

RIGOL

Programming Guide

DP1308A Programmable Linear DC Power Supply

May 2009

RIGOL Technologies, Inc.

Copyright

© 2009 RIGOL Technologies, Inc. All Rights Reserved.

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 no matter by incidental or by 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, DP1308A has passed CE, cTUVus and LXI certification.

Structure of This Document

Chapter 1 Programming Overview

This chapter introduce you how to realize remote control to via SCPI commands.

Chapter 2 DP1308A Command Systems

This chapter gives detailed information on each command supported by DP1308A, including command format, function description, considerations when using command as well as some application examples.

Chapter 3 Programming Examples

This chapter provides some programming examples about common features in Visual C++ 6.0 and Visual Basic 6.0 development environment.

Command Quick Reference A-Z

The Appendix lists all the commands alphabetically in favor of quick reference.

Hint:

For the newest edition of this manual please go to http://www.rigolna.com/download_manual.aspx to download.

Contents

Chapter 1 Programming Overview	1-1
Communication Interfaces.....	1-2
SCPI Commands Introduction.....	1-3
Command Syntax	1-3
Symbol Description	1-4
Parameter Type	1-5
Command Abbreviation	1-6
Chapter 2 DP1308A Command Systems	2-1
General Commands	2-2
APPLy Commands.....	2-4
INSTrument Commands	2-5
MEASure Commands	2-7
OUTPut Commands	2-8
SOURce Commands.....	2-13
DISPlay Commands	2-15
SYSTem Commands.....	2-16
STORe/RECAll Commands	2-24
Chapter 3 Programming Examples	3-1
Prepare for Programming	3-2
Program in Visual C++ 6.0.....	3-3
Program in Visual Basic 6.0	3-6
Command Quick Reference A-Z	1

Chapter 1 Programming Overview

This chapter introduces you how to control DP1308A via SCPI commands (Standard Commands for Programmable Instruments).

The chapter contains following topics:

- Communication Interfaces
- SCPI Commands Introduction
 - Command Syntax
 - Symbol Description
 - Parameter Type
 - Command Abbreviation

Communication Interfaces

DP1308A provides three kinds of interfaces for connecting with computer: LAN, USB Device and GPIB.

Operations that you can do with a computer and a generator include:

- Setup the DP1308A
- Output programmable linear DC power supply

Connection:

Please connect the USB Device port of DP1308A with the USB interface on the computer using an USB cable.

SCPI Commands Introduction

Command Syntax

SCPI is a kind of command language intends for test and measurement instrument based on ASCII. The commands present a hierarchy and contain different sub-systems that distinguished by different root keywords. Each of command consists of a "Root" keyword and one or multilayered keywords separated by ":". Parameters are permitted to follow the keywords; "?" appeared following a command line denotes to query its function; besides, "space" is used to divide command and parameter.

For example:

```
SYSTem:LANGuage:TYPE {EN|CH}
```

```
SYSTem:LANGuage:TYPE?
```

SYSTem is the root keyword of this command, **LANGuage** and **TYPE** 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. "space" is used to divide the command keyword SYSTem:LANGuage:TYPE and the parameter {EN|CH}. SYSTem:LANGuage:TYPE? denotes to query.

Besides, "," is used for separating different parameters, for example:

```
OUTPut:TIMer {P6V|P25V|N25V},<secnum>,<volt>,<curr>,<time>
```

Symbol Description

The following symbols are not “real” parts of **SCPI** commands, but they are usually used to assist to explain the parameters contained in a command line.

1. Braces { }

The parameters enclosed in a command line such as: {EN|CH}

2. Vertical bar |

The vertical bar separates two or more optional parameters. Only one parameter could be selected for a command such as: Only ON or OFF could be selected from {ON|OFF}.

3. Square brackets []

The contents such as keywords or parameters enclosed in square brackets could be omitted but all of them would be executed regardless whether they were omitted such as:

SYSTem:BEEPer [:IMMEdiate] {ON|OFF}

Thereinto, [:IMMEdiate] can be omitted.

4. Triangle Brackets < >

The parameter value enclosed in < > must be an effective value such as:

SYSTem:BRIGht <brightness>

Thereinto, <brightness> must be a numerical value such as: SYSTem:BRIGht 5.

Parameter Type

The commands contains 6 kinds of parameters, different parameters has different setting methods.

1. MINimum and MAXimum

Generally, MINimum or MAXimum is used to replace the parameter in some commands. For example:

CURRent {<current>|MINimum|MAXimum}

You can set specific current value using<current> or directly set the current as MINimum or MAXimum.

2. Boolean

The parameter should be "OFF" or "ON". For example:

DISPlay:FOCUS {ON|OFF},

"ON" denotes turn on (enable) the focus display, "OFF" denotes turn off.

3. Consecutive Integer

The parameter should be a consecutive integer. For example:

SYSTem:BRIGht <brightness>

<brightness> could be an integer between 1 and 8 (including 1 and 8).

4. Consecutive Real Number

The parameters can be any value only within effective range and precision permitting. For example:

CURRent {<current>|MINimum|MAXimum}

The command sets the current value of current channel, <count> could be any real number within the permitting range.

5. Discrete

The parameters can only be the cited value. For example:

RECALL:LOCAL {1|2|3|4}, the parameter could be 1, 2, 3 or 4.

6. ASCII Character String

The parameter should be composed of ASCII character string. For example:

STORe:LOCAL {1|2|3|4} <name>.

Thereint, <name> should be a ASCII character string.

Command Abbreviation

According to SCPI syntax, most of commands are consists of uppercase and lowercase letters together, and those uppercase letters are the abbreviation of a command. All the commands are case-insensitive, so you can use any kind of them. But if use abbreviation, the capital letters specified in commands must be written completely. For example:

INSTrument:NSElect?

also could be:

INST:NSEL? or inst:nsel?

Chapter 2 DP1308A Command Systems

This chapter gives detailed information on each command supported by DP1308A, including command format, function description, using considerations as well as some application examples.

