":EDGE", ":PULSe" or ":SLOPe".

Returned Format:

Query returns "DC", "AC", 'HF" or "LF".

Example:

:TRIG:ALT:EDGE:COUP DC	Set the coupling condition as "DC"
:TRIG:ALT:EDGE:COUP?	Query returns "DC"

16. :TRIGger:ALTernation<mode>:HOLDoff

Command Format:

:TRIGger:ALTernation<mode>:HOLDoff <count>

:TRIGger:ALTernation<mode>:HOLDoff?

Function Explanation:

The commands are used to set and query the trigger holdoff. The options of <mode> are :EDGE, :PULSe, :SLOPe or :VIDEO. The range of <count> is 500ns~1.5s.

Returned Format:

Query returns the set value of <count> in scientific numeric notation in s.

Example:

:TRIG:ALT:PULS:HOLD 1 Set the trigger holdoff as 1s
:TRIG:ALT:PULS:HOLD? Query returns 1.000e+00

17. :TRIGger:ALTernation<mode>:SENSitivity

Command Format:

:TRIGger:ALTernation<mode>:SENSitivity <count>
:TRIGger:ALTernation<mode>:SLOPe:SENSitivity?

Function Explanation:

The commands are used to set and query the trigger sensitivity of ALTernation trigger. The options of <mode> are ":EDGE", ":PULSe", ":SLOPe" or ":VIDEO". The range of <count> is 0.1~1.

Returned Format:

Query returns the set value of <count> in scientific numeric notation in div.

Example:

:TRIG:ALT:EDGE:SENS 0.5 Set the trigger sensitivity as 0.5div
:TRIG:ALT:EDGE:SLOP:SENS? Query returns 5.00e-01

STORage Commands

STORage Command is used to recall the factory settings.

STORage Command includes:

- :STORage:FACTory:LOAD

The detailed information of this command is given below:

1. :STORage:FACTory:LOAD

Command Format:

:STORage:FACTory:LOAD

Function Explanation:

The command is used to recall the factory settings.

MATH Commands

MATH Commands are used to excute adding, subtracting, multiplying and FFT operaion from channels and display the results that also can be measured by grid or cursor.

MATH Commands include:

- :MATH:DISPlay
- :MATH:OPERate
- :FFT:DISPlay

The detailed information of each command are given as follows:

1. :MATH:DISPlay

Command Format:

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

Function Explanation:

The commands are used to turn the **MATH** function "ON" or "OFF".

Returned Format:

Query returns "ON" or "OFF".

Example:

:MATH:DISP ON	Set the math function "ON"
:MATH:DISP?	Query returns "ON"

2. :MATH:OPERate

Command Format:

:MATH:OPERate <operate>
:MATH:OPERate?

Function Explanation:

The commands are used to set and query type of math operation for the current channel. The options of <operate> are A+B, A-B, AB or FFT.

Returned Format:

Query returns A+B, A-B, A*B or FFT.

Example:

:MATH:OPER AB	Set the type of math operation as AB
:MATH:OPER?	Query returns A*B

3. :FFT:DISPlay

Command Format:

:FFT:DISPlay {ON|OFF}

:FFT:DISPlay?

Function Explanation:

The commands are used to set and query the FFT function ”ON” or “OFF”.

Returned Format:

Query returns “ON” or “OFF”.

Example:

:FFT:DISP ON	Set the FFT function on
:FFT:DISP?	Query returns “ON”

CHANnel Commands

Both channels of the instrument can be controlled by CHANnel Commands. Each channel has an independent vertical menu and each option could be set 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>:VERNier
- :CHANnel<n>:INPut

The detailed information of each command are given as follows:

1. :CHANnel<n>:BWLimit

Command Format:

:CHANnel<n>:BWLimit {ON|OFF}
:CHANnel<n>:BWLimit?

Function Explanation:

The commands are used to set and query the bandwidth limit "ON" or "OFF". <n> is 1 or 2.

Returned Format:

Query returns "ON" or "OFF".

Example:

:CHAN2:BWL 0FF	Set Channel 2 bandwidth limit "OFF"
:CHAN2:BWL?	Query returns "OFF"

2. :CHANnel<n>:COUPling

Command Format:

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

Function Explanation:

The commands are used to set and query the coupling as "DC", "AC" or "GND". <n> is 1 or 2.

Returned Format:

Query returns "DC", "AC" or "GND".

Example:

:CHAN2:COUP DC	Set the coupling type of Channel 2 as "DC"
:CHAN2:COUP?	Query returns "DC"

3. :CHANnel<n>:DISPlay

Command Format:

:CHANnel<n>:DISPlay {ON|OFF}
:CHANnel<n>:DISPlay?

Function Explanation:
The commands are used to set and query channels "ON" or "OFF".
<n> is 1 or 2.

Returned Format:
Query returns 1/0 Thereinto, "1" indicates the channel is open, "0" indicates the channel is closed.

Example:

:CHAN2:DISP ON	Set the display of channel 2 on
:CHAN2:DISP?	Query returns "ON"

4. :CHANnel<n>:INVert

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

Function Explanation:
The commands are used to set and query the waveform invert "ON" or "OFF".

Returned Format:
Query returns "ON" or "OFF".

Example:

:CHAN2:INV OFF	Set the waveform invert of channel 2 "OFF"
:CHAN2:INV?	Query returns "OFF".

5. :CHANnel<n>:OFFSet

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

Function Explanation:

The commands are used to set and query the vertical offset of the waveform. <n> is 1 or 2.

When Scale>200mV, the range of <offset> is -40V~+40V;

When Scale≤200mV, the range of <offset> is -800mV~+800mV.

Returned Format:

Query returns the value of <offset> in V.

Example:

:CHAN2:OFFS 20	Set the vertical offset of channel 2 as 20V
:CHAN2:OFFS?	Query returns 2.000e+01

6. :CHANnel<n>:PROBe

Command Format:

:CHANnel<n>:PROBe <attn>

:CHANnel<n>:PROBe?

Function Explanation:

The commands are used to set and query the attenuation factor of the probe to keep the gauge exact. The options of <attn> are 1, 5, 10, 50, 100, 500 or 1000. <n> is 1 or 2.

Returned Format:

Query returns the set value of <attn> in scientific numeric notation.

Example:

:CHAN2:PROB 10	Set probe attenuation of channel 2 as 10X
:CHAN2:PROB?	Query returns 1.000e+01

7. :CHANnel<n>:SCALe

Command Format:

:CHANnel<n>:SCALe <range>

:CHANnel<n>:SCALe?

Function Explanation:

The commands are used to set and query the vertical range of the amplified waveform. <n> is 1 or 2.
When Probe is 1X, the <range> is 1mV~10V;
When Probe is 5X, the <range> is 5mV~50V;
When Probe is 10X, the <range> is 10mV~100V;
When Probe is 50X, the <range> is 50mV~500V;
When Probe is 100X, the <range> is 100mV~1000V;
When Probe is 500X, the <range> is 500mV~5000V;
When Probe is 1000X, the <range> is 1V~10000V.

Returned Format:

Query returns the set value of <range> in scientific numeric notation in V.

Example:

:CHAN2:PROB 10	Set probe attenuation as 10X
:CHAN2:SCAL 20	Set vertical scale as 20V
:CHAN2:SCAL?	Query returns 2.000e+01

8. :CHANnel<n>:FILTer

Command Format:

:CHANnel<n>:FILTer {ON|OFF}
:CHANnel<n>:FilTer?

Function Explanation:

The commands are used to set and query the digital filter "ON" or "OFF". <n> is 1 or 2.

Returned Format:

Query returns "ON" or "OFF".

Example:

:CHAN2:FILT OFF	Set the digital filter of channel 2 off
:CHAN2:FILT?	Query returns "OFF"

9. :CHANnel<n>:VERNier

Command Format:

:CHANnel<n>:VERNier {ON|OFF}
:CHANnel<n>:VERNier?

Function Explanation:

The commands are used to set and query the adjustment mode of the vertical scaling.

When the setting is "OFF", the scale adjustment manner is "Coarse" and vertical sensitivity will be adjusted in 1-2-5 regulation.

When the setting is "ON", the manner is "Fine". The fine adjustment will subdivide during the "Coarse" range to improve the vertical resolution.

Returned Format:

Query returns "Coarse" or "Fine".

Example:

:CHAN2:VERN ON	Set the adjustment mode as "Fine"
:CHAN2:VERN?	Query returns "Fine"

10. :CHANnel<n>:INPut

Command Format:

:CHANnel<n>:INPut {50OHM|1MOHM}
:CHANnel<n>:INPut?

Function Explanation:

The commands are used to set and query the input impedance of CH1 or CH2. <n> is 1 or 2.

Returned Format:

Query returns "50ohm" or "1Mohm".

Example:

:CHAN2: INP 50OHM	Set the input impedance of channel2 as 50Ω
:CHAN2: INP?	Query returns "50ohm"

Note: This command is effective only for DS1202CA and DS1302CA .

MEASure Commands

MEASure Commands are the fundamental measurement operations and the measurement results are usually expressed in scientific numeric 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:TOTal
- :MEASure:SOURce

The detailed information of each command are given as follows:

1. :MEASure:CLEar

Command Format:
:MEASure:CLEar

Function Explanation:
This command is used to clear the current value.

2. :MEASure:VPP?