DP1308A contains following subcommands systems:

- General Commands
- APPLy Commands
- INSTRument Commands
- MEASure Commands
- OUTPut Commands
- SOURce Commands
- DISPlay Commands
- SYSTem Commands
- STORe/RECAll Commands

General Commands

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

DP1308A supports following **IEEE488.2** commands:

1. *IDN?
2. *RST
3. *TST?
4. *SAV
5. *RCL

The detailed information of each command about **IEEE488.2** are:

1. *IDN?	
Command Format	*IDN?
Function	Query ID character string of the instrument, including a field separated by 4 commas ","; manufactory, model, serial number and the edition number that consists of numbers and separated by "." .
Return Value	Rigol Technologies,DP1308A,DP1A110300105,00.01.00.00.01.02.01.01.03.00
2. *RST	
Command Format	*RST
Function	Reset the system parameters of the instrument.
3. *TST?	
Command Format	*TST?
Function	Query the results of self-test.
Return Value	The query returns Pass or Error.
4. *SAV	
Command Format	*SAV {1 2 3 4},<name>

Function	Save the current system state to the nonvolatile memory as the name assigned by <name>.
Explanation	4 store locations are available for storing the state of the instrument. (number "1", "2", "3" and "4")
5. *RCL	
Command Format	*RCL {1 2 3 4}
Function	Recall the state of the instrument that has been saved.

APPLy Commands

APPLy commands are used for quickly choosing the channel that needs to measure and its voltage and current value. Among these commands, it makes the control over remote interface more easier.

DP1308A supports following **APPLy** commands:

1. APPLy
2. APPLy?

The detailed information of each command are:

1. APPLy	
Command Format	APPLy {P6V P25V N25V}[,{<volt> DEFault MINimum MAXimum}[,{<curr> DEFault MINimum MAXimum}]]
Function	Set the voltage and current value for a channel.
Explanations	If only set a parameter, <volt> is set. If no parameters is setup, the command selects the channe like INSTrument[:SELEct] command.
Example	1. APPL P25V,10,0.5 Set the voltage and current on +25V channel as 10 V and 0.5 A. 2. APPL P6V,6 set the voltage on +6V channel as +6V. 3. APPL N25V Select -25V channel as the current channel.
2. APPLy?	
Command Format	APPLy? [{P6V P25V N25V}]
Function	Query the current voltage and current value of the specified channel.
Explanations	If no channel is specified, the query returns the voltage and current value of the current channel.
Example	P25V,Limit,10.00V,0.500A

INSTRument Commands

Being different with **Apply** commands, **INSTRument** commands are more flexibility in channel selection, setup and query as well as others, which provide basic method for choosing operation channel.

DP1308A supports following **INSTRument** commands:

1. INSTRument[:SELEct]
2. INSTRument[:SELEct]?
3. INSTRument:NSELEct
4. INSTRument:NSELEct?

The detailed information of each command are:

1. INSTRument[:SELEct]	
Command Format	INSTRument[:SELEct] {P6V P25V N25V}
Function	Select the channel you want to use.
Example	INST:SELE P25V
2. INSTRument[:SELEct]?	
Command Format	INSTRument[:SELEct]?
Function	Query the current channel.
Return Value	The query returns P6V, P25V or N25V.
3. INSTRument:NSELEct	
Command Format	INSTRument:NSELEct {1 2 3}
Function	Select the channel you want to use.
Explanations	The channel identifiers in comand INSTRument[:SELEct] are expressed in numbers, thereinto, "1" denotes P6V, "2" denotes P25V and "3" denotes N25V.
Example	INST:NSEL 2
4. INSTRument:NSELEct?	
Command	INSTRument:NSELEct?

Format	
Function	Query the current channel.
Return Value	The query returns 1, 2 or 3, which denotes P6V, P25V and N25V, separately.

MEASure Commands

MEASure commands are used for querying the voltage, current or power value at the current output terminal of the instrument.

DP1308A supports following **MEASure** commands:

1. MEASure:CURRent[:DC]?
2. MEASure[:VOLTage][:DC]?
3. MEASure:POWEr[:DC]?

The detailed information of each command are:

1. MEASure:CURRent[:DC]?	
Command Format	MEASure:CURRent[:DC]?[{P6V P25V N25V}]
Function	Query the current value at the output terminal of specified channel.
Explanations	If no channel is specified, the command queries the current value at the output terminal of current channel.
Example	Such as: 0.053
2. MEASure[:VOLTage][:DC]?	
Command Format	MEASure[:VOLTage][:DC]?[{P6V P25V N25V}]
Function	Query the voltage value at the output terminal of specified channel.
Explanations	If no channel is specified, the command queries the voltage value at the output terminal of current channel.
Return Value	Such as: 3.331
3. MEASure:POWEr[:DC]?	
Command Format	MEASure:POWEr[:DC]?[{P6V P25V N25V}]
Function	Query the power value at the output terminal of specified channel.
Explanations	If no channel is specified, the command queries the power value at the output terminal of current channel.
Return Value	Such as: 0.06

OUTPut Commands

OUTPut commands are used for configuring and querying different settings about the instrument including On/Off state, OCP and OVP (for channel), track mode, waveform display and timing function and others.

DP1308A supports following **OUTPut** commands:

1. OUTPut[:STATe]
2. OUTPut[:STATe]?
3. OUTPut:OVP:STATe
4. OUTPut:OVP:STATe?
5. OUTPut:OVP
6. OUTPut:OVP?
7. OUTPut:OCP:STATe
8. OUTPut:OCP:STATe?
9. OUTPut:OCP
10. OUTPut:OCP?
11. OUTPut:TRACK
12. OUTPut:TRACK?
13. OUTPut:WAVE
14. OUTPut:WAVE?
15. OUTPut:TIMER
16. OUTPut:TIMER?
17. OUTPut:TIMER:STATe
18. OUTPut:TIMER:STATe?

The detailed information of each command are:

1. OUTPut[:STATe]	
Command	OUTPut[:STATe] {P6V P25V N25V},{OFF ON}
Format	
Function	Turn on or off the specified channel.
Example	OUTP:STAT N25V, ON
2. OUTPut[:STATe]?	
Command	OUTPut[:STATe]? {P6V P25V N25V}
Format	

Function	Query whether the specified channel is turned on.
Return Value	The query returns ON or OFF.
3. OUTPut:OVP:STATe	
Command Format	OUTPut:OVP:STATe {P6V P25V N25V},{ON OFF}
Function	Enable or disable the overvoltage protection function (OVP) of the specified channel.
Example	OUTP:OVP:STAT N25V,ON
4. OUTPut:OVP:STATe?	
Command Format	OUTPut:OVP:STATe? {P6V P25V N25V}
Function	Query whether the overvoltage protection function (OVP) of the specified channel is enabled or not.
Return Value	The query returns ON or OFF.
5. OUTPut:OVP	
Command Format	OUTPut:OVP {P6V P25V N25V},<value>
Function	Setup the overvoltage protection value (OVP) of the specified channel.
Explanations	<value> denotes the overvoltage protection value, which varies on different channels: 1. On +6V channel: 0.1V~6.5V; 2. On +25V channel: 0.1V~27V; 3. On -25V channel: -0.1V~-27V.
Example	OUTP:OVP P6V,2
6. OUTPut:OVP?	
Command Format	OUTPut:OVP? {P6V P25V N25V}
Function	Query the overvoltage protection value (OVP) of the specified channel.
Return Value	Such as: -27.0
7. OUTPut:OCP:STATe	

Command Format	OUTPut:OCP:STATe {P6V P25V N25V},{ON OFF}
Function	Enable or disable the overcurrent protection function (OCP) of the specified channel.
Example	OUTP:OCP:STAT P25V,ON
8. OUTPut:OCP:STATe?	
Command Format	OUTPut:OCP:STATe? {P6V P25V N25V}
Function	Query whether the overcurrent protection function (OCP) of the specified channel is enabled or not.
Return Value	The query returns ON or OFF.
9. OUTPut:OCP	
Command Format	OUTPut:OCP {P6V P25V N25V},<value>
Function	Setup the overcurrent protection value (OCP) of the specified channel.
Explanations	<value> denotes the overcurrent protection value, which varies on different channels: <ol style="list-style-type: none"> 1. On +6V channel: 0.1A~5.5A; 2. On +25V channel: 0.1A~1.2A; 3. On -25V channel: 0.1A~1.2A.
Example	OUTP:OCP N25V,1
10. OUTPut:OCP?	
Command Format	OUTPut:OCP? {P6V P25V N25V}
Function	Query the overcurrent protection value (OCP) of the specified channel.
Return Value	Such as: 1.0
11. OUTPut:TRACK	
Command Format	OUTPut:TRACK {P25V N25V OFF}
Function	Setup the track state of the specified channel.
Explanations	<ul style="list-style-type: none"> ● Track function is available only on +25V and -25V channel.

	<ul style="list-style-type: none"> ● The electrical level of $\pm 25V$ channel would be the same as soon as the track function is enabled. In track mode, the voltage setting value of channel under tracking would be vary with the voltage setting value of channel been tracked, which is also applied to voltage output value if both +25V and -25V channel are in CV mode. ● To exit track mode, please select "OFF" parameter.
Example	OUTP:TRAC P25V
12. OUTPut:TRACK?	
Command Format	OUTPut:TRACK?
Function	Query the track state of the specified channel.
Return Value	Such as: TRACK_P25_ON
13. OUTPut:WAVE	
Command Format	OUTPut:WAVE {P6V P25V N25V},{OFF ON}
Function	Turn on or off the waveform display function of the specified channel.
Example	OUTP:WAVE N25V, OFF
14. OUTPut:WAVE?	
Command Format	OUTPut:WAVE? {P6V P25V N25V}
Function	Query whether the waveform display function of the specified channel is turned on.
Return Value	The query returns ON or OFF.
15. OUTPut:TIMEr	
Command Format	OUTPut:TIMEr {P6V P25V N25V},<secnum>,<volt>,<curr>,<time>
Function	Setup the timing parameters of the specified channel.
Explanations	The values such as voltage<volt>, current<curr> and output time <time> of a channel could be defined as required, and up to five groups of outputs could be set per channel, the range of <secnum> is 1~5.

Example	OUTP:TIME P25V,2,5,0.5,2
16. OUTPut:TIMEr?	
Command Format	OUTPut:TIMEr? {P6V P25V N25V}
Function	Query the timing parameters of the specified channel.
Return Value	The query returns the five groups of timing parameters and each group of them contains: <secnum>, <volt>, <curr>, <time>; such as: 1,0.00,0.000,0; 2,0.00,0.000,0; 3,0.00,0.000,0; 4,0.00,0.000,0; 5,0.00,0.000,0;
17. OUTPut:TIMEr:STATe	
Command Format	OUTPut:TIMEr:STATe {P6V P25V N25V},{OFF ON}
Function	Turn off or on the timing output of the specified channel.
Example	OUTP:TIME:STAT P6V,ON
18. OUTPut:TIMEr:STATe?	
Command Format	OUTPut:TIMEr:STATe? {P6V P25V N25V}
Function	Query whether the timing output of the specified channel is enabled or not.
Return Value	The query returns PAUSE, RUN or OFF.

SOURce Commands

SOURce commands are used for setting the current and voltage value for the current channel.

DP1308A supports following **SOURce** commands:

1. [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]
2. [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?
3. [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]
4. [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]?

The detailed information of each command are:

1. [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]	
Command	[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]
Format	{<current> MINimum MAXimum}
Function	Setup the current value of the current channel.
Explanations	<ul style="list-style-type: none"> ● The current of the current channel would be the setting value as soon as this command was received successfully. ● If select "<current>", the current value will be set manually; "MINimum" parameter denotes set the current value as minimum, and "MAXimum" denotes maximum. ● Different channels has different current ranges. On +6V channel: 0~5.25A; On ±25V channel: 0~1.05A
Example	SOUR:CURR:LEV:IMM:AMPL 0.5
2. [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?	
Command	[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]? [MINimum
Format	MAXimum]
Function	Query the current value of the current channel.
Return Value	<ul style="list-style-type: none"> ● The query returns the current value of the current channel once receive command. ● If select "MINimum", that denotes to query and return the minimum value and "MAXimum" denotes maximum.
3. [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]	

Command Format	[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] {<voltage> MINimum MAXimum}
Function	Setup voltage value of the current channel.
Explanations	<ul style="list-style-type: none"> ● The voltage of the current channel would be the setting value as soon as this command was received successfully. ● If select "<voltage>", the voltage value will be set manually; "MINimum" denotes set the current value as minimum, and "MAXimum" denotes maximum. ● Different channles has different voltage ranges. On +6V: 0~+6.3V; On +25V: 0~+26.25V; On -25V: 0~-26.25V.
Example	SOUR:VOLT:LEV:IMM:AMPL MIN
4. [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]?	
Command Format	[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? [MINimum MAXimum]
Function	Query the voltage value of the current channel.
Return Value	<ul style="list-style-type: none"> ● The query returns the voltage value of the current channel once receive command. ● If select "MINimum", that denotes to query and return the minimum value and "MAXimum" denotes maximum.

DISPlay Commands

DISPlay commands are used for setting the display parameters for the instrument.