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

Function Explanation:
This command is used to query the Peak-Peak value of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as 6.48e+03, unit is V.

3. :MEASure:VMAX?

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

Function Explanation:
This command is used to query the maximum value of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as 3.24e+03, unit is V.

4. :MEASure:VMIN?

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

Function Explanation:
This command is used to query the minimum value for the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as -3.24e+03, unit is V.

5. :MEASure:VAMPlitude?

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

Function Explanation:
This command is used to query the vamplitude value of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as 6.48e+03, unit is V.

6. :MEASure:VTOP?

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

Function Explanation:
This command is used to query the top value of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as 3.24e+03, unit is V.

7. :MEASure:VBASe?

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

Function Explanation:
This command is used to query the base value of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as -3.24e+03, unit is V.

8. :MEASure:VAVerage?

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

Function Explanation:
This command is used to query the average value of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as 2.67e-01, unit is V.

9. :MEASure:VRMS?

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

Function Explanation:
This command is used to query the root-mean-square value of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as 1.36e+06, unit is V.

10. :MEASure:OVERshoot?

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

Function Explanation:

This command is used to query the overshoot value of the waveform. Generally, signal overshoot is in terms of percentage, and the transformational relation of the overshoot value and percentage is: overshoot*100. For example: overshoot = 8.00e3 mens this signal overshoot is 0.8%.

The options of <source> are CHANnel1 or CHANnel2.

Returned Format:

Such as 8.00e-03.

11. :MEASure:PREShoot?

Command Format:

:MEASure:PREShoot? [<source>]

Function Explanation:

This command is used to query the preshoot value of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:

Such as 8.00e-03.

12. :MEASure:FREQuency?

Command Format:

:MEASure:FREQuency? [<source>]

Function Explanation:

This command is used to query the waveform frequency. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:

Such as 1.00e+03, the unit is Hz.

13. :MEASure:RISetime?

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

Function Explanation:
This command is used to query the rise time of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as 3.10e-04, the unit is s.

14. :MEASure:FALLtime?

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

Function Explanation:
This command is used to query the fall time of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as 3.10e-04, the unit is s.

15. :MEASure:PERiod?

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

Function Explanation:
This command is used to query the waveform period. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as 1.00e-03, the unit is s.

16. :MEASure:PWIDth?

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

Function Explanation:
This command is used to query the positive pulse width of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as 5.00e-04, the unit is s.

17. :MEASure:NWIDth?

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

Function Explanation:
This command is used to query the negative pulse width of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as 5.00e-04, the unit is s.

18. :MEASure:PDUTycycle?

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

Function Explanation:
This command is used to query the positive duty cycle of the waveform. Generally, duty cycle is in terms of percentage, and the transformational relation of the overshoot value and percentage is: PDUT*100. For example: PDUT = 5.00e-1 mens this duty cycle is 50%.

Returned Format:
Such as 5.00e-01.

19. :MEASure:NDUTycycle?

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

Function Explanation:
This command is used to query the negative duty cycle of the waveform. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as 5.00e-01.

20. :MEASure: PDELay?

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

Function Explanation:
This command is used to query the delay of channel 1 or channel 2 relative to the positive edge. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as <-1.00e-04.

21. :MEASure:NDELay?

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

Function Explanation:
This command is used to query the the delay of channel 1 or channel 2 relative to the negative edge. The options of <source> are CHANnel1 or CHANnel2.

Returned Format:
Such as <-1.00e-04.

22. :MEASure:TOTal

Command Format:

:MEASure:TOTal {ON|OFF}
:MEASure:TOTal?

Function Explanation:

The commands are used to set and query the On/Off state of all measurement functions.

Returned Format:

Query returns "ON" or "OFF".

Example:

:MEAS:TOT ON	Set all the measurement functions "ON"
:MEAS:TOT?	Query returns "ON"

23. :MEASure:SOURce

Command Format:

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

Function Explanation:

The commands are used to set and query the current channel. The options of <source> are "CHANnel1" or "CHANnel2".

Returned Format:

Query returns "CH1" or "CH2".

Example:

:MEAS:SOUR CHAN1	Measure the waveform of "CHANnel1"
:MEAS:SOUR?	Query returns "CH1"

WAVeform Commands

WAVeform Commands are used to read the waveform datum.

WAVeform Commands include:

- :WAVeform:DATA?
- :WAVeform:MEMORYDATA?
- :WAVeform:POINts:MODE

The detailed information of each command are given as follows:

1. :WAVeform:DATA?

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

Function Explanation:
This command is used to read the waveform datum from the designated source. The options of <source> are CHANnel1, CHANnel2, MATH or FFT.

Returned Format:
Query returns 600 points when selecting CHANnel1, CHANnel2 and MATH; While query retures 500 points when selecting FFT.

2. :WAVeform:MEMORYDATA?

Command Format:
:WAVfeorm:MEMORYDATA? [<source>]

Function Explanation:
This command is used to read the waveform data from the designated source. In **STOP** state, the waveform datum come from the memory. 10k datum can be read in half-channel* state and 5k in other states.
In **RUN** state, the waveform datum come from the screen.
The options of <source> are CHANnel1 or CHANnel2.
Note*: The half-channel state means only one channel is open, the MATH function is closed and the timebase should be less than 20ns.

Returned Format:
Query returns the data have been read.

3. :WAVeform:POINts:MODE

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

Function Explanation:

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

Data points returned by **:WAVeform:DATA?** in different modes:

	NORMal	**RAW**	**MAX**
MATH	600	600	In **RUN** state, MAX is the same with NORMal; in **STOP** state, MAX is the same with RAW.
FFT	500	500	
CHx	600	5120	
Half-Channel CHx	600	10240	

Returned Format:
The query returns NORMal, MAXimum or RAW.

Example:

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

KEY Commands

KEY Commands are used to control the keys and knobs on the front panel of the oscilloscope.

KEY Commands include:

- :KEY:LOCK
- :KEY:RUN
- :KEY:AUTO
- :KEY:CHANnel1
- :KEY:CHANnel2
- :KEY:MATH
- :KEY:REF
- :KEY:F1
- :KEY:F2
- :KEY:F3
- :KEY:F4
- :KEY:F5
- :KEY:MNUoff
- :KEY:MEASure
- :KEY:CURSor
- :KEY:ACQuire
- :KEY:DISPlay
- :KEY:STORage
- :KEY:UTILity
- :KEY:MNUTIME
- :KEY:MNUTRIG
- :KEY:Trig%50
- :KEY:FORCe
- :KEY:V_POS_INC
- :KEY:V_POS_DEC
- :KEY:V_SCALE_INC
- :KEY:V_SCALE_DEC
- :KEY:H_SCALE_INC
- :KEY:H_SCALE_DEC
- :KEY:TRIG_LVL_INC
- :KEY:TRIG_LVL_DEC
- :KEY:H_POS_INC
- :KEY:H_POS_DEC
- :KEY:PROMPT_V
- :KEY:PROMPT_H
- :KEY:FUNCtion
- :KEY:+FUNCtion
- :KEY:-FUNCtion
- :KEY:PROMPT_V_POS
- :KEY:PROMPT_H_POS
- :KEY:PROMPT_TRIG_LVL
- :KEY:OFF

The detailed information of each command are given as follows:

1. :KEY:LOCK

Command Format:

:KEY:LOCK {ENABle|DISable}

:KEY:LOCK?

Function Explanation:

The commands are used to enable or disable the remote control of the front panel keys.

Returned Format:

Query returns "ENABLE" or "DISABLE".

Example:

:KEY:LOCK ENAB	Enable the key on the front panel
:KEY:LOCK?	Query returns "ENABLE"

2. :KEY:RUN

Command Format:

:KEY:RUN

Function Explanation:

This command is used to control the **RUN/STOP** mode of the oscilloscope. Repeating the command would make the oscilloscope switched between RUN and STOP.

3. :KEY:AUTO

Command Format:

:KEY:AUTO

Function Explanation:

This command is equivalent to the **AUTO** key, which enables the oscilloscope to set the optimum conditions to display the input signal (waveform).

4. :KEY:CHANnel1

Command Format:
:KEY:CHANnel1

Function Explanation:
This command is used to set the on-state for **Channel 1** and its menu . Repeating the command would make the channel switched between ON and OFF.

5. :KEY:CHANnel2

Command Format:
:KEY:CHANnel2

Function Explanation:
This command is used to set the on-state for **Channel 2** and its menu . Repeating the command would make the channel switched between ON and OFF.

6. :KEY:MATH

Command Format:
:KEY:MATH

Function Explanation:
This command is used to set the on-state for **MATH** function and its menu. Repeating the command would make the function switched between ON and OFF.

7. :KEY:REF

Command Format:
:KEY:REF

Function Explanation:

This command is used to set the on-state for **waveform reference** function and its menu. Repeating the command would make the reference switched switch between ON and OFF.

8. :KEY:F1

Command Format:
:KEY:F1

Function Explanation:
This command is similar to press and select the option of **Menu F1**. Repeating this command would enter next level of menu.

9. :KEY:F2

Command Format:
:KEY:F2

Function Explanation:
This command is similar to press and select the option of **Menu F2.** Repeating this command would enter next level of menu.

10. :KEY:F3

Command Format:
:KEY:F3

Function Explanation:
This command is similar to press and select the option of **Menu F3**. Repeating this command would enter next level of menu.

11. :KEY:F4

Command Format:
:KEY:F4

Function Explanation:

This command is similar to press and select the option of **Menu F4**. Repeating this command would enter next level of menu.

12. :KEY:F5

Command Format:

:KEY:F5

Function Explanation:

This command is similar to press and select the option of **Menu F5**. Repeating this command would enter next level of menu.

13. :KEY:MNUoff

Command Format:

:KEY:MNUoff

Function Explanation:

This command is used to set the on-state for current **menu**. Repeating the command would make the current menu switched between ON and OFF.

14. :KEY:MEASure

Command Format:

:KEY:MEASure

Function Explanation:

This command is used to set the on-state for **measure function** and its menu. Repeating the command would make the measure switched between ON and OFF.

15. :KEY:CURSor

Command Format:

:KEY:CURSor

Function Explanation:

This command is used to set the on-state for **cursor function** and its menu. Repeating the command would make the cursor switched between four cursor modes.

16. :KEY:ACQuire

Command Format:

:KEY:ACQuire

Function Explanation:

This command is used to set the on-state for **acquire function** and its menu. Repeating the command would make the acquisition switched between ON and OFF.

17. :KEY:DISPlay

Command Format:

:KEY:DISPlay

Function Explanation:

This command is used to set the on-state for **display function** and its menu. Repeating the command would make the display switched between ON and OFF.

18. :KEY:STORage

Command Format:

:KEY:STORage

Function Explanation:

This command is used to set the on-state for **storage function** and its menu. Repeating the command would make the storage switched between ON and OFF.

19. :KEY:UTILity

Command Format:
:KEY:UTILity

Function Explanation:
This command is used to set the on-state for **utility function** and its menu. Repeating the command would make the utility switched between ON and OFF.

20. :KEY:MNUTIME

Command Format:
:KEY:MNUTIME

Function Explanation:
This command is used to set the on-state for **horizontal system** and its menu. Repeating the command would make this function switched between ON and OFF.

21. :KEY:MNUTRIG

Command Format:
:KEY:MNUTRIG

Function Explanation:
This command is used to set the on-state for **trigger system** and its menu. Repeating the command would make this function switched between ON and OFF.

22. :KEY:Trig%50

Command Format:
:KEY:Trig%50

Function Explanation:

The command is used to set the trigger level to the vertical midpoint of the signal amplitude.

23. :KEY:FORCe

Command Format:
:KEY:FORCe

Function Explanation:
This command is used to release the remote control.

24. :KEY:V_POS_INC

Command Format:
:KEY:V_POS_INC

Function Explanation:
This command is used to move the waveform display up gradually.

25. :KEY:V_POS_DEC

Command Format:
:KEY:V_POS_DEC

Function Explanation:
This command is used to move the waveform display down gradually.

26. :KEY:V_SCALE_INC

Command Format:
:KEY:V_SCALE_INC

Function Explanation:
This command is used to increase the vertical scale of the current channel with the 1-2-5 step sequence.

27. :KEY:V_SCALE_DEC

Command Format:
:KEY:V_SCALE_DEC

Function Explanation:
This command is used to decrease the vertical scale of the current channel with the 5-2-1 step sequence.

28. :KEY:H_SCALE_INC

Command Format:
:KEY:H_SCALE_INC

Function Explanation:
This command is used to decrease the horizontal scale of the current channel with the 5-2-1 step sequence.

29. :KEY:H_SCALE_DEC

Command Format:
:KEY:H_SCALE_DEC

Function Explanation:
This command is used to increase the horizontal scale of the current channel with the 1-2-5 step sequence.

30. :KEY:TRIG_LVL_INC

Command Format:
:KEY:TRIG_LVL_INC

Function Explanation:
This command is used to increase the trigger level gradually.

31. :KEY:TRIG_LVL_DEC

Command Format:
:KEY:TRIG_LVL_DEC

Function Explanation:
This command is used to decrease the trigger level gradually.

32. :KEY:H_POS_INC

Command Format:
:KEY:H_POS_INC

Function Explanation:
This command is used to increase the horizontal offset of the current channel gradually.

33. :KEY:H_POS_DEC

Command Format:
:KEY:H_POS_DEC

Function Explanation:
This command is used to decrease the horizontal offset of the current channel with gradually.

34. :KEY:PROMPT_V

Command Format:
:KEY:PROMPT_V

Function Explanation:
This command is used to switch the vertical scale between Coarse and Fine. Coarse step sequence is 1-2-5, Fine is to adjust gradually.

35. :KEY:PROMPT_H

Command Format:
:KEY:PROMPT_H

Function Explanation:
This command is used to set the on-state for **delayed scan** function. Repeating the command would make the delay switched between ON and OFF.

36. :KEY:FUNCtion

Command Format:
:KEY:FUNCtion

Function Explanation:
This command is used to press the multi-function knob.

37. :KEY:+FUNCtion

Command Format:
:KEY:+FUNCtion

Function Explanation:
This command is used to increase the offset of the multi-function knob. (Equivalent turning the knob clockwise)

38. :KEY:-FUNCtion

Command Format:
:KEY:-FUNCtion

Function Explanation:
This command is used to decrease the offset of the multi-function knob. (Equivalent turning the knob anti-clockwise)

39. :KEY:PROMPT_V_POS

Command Format:
:KEY:PROMPT_V_POS

Function Explanation:
This command is used to reset the vertical display of the waveform to the zero point.

40. :KEY:PROMPT_H_POS

Command Format:
:KEY:PROMPT_H_POS

Function Explanation:
This command is used to adjust the trigger offset (or the delayed scan offset) to the horizontal zero point.

41. :KEY:PROMPT_TRIG_LVL

Command Format:
:KEY:PROMPT_TRIG_LVL

Function Explanation:
This command is used to adjust the trigger level to the central position of the screen.

42. :KEY:OFF

Command Format:
:KEY:OFF

Function Explanation:
This command is used to turn off CH1, CH2, MATH and REF keys one by one.

CURSor Commands

CURSor Commands are used to set cursor parameters and measure screen datum by manual, track or automatical mode.

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
- :CURSor:TRACk:CURAY?
- :CURSor:TRACk:CURBY?

The detailed information of each command are given as follows:

1. :CURSor:MODE

Command Format:
:CURSor:MODE {CLOSe|MANUal|TRACk|MEASure}
:CURSor:MODE?

Function Explanation:
This command is used to set and query system cursor mode as "CLOSe", "MANUal", "TRACk" or "MEASure".

Returned Format:
Query returns CLOSe, MANUal, TRACk or MEASure.

Example:

:CURS:MODE TRAC	Set cursor mode as "TRACk"
:CURS:MODE?	Query returns TRACk

2. :CURSor:MANUal:TYPE

Command Format:
:CURSor:MANUal:TYPE {TIME|AMPlitude}
:CURSor:MANUal:TYPE?

Function Explanation:
The commands are used to set and query cursor type as "TIME" or "AMPlitude" in manual mode.

Returned Format:
Query returns TIME or AMPlitude.

Example:

:CURS:MANU:TYPE TIME	Set cursor type as "TIME"
:CURS:MANU:TYPE?	Query returns TIME

3. :CURSor:MANUal:SOURce

Command Format:

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

Function Explanation:
The commands are used to set and query the source used in manual mode. The options of <source> are "CHANnel1", "CHANnel2" or "MATH".

Returned Format:
Query returns CHANnel1, CHANnel2 or MATH.

Example:

:CURS:MANU:SOUR CHAN1	Set the source as "CHANnel1"
:CURS:MANU:SOUR?	Query returns CHANnel1

4. :CURSor:MANUal:CURAX

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

Function Explanation:
The commands are used to set and query the position of cursor AX in manual mode. The range of <value> is 4~297.

Returned Format:
Query returns the set value of <value>.

Example:

:CURS:MANU:CURAX 100	Set position of cursor AX as 100
:CURS:MANU:CURAX?	Query returns 100

5. :CURSor:MANUal:CURAY

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

Function Explanation:

The commands are used to set and query the position of cursor AY in manual mode. The range of <value> is 4~194.

Returned Format:

Query returns the set value of <value>.

Example:

:CURS:MANU:CURAY 100	Set position of cursor AY as 100
:CURS:MANU:CURAY?	Query returns 100

6. :CURSor:MANUal:CURBX

Command Format:

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

Function Explanation:

This command is used to set and query the position of cursor BX in manual mode. The range of <value> is 4~297.

Returned Format:

Query returns the set value of <value>.

Example:

:CURS:MANU:CURBX 100	Set position of cursor BX as 100
:CURS:MANU:CURBX?	Query returns 100

7. :CURSor:MANUal:CURBY

Command Format:

:CURSor:MANUal:CURBY <value>
:CURSor:MANUal:CURBY?

Function Explanation:

This command is used to set and query the position of BY in manual mode.

The range of <value> is 4~194.

Returned Format:
Query returns the set value of <value>.

Example:

:CURS:MANU:CURBY 100	Set position of BY as 100
:CURS:MANU:CURBY?	Query returns 100

8. :CURSor:TRACk:SOURceA

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

Function Explanation:
This command is used to set and query the measure channel tracked by cursor A in track mode. The options of <source> are "CHANnel1", "CHANnel2", "MATH" or "NONE".

Returned Format:
Query returns CHANnel1, CHANnel2, MATH or NONE.