DP1308A supports following **DISPlay** command:

1. DISPlay:FOCUs

1. DISPlay:FOCUs	
Command Format	DISPlay:FOCUs {ON OFF}
Function	Enable or disable the focus display of the instrument.
Example	DISP:FOCU OFF

SYSTem Commands

SYSTem commands provide some system information such as: language, function setting, power-on mode, instrument control, LAN setting, system self-test and the like.

DP1308A supports following **SYSTem** commands:

1. SYSTem:LANGuage:TYPE
2. SYSTem:LANGuage:TYPE?
3. SYSTem:BEEPer[:IMMediate]
4. SYSTem:BEEPer[:IMMediate]?
5. SYSTem:BRIGht
6. SYSTem:BRIGht?
7. SYSTem:POWEron
8. SYSTem:POWEron?
9. SYSTem:LOCal
10. SYSTem:REMOte
11. SYSTem:COMMunicate:LAN:DHCP[:STATe]
12. SYSTem:COMMunicate:LAN:DHCP[:STATe]?
13. SYSTem:COMMunicate:LAN:AUTOip[:STATe]
14. SYSTem:COMMunicate:LAN:AUTOip[:STATe]?
15. SYSTem:COMMunicate:LAN:MANualip[:STATe]
16. SYSTem:COMMunicate:LAN:MANualip[:STATe]?
17. SYSTem:COMMunicate:LAN:IPADdress
18. SYSTem:COMMunicate:LAN:IPADdress?
19. SYSTem:COMMunicate:LAN:SMASk
20. SYSTem:COMMunicate:LAN:SMASk?
21. SYSTem:COMMunicate:LAN:GATEway
22. SYSTem:COMMunicate:LAN:GATEway?
23. SYSTem:COMMunicate:LAN:DNS
24. SYSTem:COMMunicate:LAN:DNS?
25. SYSTem:COMMunicate:GPIB:ADDRes
26. SYSTem:COMMunicate:GPIB:ADDRes?
27. SYSTem:COMMunicate:LAN:APPLy
28. SYSTem:SELF:TEST:AD?
29. SYSTem:SELF:TEST:DA?
30. SYSTem:SELF:TEST:FPGA?
31. SYSTem:SELF:TEST:USB?

- 32. SYSTem:SELF:TEST:LAN?
- 33. SYSTem:SELF:TEST:FAN?
- 34. SYSTem:OTP
- 35. SYSTem:OTP?

The detailed information of each command are:

1. SYSTem:LANGUage:TYPE	
Command Format	SYSTem:LANGUage:TYPE {EN CH}
Function	Choose the type of system language.
Example	SYST:LANG:TYPE CH
2. SYSTem:LANGUage:TYPE?	
Command Format	SYSTem:LANGUage:TYPE?
Function	Query the selected type of system language.
Return Value	The query returns English or Chinese.
3. SYSTem:BEEPer[:IMMediate]	
Command Format	SYSTem:BEEPer[:IMMediate] {ON OFF}
Function	Turn on or off the beeper.
Explanations	The instrument will make a sound when a system prompting message appears or any key is pressed after the beeper is turned on.
Example	SYST:BEEP:IMM ON
4. SYSTem:BEEPer[:IMMediate]?	
Command Format	SYSTem:BEEPer[:IMMediate]?
Function	Query whether the beeper is enabled.
Return Value	The query returns ON or OFF.
5. SYSTem:BRIGht	
Command Format	SYSTem:BRIGht <brightness>

Function	Adjust the LCD brightness of the instrument.
Explanations	The available range of <brightness> is: 1~8.
Example	SYST:BRIG 5
6. SYSTem:BRIGht?	
Command Format	SYSTem:BRIGht?
Function	Query the LCD brightness of the instrument.
Return Value	Such as: 6
7. SYSTem:POWEron	
Command Format	SYSTem:POWEron {DEF LAST}
Function	Setup the power-on mode of system.
Explanations	<ul style="list-style-type: none"> ● "DEF": denotes the system will be restored to factory defaults when restart the instrument. ● "LAST": denotes the system will be restored to the settings before last power-off.
Example	SYST:POWE DEF
8. SYSTem:POWEron?	
Command Format	SYSTem:POWEron?
Function	Query the power-on mode of system.
Return Value	The query returns Last or Default.
9. SYSTem:LOCal	
Command Format	SYSTem:LOCal
Function	Setup the instrument as local mode.
Explanations	All keys on the front panel could be used normally in local mode.
Example	SYST:LOC
10. SYSTem:REMOte	
Command Format	SYSTem:REMOte
Function	Setup the instrument as remote mode.

Explanations	All keys on the front panel are disabled in remote mode except for "Local" key (numeric key 7).
Example	SYST:REM
11. SYSTem:COMMunicate:LAN:DHCP[:STATe]	
Command Format	SYSTem:COMMunicate:LAN:DHCP[:STATe] {ON OFF}
Function	Enable or disable DHCP mode.
Example	SYST:COMM:LAN:DHCP:STAT ON
12. SYSTem:COMMunicate:LAN:DHCP[:STATe]?	
Command Format	SYSTem:COMMunicate:LAN:DHCP[:STATe]?
Function	Query whether the DHCP mode is enabled or not.
Return Value	The query returns ON or OFF.
13. SYSTem:COMMunicate:LAN:AUTOip[:STATe]	
Command Format	SYSTem:COMMunicate:LAN:AUTOip[:STATe] {ON OFF}
Function	Enable or disable AUTOip mode.
Example	SYST:COMM:LAN:AUTO:STAT ON
14. SYSTem:COMMunicate:LAN:AUTOip[:STATe]?	
Command Format	SYSTem:COMMunicate:LAN:AUTOip[:STATe]?
Function	Query whether the AUTOip mode is enabled or not.
Return Value	The query returns ON or OFF.
15. SYSTem:COMMunicate:LAN:MANualip[:STATe]	
Command Format	SYSTem:COMMunicate:LAN:MANualip[:STATe] {ON OFF}
Function	Enable or disable MANualip mode.
Example	SYST:COMM:LAN:MAN:STAT ON
16. SYSTem:COMMunicate:LAN:MANualip[:STATe]?	
Command Format	SYSTem:COMMunicate:LAN:MANualip[:STATe]?