Example:

:CURS:TRAC:SOURA CHAN1	Set channel tracked by cursor A as "CHANnel1"
:CURS:TRAC:SOURA?	Query returns CHANnel1

9. :CURSor:TRACk:SOURceB

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

Function Explanation:
This command is used to set and query the measure channel tracked by cursor B in track mode. The options of <source> are "CHANnel1", "CHANnel2",

"MATH" or "NONE".

Returned Format:
Query returns CHANnel1, CHANnel2, MATH or NONE.

Example:

:CURS:TRAC:SOURB CHAN1	Set channel tracked by cursor B as "CHANnel1"
:CURS:TRAC:SOURB?	Query returns CHANnel1

10. :CURSor:TRACk:CURA

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

Function Explanation:
This command is used to set and query the position of cursor A in track mode. The range of <value> is 4~297.

Returned Format:
Query returns position of cursor A.

Example:

:CURS:TRAC:CURA 100	Set position of cursor as 100
:CURS:TRAC:CURA?	Query returns 100

11. :CURSor:TRACk:CURB

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

Function Explanation:
The commands are used to set and query the position of cursor B in track mode. The range of <value> is 4~297.

Returned Format:
Query returns position of cursor B.

Example:

:CURS:TRAC:CURB 100	Set position of cursor B as 100
:CURS:TRAC:CURB?	Query returns 100

12. :CURSor:TRACk:CURAY?

Command Format:
:CURSor:TRACk:CURAY?

Function Explanation:
This command is used to query the position of cursor AY in track mode.

Returned Format:
Query returns the position of cursor AY within 4~194.

13. :CURSor:TRACk:CURBY?

Command Format:
:CURSor:TRACk:CURBY?

Function Explanation:
This command is used to set and query the position of cursor BY in track mode.

Returned Format:
Query returns position of cursoe BY within 4~194.

Other Commands

Except for the commands to control the basic operation of instrument, some commands used to control system language, frequency counter and beeper will be introduced as follows:

Other Commands include:

- :INFO:LANGuage
- :COUNter:ENABle
- :BEEP:ENABle
- :BEEP:ACTion

The detailed information of each command are given as follows:

1. :INFO:LANGuage

Command Format:

:INFO:LANGuage <lang>
:INFO:LANGuage?

Function Explanation:

The commands are used to set and query the system language. The options of <lang> are "SIMPlifiedchinese", "TRADitionalchinese", "ENGLish", "KORean", "JAPanese", "FRENch", "GERMan", "RUSSian", "SPANish" or "PORTuguese".

Returned Format:

Query returns the set value of <lang>: Simplified Chinese, Traditional Chinese, English, Korean, Japanese, French, German, Russian, Spanish or Portuguese.

Example:

:INFO:LANG SIMP	Set the system language as "SIMPlifiedchinese"
:INFO:LANG?	Query returns Simplified Chinese

2. :COUNter:ENABle

Command Format:

:COUNter:ENABle {ON|OFF}
:COUNter:ENABle?

Function Explanation:

The commands are used to set and query the frequency counter function"ON" or "OFF".

Returned Format:

Query returns "ON" or "OFF".

Example:

:COUN:ENAB ON	Set the frequency counter function "ON"
:COUN:ENAB?	Query returns "ON"

3. :BEEP:ENABle

Command Format:

:BEEP:ENABle {ON|OFF}

:BEEP:ENABle?

Function Explanation:

The commands are used to set and query the beeper function "ON" or "OFF".

Returned Format:

Query returns "ON" or "OFF".

Example:

:BEEP:ENAB ON	Set the beeper "ON"
:BEEP:ENAB?	Query returns "ON"

4. :BEEP:ACTion

Command Format:

:BEEP:ACTion

Function Explanation:

This command is able to make a beep once (no matter whether it is opern or not).

Chapter 3 Programming Example

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 USB and RS232 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.

Both USB and RS232 ports can be used to achiece communication between DS1000CA series digital oscilloscopes and PC.

1. **By RS232**
   Please use RS232 cable to connect RS232 port on the rear panel of DS1000CA with PC.

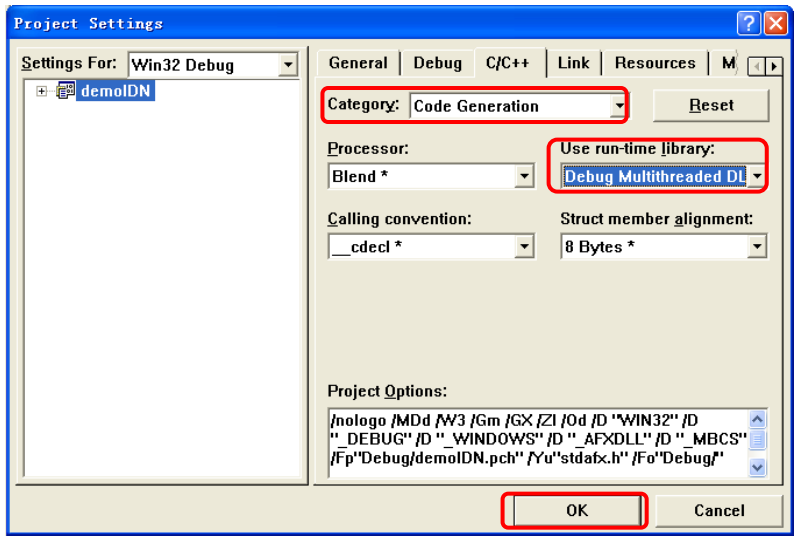
2. **By USB**
   Also, you can connect USB Device port on the rear panel of DS1000CA with PC by USB cable. After successful connection, turn on the instrument, a dialog will guide you to install the driver of "USB Test and Measurement Device" on the PC. See the figure below:

At present, 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 (Thereinto, the fist three program examples are based on USB communication, the last one is based on RS232 communicaion).

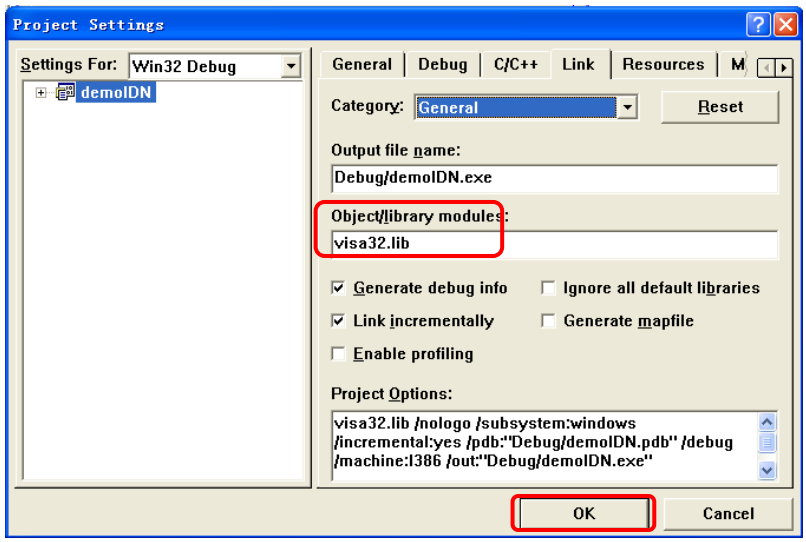
Program in Visual C++ 6.0

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

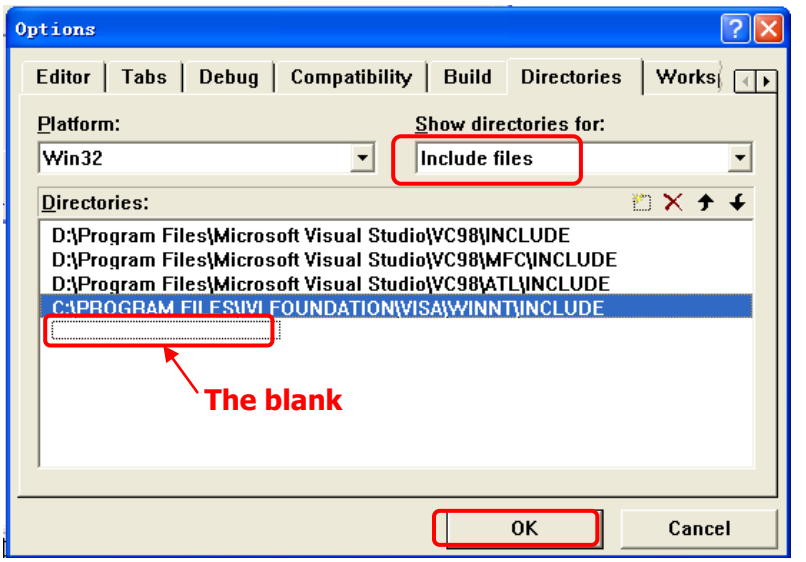
1. Create a project based on MFC.
2. Choose Project→Settings→C/C++; select "Code Generation" in Category and "Debug Multithreaded DLL" in Use run-time library; click OK.