Function	Query whether the MANualip mode is enabled or not.
Return Value	The query returns ON or OFF.
17. SYSTem:COMMunicate:LAN:IPADdress	
Command Format	SYSTem:COMMunicate:LAN:IPADdress <ip>
Function	Setup the IP address of the LAN.
Explanations	The new IP address does not go into effect until the command SYSTem:COMMunicate:LAN:APPLY has been executed after this command.
Example	SYST:COMM:LAN:IPAD 172.16.3.32
18. SYSTem:COMMunicate:LAN:IPADdress?	
Command Format	SYSTem:COMMunicate:LAN:IPADdress?
Function	Query the IP address of the LAN.
Return Value	Such as: 255.0.5.255
19. SYSTem:COMMunicate:LAN:SMASk	
Command Format	SYSTem:COMMunicate:LAN:SMASk <submask>
Function	Query the subnet mask of the LAN.
Explanations	The new subnet mask does not go into effect until the command SYSTem:COMMunicate:LAN:APPLY has been executed after this command.
Example	SYST:COMM:LAN:SMAS 255.255.255.0
20. SYSTem:COMMunicate:LAN:SMASk?	
Command Format	SYSTem:COMMunicate:LAN:SMASk?
Function	Query the subnet mask of the LAN.
Return Value	Such as: 255.255.255.0
21. SYSTem:COMMunicate:LAN:GATEway	
Command Format	SYSTem:COMMunicate:LAN:GATEway <gateway>
Function	Setup the gateway of the LAN.

Explanations	The new gateway does not go into effect until the command SYSTem:COMMunicate:LAN:APPLy has been executed after this command.
Example	SYST:COMM:LAN:GATE 172.16.3.1
22. SYSTem:COMMunicate:LAN:GATEway?	
Command Format	SYSTem:COMMunicate:LAN:GATEway?
Function	Query the gateway of the LAN.
Return Value	Such as: 172.16.3.1
23. SYSTem:COMMunicate:LAN:DNS	
Command Format	SYSTem:COMMunicate:LAN:DNS <dns>
Function	Setup the DNS server of the LAN.
Explanations	The new DNS server setting does not go into effect until the command SYSTem:COMMunicate:LAN:APPLy has been executed after this command.
Example	SYST:COMM:LAN:DNS 172.16.2.3
24. SYSTem:COMMunicate:LAN:DNS?	
Command Format	SYSTem:COMMunicate:LAN:DNS?
Function	Query the DNS server of the LAN.
Return Value	Such as: 172.16.2.3
25. SYSTem:COMMunicate:GPIB:ADDRess	
Command Format	SYSTem:COMMunicate:GPIB:ADDRess <gpib address>
Function	Setup the GPIB address of the instrument.
Explanations	The available range of <gpib address> is: 1~30.
Example	SYST:COMM:GPIB:ADDR 10
26. SYSTem:COMMunicate:GPIB:ADDRess?	
Command Format	SYSTem:COMMunicate:GPIB:ADDRess?
Function	Query the GPIB address of the instrument.

Return Value	Such as: GPIB Address:10
27. SYSTem:COMMunicate:LAN:APPLy	
Command Format	SYSTem:COMMunicate:LAN:APPLy
Function	Apply the selected parameters about LAN.
Example	SYST:COMM:LAN:APPL
28. SYSTem:SELF:TEST:AD?	
Command Format	SYSTem:SELF:TEST:AD?
Function	Query the self-test result of the AD converter.
Return Value	The query returns Pass or Error after self-test.
29. SYSTem:SELF:TEST:DA?	
Command Format	SYSTem:SELF:TEST:DA?
Function	Query the self-test result of the DA converter.
Return Value	The query returns Pass or Error after self-test.
30. SYSTem:SELF:TEST:FPGA?	
Command Format	SYSTem:SELF:TEST:FPGA?
Function	Query the self-test result of the FPGA module.
Return Value	The query returns Pass or Error after self-test.
31. SYSTem:SELF:TEST:USB?	
Command Format	SYSTem:SELF:TEST:USB?
Function	Query the self-test result of the USB interface.
Return Value	The query returns Pass or Error after self-test.
32. SYSTem:SELF:TEST:LAN?	
Command Format	SYSTem:SELF:TEST:LAN?
Function	Query the self-test result of the LAN.
Return Value	The query returns Pass or Error after self-test.

33. SYSTem:SELF:TEST:FAN?	
Command Format	SYSTem:SELF:TEST:FAN?
Function	Query the self-test result of the fan.
Return Value	The query returns Pass or Error after self-test.
34. SYSTem:OTP	
Command Format	SYSTem:OTP {ON OFF}
Function	Enable or disable OTP function.
Example	SYST:OTP OFF
35. SYSTem:OTP?	
Command Format	SYSTem:OTP?
Function	Query whether the OTP is enabled or not.
Return Value	The query returns On or OFF.

STORe/RECAI Commands

STORe/RECAI commands save the system state into the internal or external memory or recall these states from assigned position in memory.

DP1308A supports following **STORe/RECAI** command:

1. STORe:LOCaI
2. STORe:EXTErnal
3. RECAI:LOCaI
4. RECAI:EXTErnal

The detailed information of each command are:

1. STORe:LOCaI	
Command Format	STORe:LOCaI {1 2 3 4},<name>
Function	Save the current system state into the internal or external nonvolatile memory as assigned name like *SAV command.
Explanations	4 memory locations are provided by DP1308A for saving the instrument state (number "1", "2", "3", "4").
Example	STOR:LOC 1,RIGOL
2. STORe:EXTErnal	
Command Format	STORe:EXTErnal <name>
Function	Save the current system state into root directory of the external memory as assigned name.
Example	STOR:EXTE RIGOL
3. RECAI:LOCaI	
Command Format	RECAI:LOCaI {1 2 3 4}
Function	Recall the file of system state from the assigned position of internal memory like *RCL command.
Example	RECA:LOC 2
4. RECAI:EXTErnal	

Command Format	RECALL:EXTERnal <name>
Function	Recall the file of system state in the root directory of the external memory.
Example	RECA:EXTE RIGOL

Chapter 3 Programming Examples

This chapter lists some programming examples in the development environments of Visual C++ 6.0 and Visual Basic 6.0. 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 DP1308A and PC via GPIB, USB, 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 following parts:

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

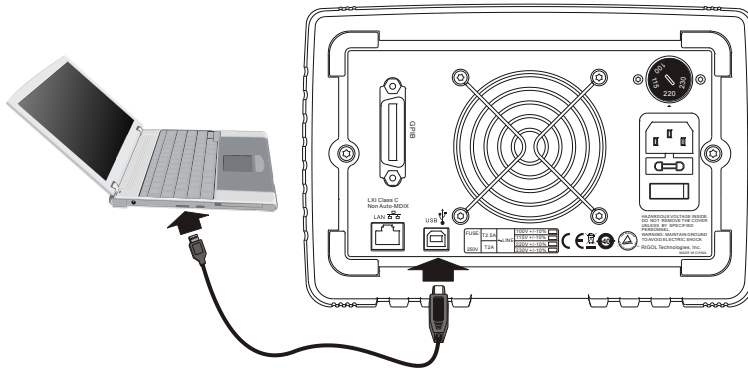
This chapter contains following topics:

- Prepare for Programming
- Program in Visual C++ 6.0
- Program in Visual Basic 6.0

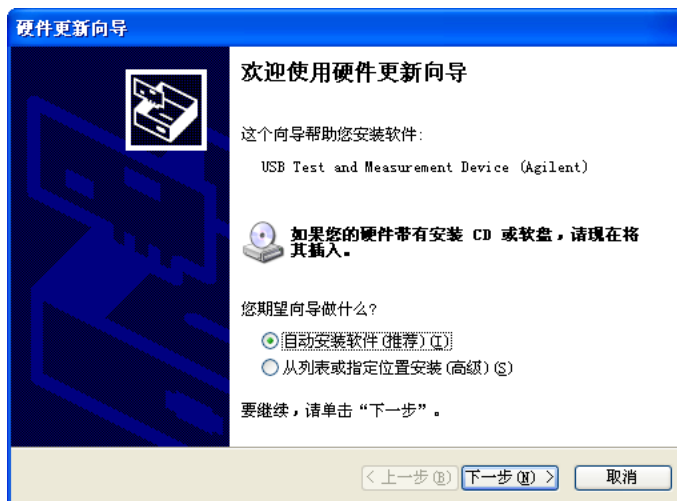
Prepare for Programming

First verify 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 DP1308A 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 (Agilent)" on the PC. See the figure below:

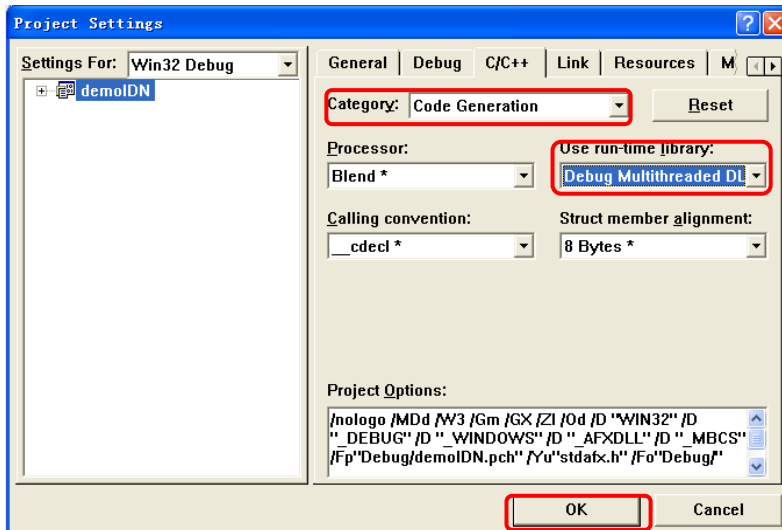


At present, you have finished the preparations. Next, we will give you some programming examples in Visual C++ 6.0 and Visual Basic 6.0.

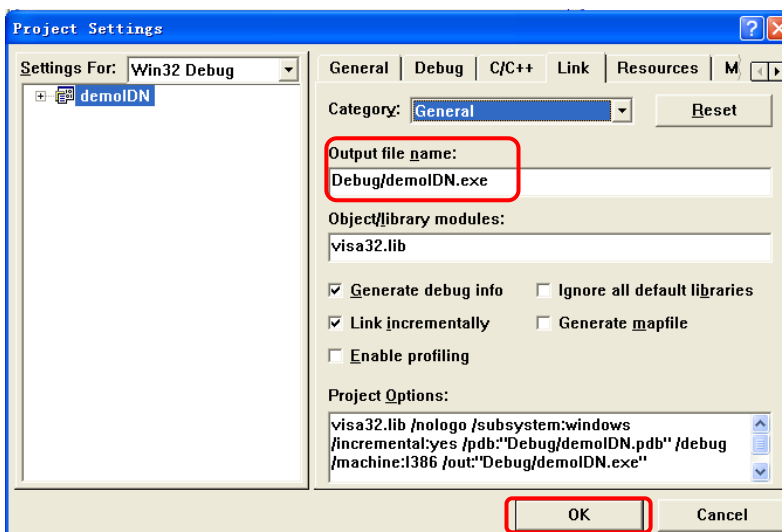
Program in Visual C++ 6.0

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**.



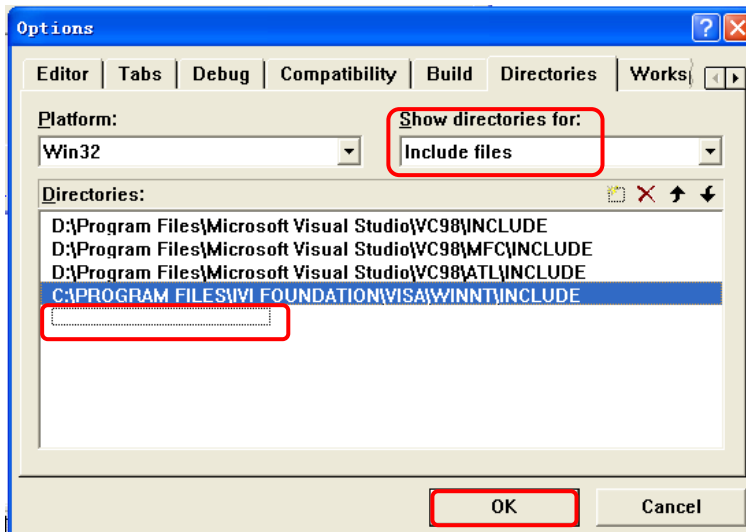
3. Choose **Project**→**Settings**→**Link**, add the file **"visa32.lib"** manually in **Object/library modules**.



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.