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

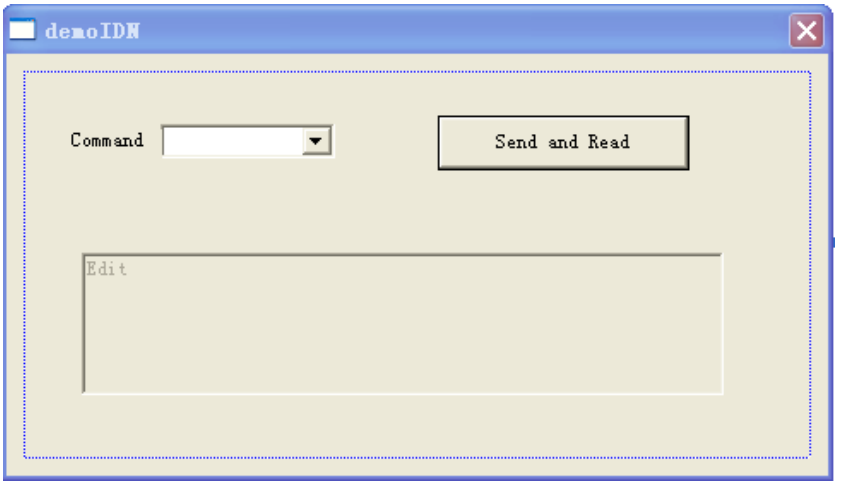


4. Choose Tools→Options→Directories; select "Include files" in Show directories for, and then dblclick the blank in Directories to add the path of "Include": C:\Program Files\IVI Foundation\VISA\WinNT\include.
   Select "Library files" in Show directories for, and then dblclick the blank in Directories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



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

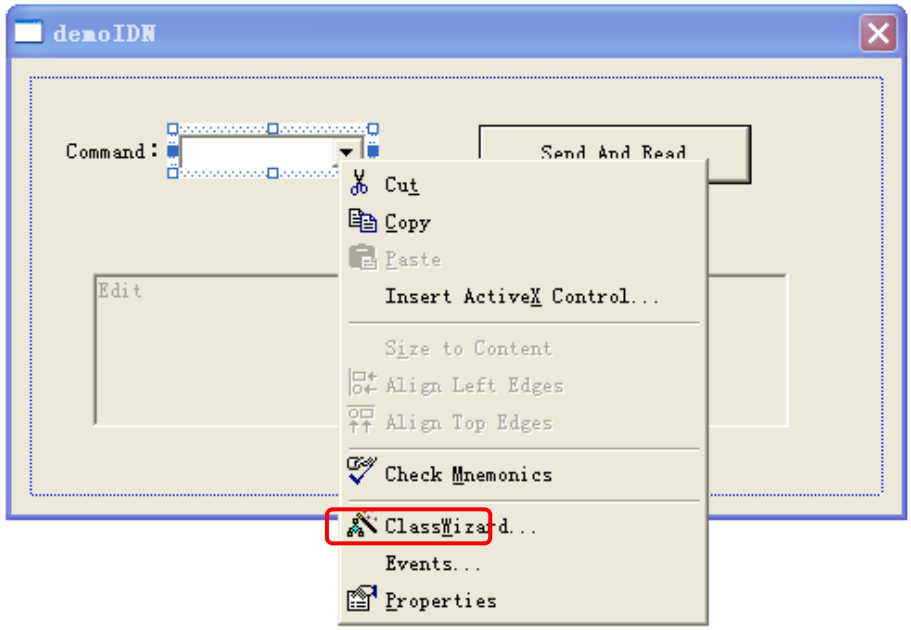
5. Add controls: Text, Com box, Button and Edit. See the figure below.

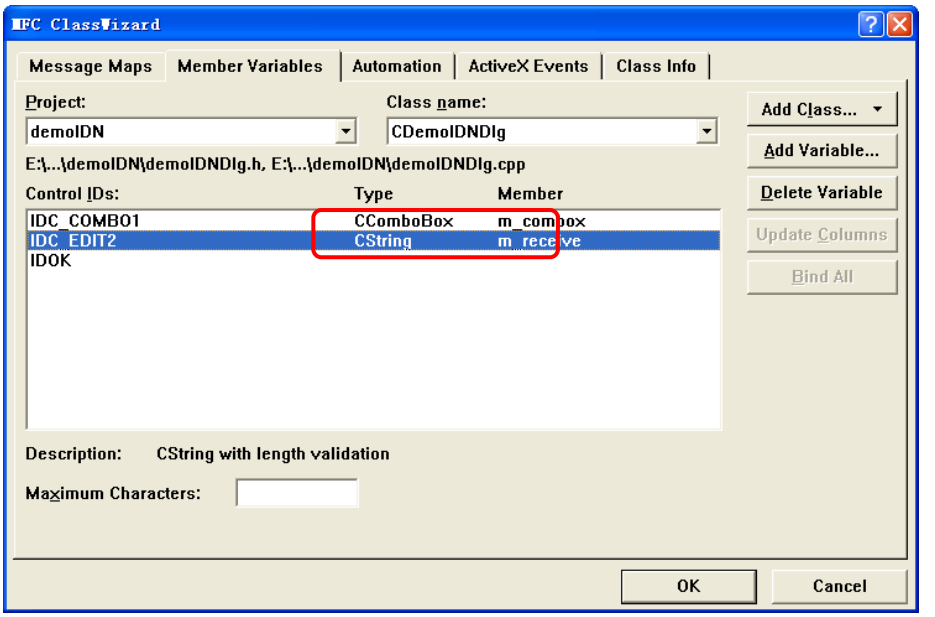


6. Modify the properties of the controls.

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);
```

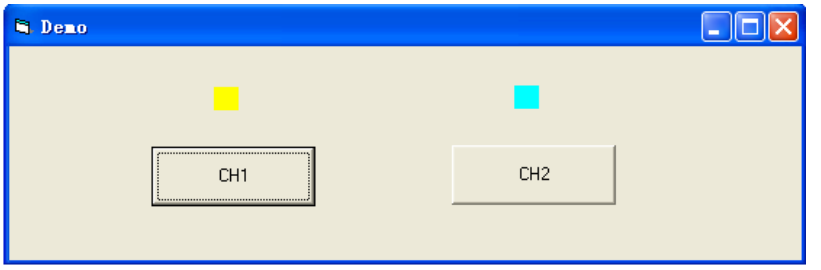
9. Save, build and run the project, you will get an EXE file. When the oscilloscope has been successfully connected with PC, choose a command such as *IDN? and click "Send and Read", the oscilloscope will return the result.

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 two Command Buttons and Lables to the demo, each button denotes each channel (CH1 and CH2).and each Label denotes different states (yellow and light 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. (the methods are similar for CH2)

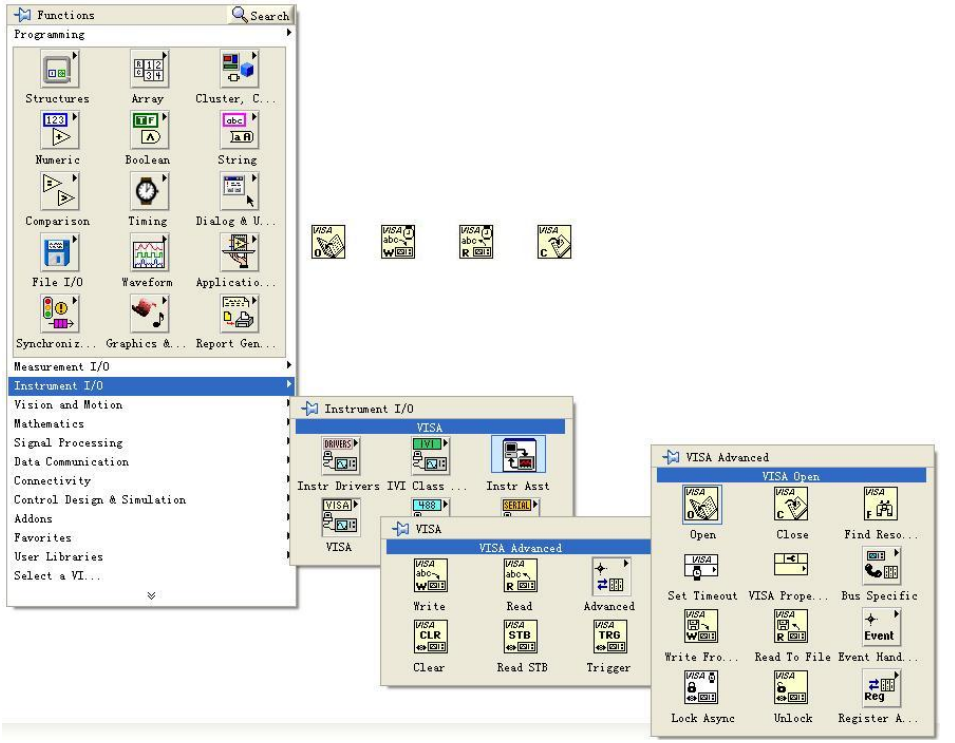
```
Dim defrm As Long
Dim vi As Long
Dim strRes As String * 200
Dim list As Long
Dim nmatches As Long
Dim matches As String * 200   ' reserve to acquire the equipment ID.

' 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.

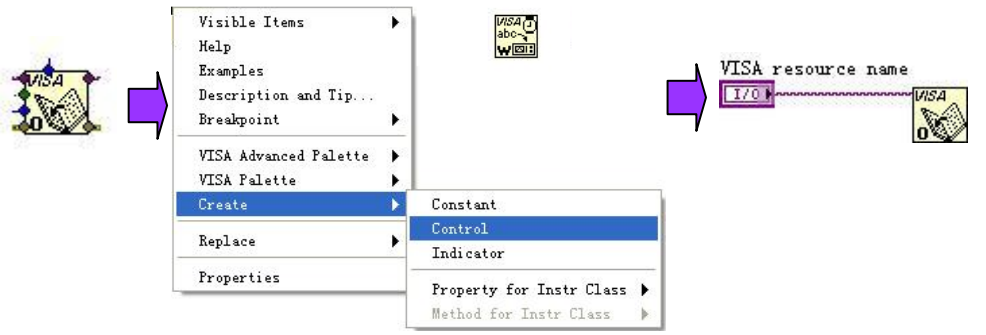
Program in LabVIEW 8.6

Open LabVIEW 8.6, take the following steps:

1. Open LabVIEW Block Diagram to enter program panel. Then separately add four functions: "VISA Open", "VISA Read", "VISA Write" and "VISA Close". See the figure below.

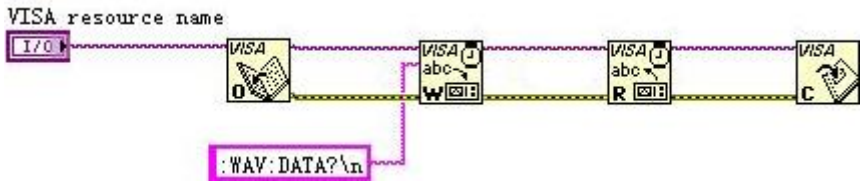


2. Move the mouse to the item of "VISA resource name" on the control of "VISA Open"; right-click the mouse to choose Create→Control. See the figure below.

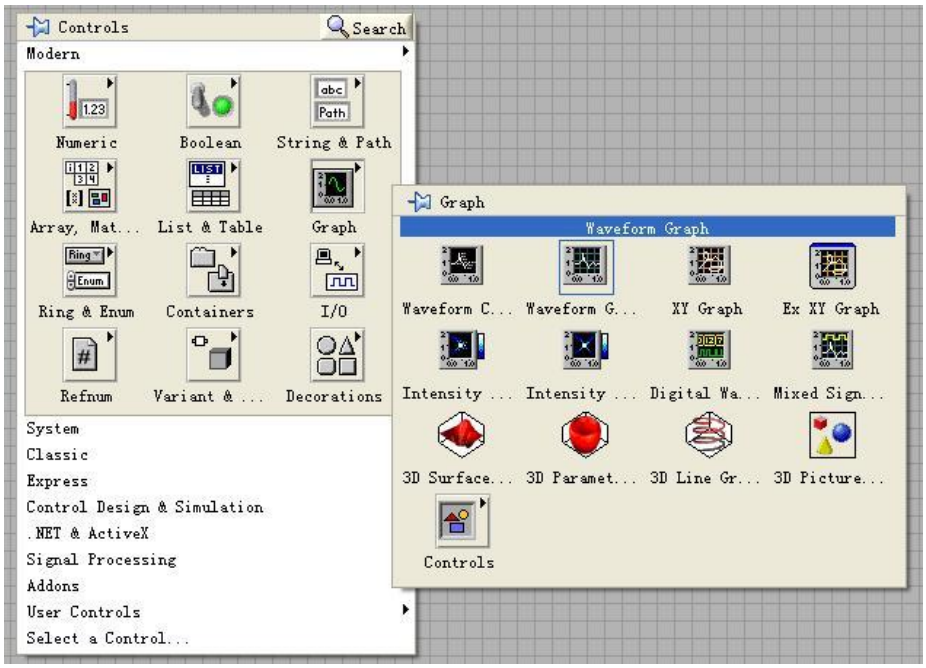


3. Separately connect "VISA resource name" with "VISA resource name out" and "error out" with "error in" of all the functions. See the figure below.

4. Add a tectbox written with: ":WAV:DATA?\n" on "write buffer" of "VISA Write, so as to read screen data.

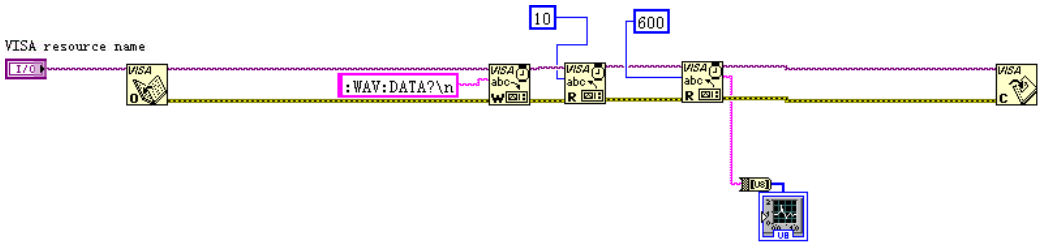


5. Open the Front Panel; choose Modern→Graph→Waveform Graph to add a Waveform Graph control. See the figure below.

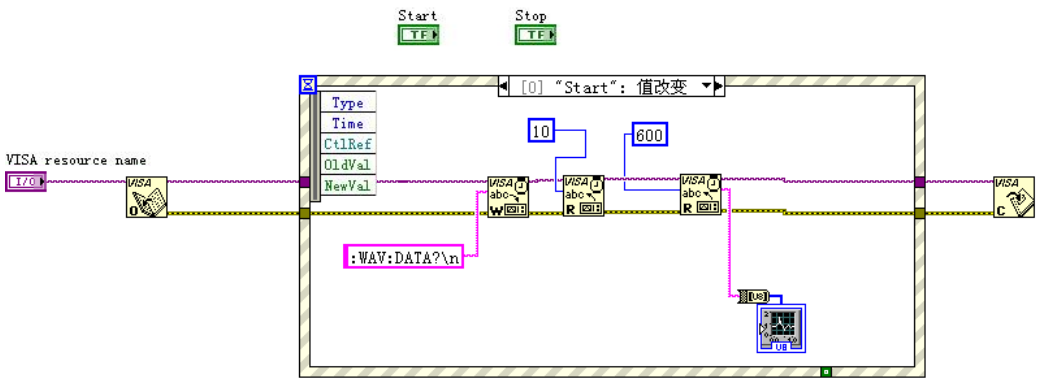


6. Open LabVIEW Block Diagram; right-click and choose "Programming"→"String"→ "String/Array/Path" and select "String To Byte Array"; then, use this function to connect "read buffer" on "VISA Read" function

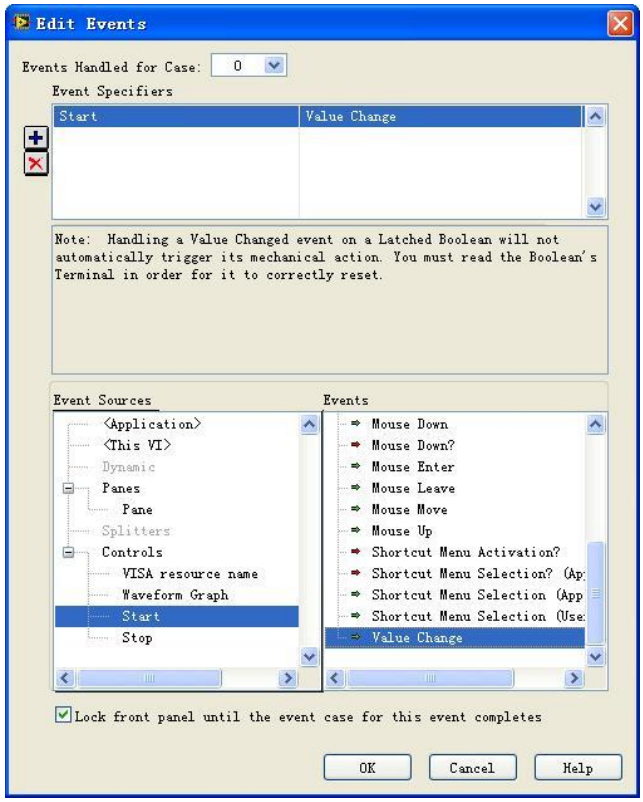
with the Waveform Graph. See the figure below.



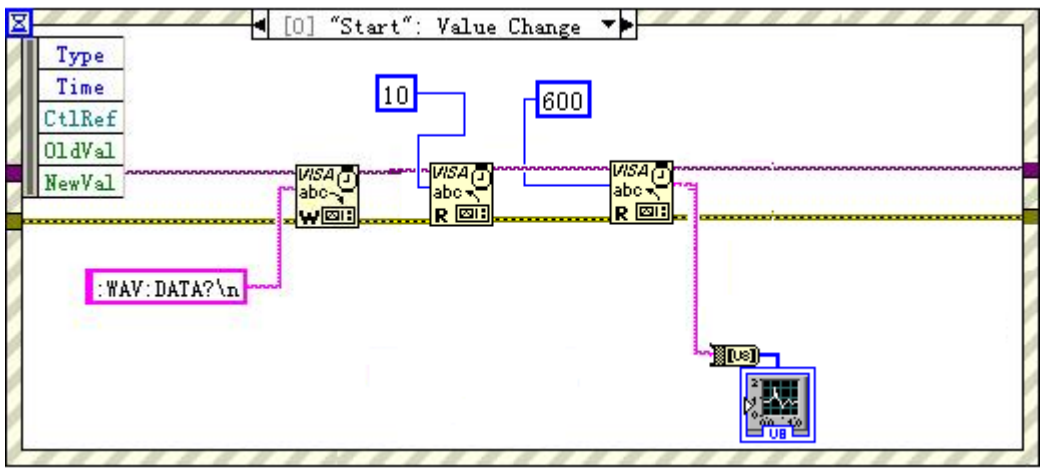
7. Add an Event Structure and a While Loop as well as two buttons, one of the buttons is used to control the start of waveform fetching, and the other one is to stop capturing. See the figure below.