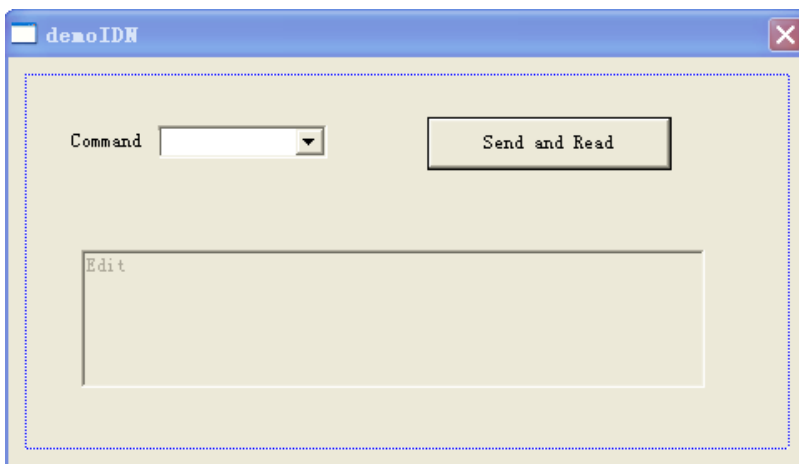
Select **"Library files"** in **Show directories for**, and then dblclick the blank in **Directories** to add the path of **"Lib"**:

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.



- 1) Name the Text "**Command**";
- 2) Choose **Data** in the property of **Com box** and input *IDN? commands manually;
- 3) Choose **General** in the property of **Edit** and select **Disable**.
- 4) Name Button "Send and Read" and dbclick it to 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);

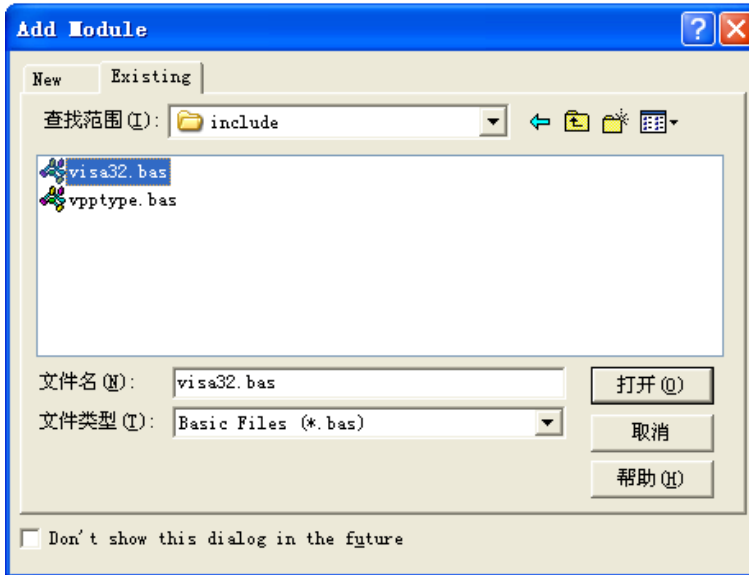
// display the results
UpdateData (TRUE);
m_receive = buf;
UpdateData (FALSE);
viClose (vi);
viClose (defaultRM);
```

6. Save, build and run the project, you will get an EXE file. When the DP1308A has been successfully connected with PC, choose ***IDN?** and click "**Send and Read**", the instrument will return the result.

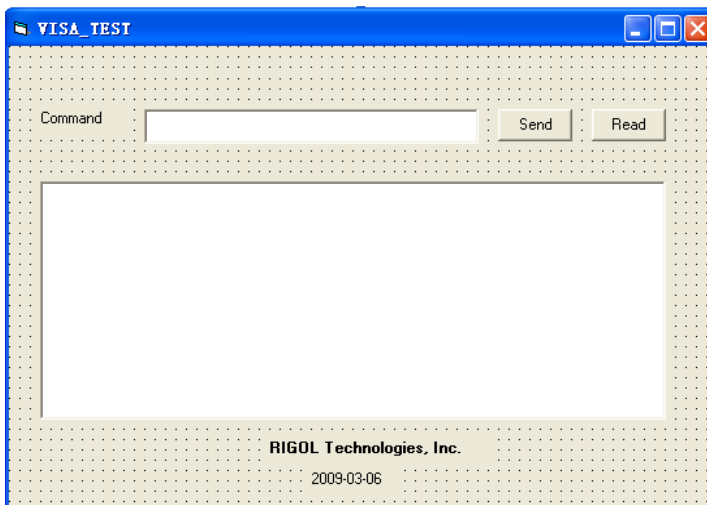
Program in Visual Basic 6.0

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 desired **Text Box** and **CommandButton** in the **Form** window to get the following interface:



4. Enter the programming environment and add the following codes:

```
Private Sub Form_Load()

Dim stat      As ViStatus
Dim dfltRM    As ViSession
Dim sesn      As ViSession
Dim fList     As ViFindList
Dim rsrcName  As String * VI_FIND_BUFLen
Dim instrDesc As String * VI_FIND_BUFLen
Dim nList     As Long
Dim i As Integer
stat = viOpenDefaultRM(dfltRM)
If (stat < VI_SUCCESS) Then
    Rem Error initializing VISA ..... exiting
    Exit Sub
End If
Rem Find all Serial instruments in the system
stat = viFindRsrc(dfltRM, "?*", fList, nList, rsrcName)

If (stat < VI_SUCCESS) Then
    Rem Error finding resources ..... exiting
    viClose (dfltRM)
    Exit Sub
End If

Combo1.AddItem rsrcName
viDeviceID(0) = rsrcName

i = 1

While (nList)
    stat = viFindNext(fList, rsrcName)

    If (stat < VI_SUCCESS) Then
        Exit Sub
    Else
        Combo1.AddItem rsrcName
        viDeviceID(i) = rsrcName
    End If
End While
```

```
    nList = nList - 1
    i = i + 1
```

```
Wend
```

```
End Sub
```

```
Public Function SendToDevice(ByVal DeviceStr As String, RmtCmd As String)
```