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.

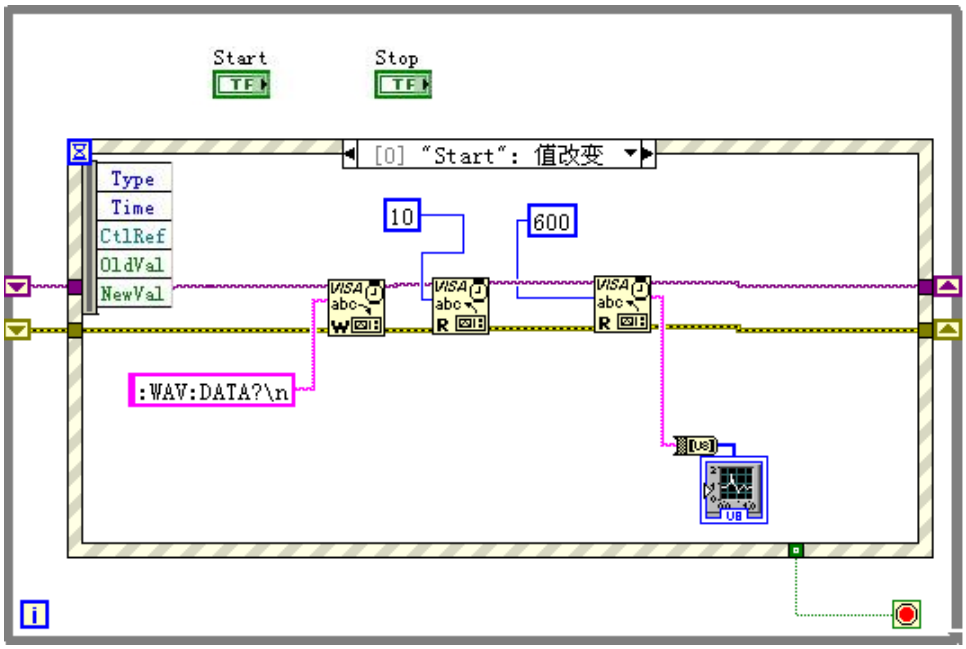


After you set the "Start" event, see the result below.

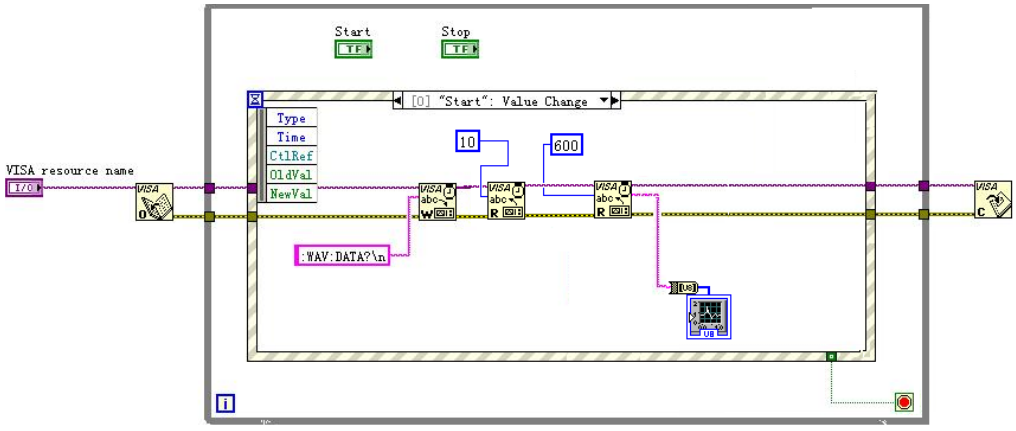


9. Add a While Loop; add "Boolean"→"True Constant" to point the event of the "Stop" button to While and exit*.

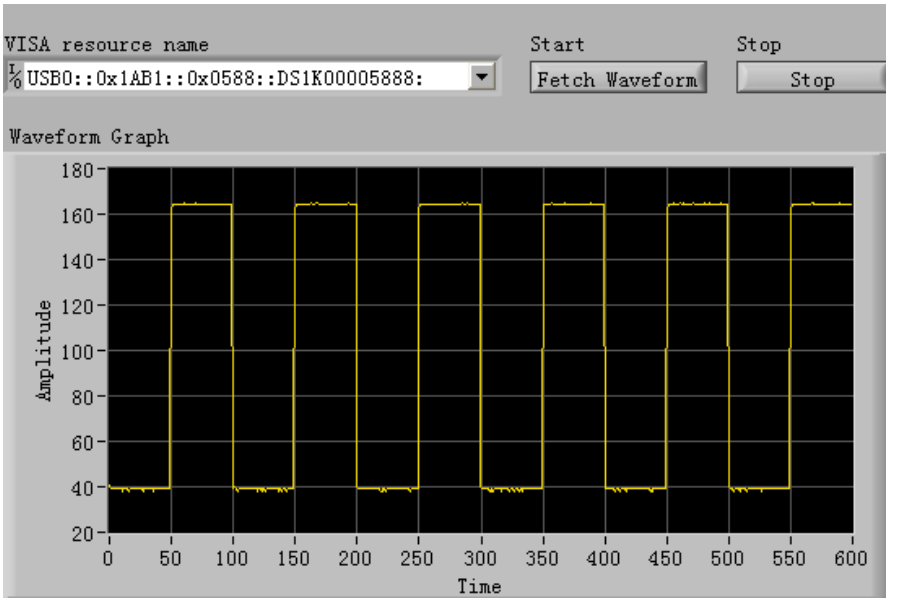
Note*: The waveform data head contains #8xxxxxxxx, 10 bytes in all, followed by waveform data points. So, 10 bytes will be read firstly, then reading 600 points waveform data.



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



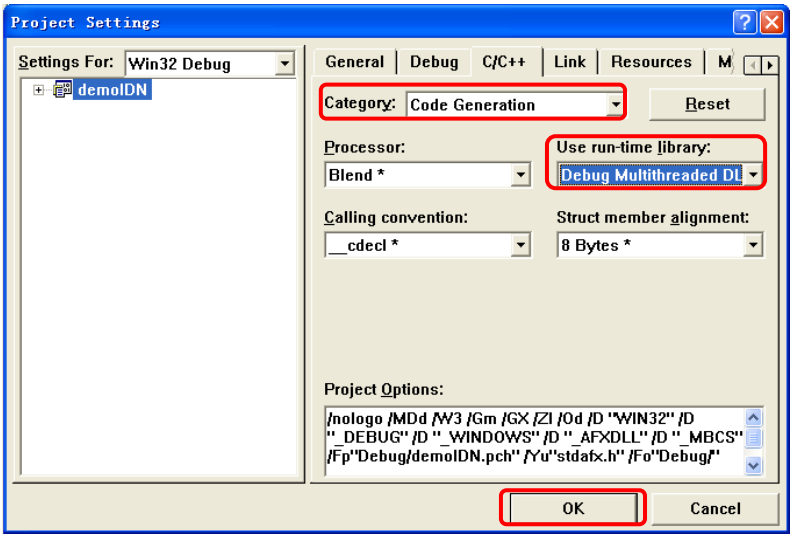
11. Adjust the style of Front Panel and click "Fetch Waveform" to get following interface. (the oscilloscope has been properly connected)


VISA resource name
USB0::0x1AB1::0x0588::DS1K00005888:
Start
Fetch Waveform
Stop
Stop
Waveform Graph
Amplitude
180
160
140
120
100
80
60
40
20
0
50
100
150
200
250
300
350
400
450
500
550
600
Time

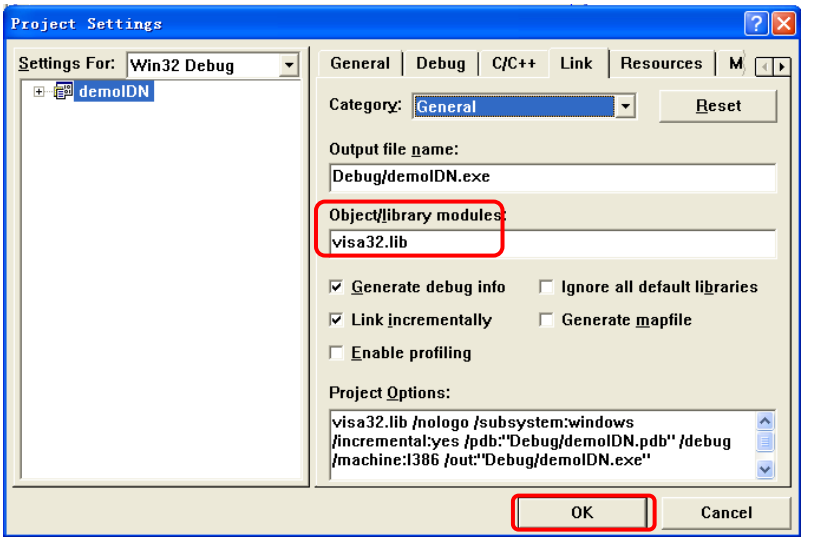
Program in Visual C++ 6.0 via Serial Port