```
    Dim stat As ViStatus
```

```
    Dim dfltRM As ViSession
```

```
    Dim sesn As ViSession
```

```
    Dim retCount As Long
```

```
    Dim cmdLen As Integer
```

```
    Dim fList As ViFindList
```

```
    Dim rsrcName As String * VI_FIND_BUFLen
```

```
    Dim instrDesc As String * VI_FIND_BUFLen
```

```
    Dim nList As Long
```

```
    cmdLen = Len(RmtCmd)
```

```
    Rem Begin by initializing the system
```

```
    stat = viOpenDefaultRM(dfltRM)
```

```
    If (stat < VI_SUCCESS) Then
```

```
        Rem Error initializing VISA...exiting
```

```
        Exit Function
```

```
    End If
```

```
    Rem Open communication with IO Device
```

```
    Rem NOTE: For simplicity, we will not show error checking
```

```
    stat = viOpen(dfltRM, DeviceStr, VI_NULL, VI_NULL, sesn)
```

```
    Rem Set the timeout for message-based communication
```

```
    stat = viSetAttribute(sesn, VI_ATTR_TMO_VALUE, 5000)
```

```
    stat = viWrite(sesn, RmtCmd, cmdLen, retCount)
```

```
    stat = viClose(sesn)
```

```
    stat = viClose(dfltRM)
```

End Function

```
Public Function ReadFromDevice(ByVal DeviceStr As String, RmtCmd As String) As String
```

```
Const MAX_CNT = 200
```

```
Dim stat As ViStatus
```

```
Dim dfltRM As ViSession
```

```
Dim sesn As ViSession
```

```
Dim retCount As Long
```

```
Dim buffer As String * MAX_CNT
```

```
Dim cmdLen As Integer
```

```
cmdLen = Len(RmtCmd)
```

```
Rem Begin by initializing the system
```

```
stat = viOpenDefaultRM(dfltRM)
```

```
If (stat < VI_SUCCESS) Then
```

```
Rem Error initializing VISA...exiting
```

```
Exit Function
```

```
End If
```

```
Rem Open communication with IO Device
```

```
Rem NOTE: For simplicity, we will not show error checking
```

```
stat = viOpen(dfltRM, DeviceStr, VI_NULL, VI_NULL, sesn)
```

```
Sleep (100)
```

```
Rem Set the timeout for message-based communication
```

```
stat = viSetAttribute(sesn, VI_ATTR_TMO_VALUE, 5000)
```

```
Sleep (100)
```

```
stat = viWrite(sesn, RmtCmd, cmdLen, retCount)
```

```
'DoEvents
```

```
Sleep (500)
```

```
stat = viRead(sesn, buffer, MAX_CNT, retCount)
```

```
'DoEvents
```

```
ReadFromDevice = buffer
```

```
Sleep (100)
```

```
stat = viClose(sesn)
```

```
stat = viClose(dfltRM)
```

End Function

5. Save and run the project, you will get a command input interface. After you connect the instrument with PC successfully, the instrument could be controlled easily via commands.

Command Quick Reference A-Z

*IDN? 2-2
 *RCL 2-3
 *RST 2-2
 *SAV 2-2
 *TST? 2-2

A

APPLy 2-4
 APPLy? 2-4

D

DISPlay:FOCUs 2-15

I

INSTRument[:SELEct] 2-5
 INSTRument[:SELEct]? 2-5
 INSTRument:NSElect 2-5
 INSTRument:NSElect? 2-5

M

MEASure:CURRent[:DC]? 2-7
 MEASure:POWEr[:DC]? 2-7
 MEASure[:VOLTagE][:DC]? 2-7

O

OUTPut:OCP 2-10
 OUTPut:OCP:STATe 2-9
 OUTPut:OCP:STATe? 2-10
 OUTPut:OCP? 2-10
 OUTPut:OVP 2-9
 OUTPut:OVP:STATe 2-9
 OUTPut:OVP:STATe? 2-9
 OUTPut:OVP? 2-9

OUTPut:TIMEr 2-11
 OUTPut:TIMEr:STATe 2-12
 OUTPut:TIMEr:STATe? 2-12
 OUTPut:TIMEr? 2-12
 OUTPut:TRACK 2-10
 OUTPut:TRACK? 2-11
 OUTPut:WAVE 2-11
 OUTPut:WAVE? 2-11
 OUTPut[:STATe] 2-8
 OUTPut[:STATe]? 2-8

R

RECALL:EXTErnal 2-24
 RECALL:LOCAl 2-24

S

[SOURce:]CURRent[:LEVel][:IMMediate][:AMP
 Litude] 2-13
 [SOURce:]CURRent[:LEVel][:IMMediate][:AMP
 Litude]? 2-13
 [SOURce:]VOLTagE[:LEVel][:IMMediate][:AMP
 Litude] 2-13
 [SOURce:]VOLTagE[:LEVel][:IMMediate][:AMP
 Litude]? 2-14
 STORe:EXTErnal 2-24
 STORe:LOCAl 2-24
 SYSTem:BEEPer[:IMMediate] 2-17
 SYSTem:BEEPer[:IMMediate]? 2-17
 SYSTem:BRIGHt 2-17
 SYSTem:BRIGHt? 2-18
 SYSTem:COMMunicate:GPIB:ADDReSS 2-21
 SYSTem:COMMunicate:GPIB:ADDReSS? 2-21
 SYSTem:COMMunicate:LAN:APPLy 2-22
 SYSTem:COMMunicate:LAN:AUTOIp[:STATe]

2-19	SYSTem:COMMunicate:LAN:SMASk? 2-20
SYSTem:COMMunicate:LAN:AUTOip[:STATe]? 2-19	SYSTem:LANGuage:TYPE 2-17
SYSTem:COMMunicate:LAN:DHCP[:STATe] 2-19	SYSTem:LANGuage:TYPE? 2-17
SYSTem:COMMunicate:LAN:DHCP[:STATe]? 2-19	SYSTem:LOCal 2-18
SYSTem:COMMunicate:LAN:DNS 2-21	SYSTem:OTP 2-23
SYSTem:COMMunicate:LAN:DNS? 2-21	SYSTem:OTP? 2-23
SYSTem:COMMunicate:LAN:GATEWay 2-20	SYSTem:POWEron 2-18
SYSTem:COMMunicate:LAN:GATEWay? 2-21	SYSTem:POWEron? 2-18
SYSTem:COMMunicate:LAN:IPADdress 2-20	SYSTem:REMOte 2-18
SYSTem:COMMunicate:LAN:IPADdress? 2-20	SYSTem:SELF:TEST:AD? 2-22
SYSTem:COMMunicate:LAN:MANualip[:STATe] 2-19	SYSTem:SELF:TEST:DA? 2-22
SYSTem:COMMunicate:LAN:MANualip[:STATe]? 2-19	SYSTem:SELF:TEST:FAN? 2-23
SYSTem:COMMunicate:LAN:SMASk 2-20	SYSTem:SELF:TEST:FPGA? 2-22
	SYSTem:SELF:TEST:USB? 2-22
	SYSTem:SELF:TEST:LAN? 2-22