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

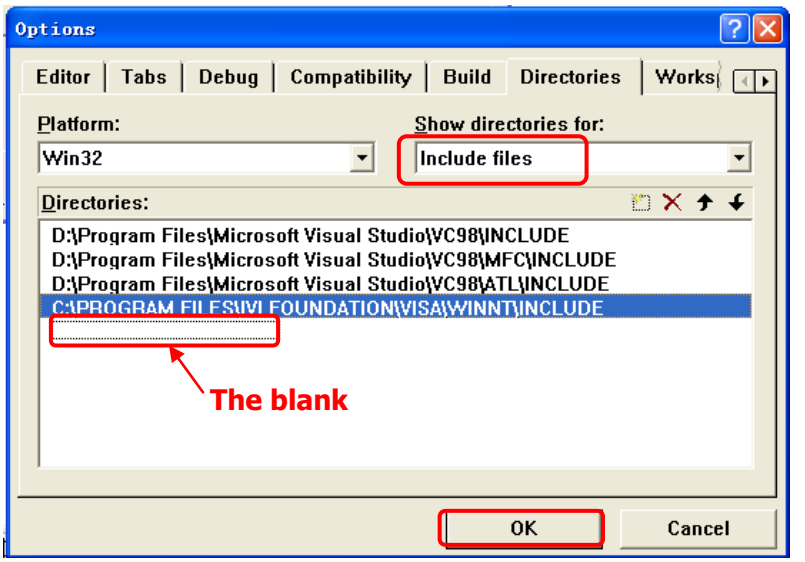
1. Create a project based on MFC.
2. Choose Project→Settings→C/C++; select "Code Generation" in Category and "Debug Multithreaded DLL" in Use run-time library; click OK.



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

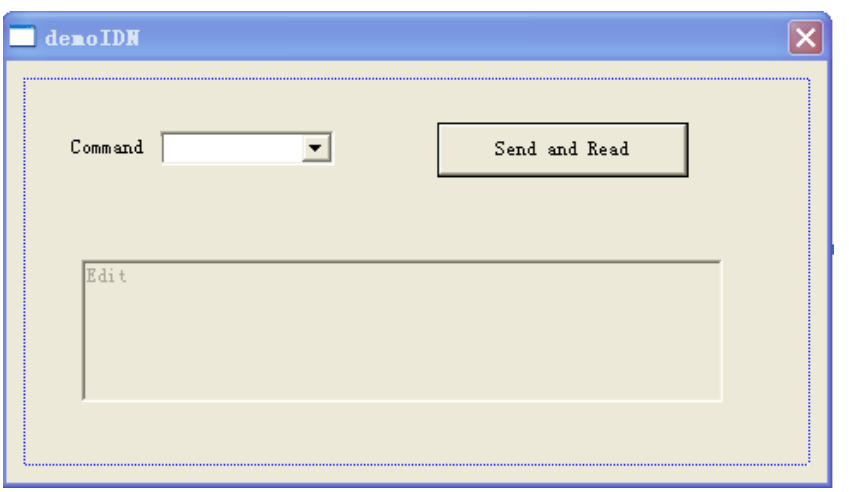


4. Choose Tools→Options→Directories; select "Include files" in Show directories for, and then dblclick the blank in Directories to add the path of "Include": C:\Program Files\IVI Foundation\VISA\WinNT\include.
   Select "Library files" in Show directories for, and then dblclick the blank in Directories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



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

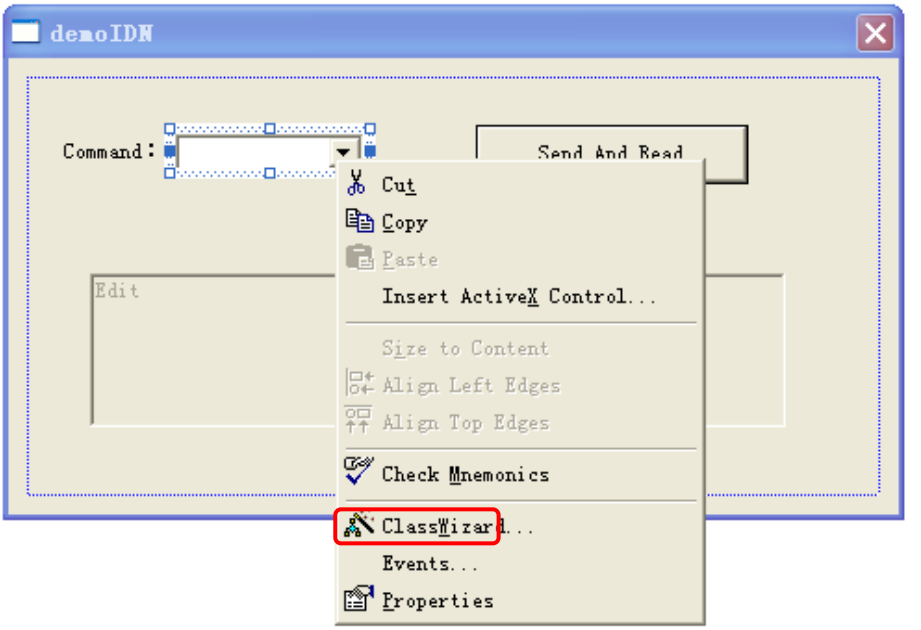
5. Add controls: Text, Com box, Button and Edit. See the figure below.

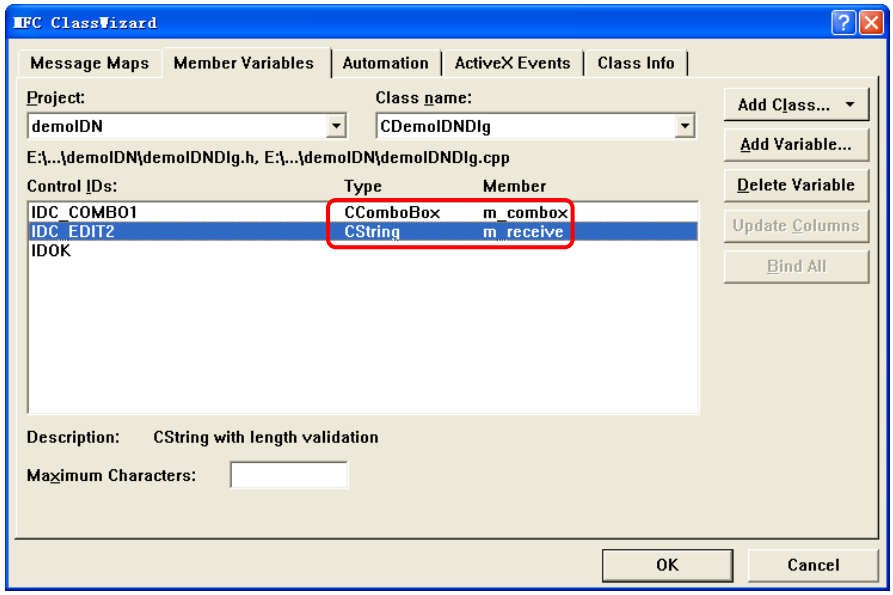


6. Modify the properties of the controls.

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 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;
unsigned long retCount;

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

//1. Open device Manager
viOpenDefaultRM (&defaultRM);

//2. Acquire RS232 resource of visa
viFindRsrc(defaultRM, "ASRL?*", &list,&nmatches, matches);
//Open serial device
viOpen (defaultRM,matches,VI_NULL,VI_NULL,&vi);
//4.Set up attribute of serial bus
viSetAttribute (vi, VI_ATTR_TMO_VALUE, 5000);
```

```
viSetAttribute (vi, VI_ATTR_TERMCHAR, 0xA);
viSetAttribute (vi, VI_ATTR_TERMCHAR_EN, VI_TRUE);

//5. Set up baud rate, stop bits and so on
viSetAttribute (vi, VI_ATTR_ASRL_BAUD, 9600);                    //Baud rate
viSetAttribute (vi, VI_ATTR_ASRL_DATA_BITS, 8);                  //Data Bits
viSetAttribute (vi, VI_ATTR_ASRL_STOP_BITS, VI_ASRL_STOP_ONE); //Stop
bits
viSetAttribute (vi, VI_ATTR_ASRL_PARITY, VI_ASRL_PAR_NONE);     //No
vertification

// send the receiving commands
m_combox.GetLBText(m_combox.GetCurSel(),strTemp);

sprintf( buf,"%s\n",strTemp.GetBuffer(10));
viWrite( vi, (unsigned char*)(&buf), strTemp.GetLength()+1, &retCount );

//Read the results
viScanf(vi,"%t\n",&buf );

m_receive = buf;

UpdateData (FALSE);

viClose (vi);
viClose (defaultRM);
```

Explanations*:

Pay attention to the following matters when adopts serial port communication.

(1) Set up a suitable bus timeout to avoid error communication cased by lower serial communication speed.

(2) In order to make a succesfull commnuication, settings of the serial port communication which contain baud rate, data bits, stop bits and parity check bits should be the same as oscilloscopes.

(3) During reading data by serial port, system will stop rading data when VISA meets end mark "0x0A". Please note this phenomenon when appears 0x0A. In addition, length of reading data is a standard to judge whether data aception is over.

Command Quick Reference A-Z

W