

# **RIGOL**

## **Programming Guide**

### **DP1308A Programmable Linear DC Power Supply**

**July 2011**

**RIGOL Technologies, Inc.**



# Guaranty and Declaration

## Copyright

© 2009 RIGOL Technologies, Inc. All Rights Reserved.

## Trademark Information

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

## Publication Number

PGH01104-1110

## Notices

- **RIGOL** products are protected by patent law in and outside of P.R.C.
- **RIGOL** Technologies, Inc. reserves the right to modify or change parts of or all the specifications and pricing policies at company's sole decision.
- Information in this publication replaces all previously corresponding material.
- **RIGOL** shall not be liable for losses caused by either incidental or consequential in connection with the furnishing, use or performance of this manual as well as any information contained.
- Any part of this document is forbidden to be copied or photocopied or rearranged without prior written approval of **RIGOL**.

## Product Certification

**RIGOL** guarantees this product conforms to the national and industrial standards in China. International standard conformance certification is in progress, e.g. ISO. At present, DP1308A has passed CE, cTUVus and LXI certification.

## Contact Us

If you have any problem or requirement when using our products, please contact **RIGOL** Technologies, Inc. or your local distributors, or visit: [www.rigol.com](http://www.rigol.com)

# Document Overview

## Chapter 1 Programming Overview

This chapter introduces how to realize remote control of DP1308A 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 how to realize the common functions of DP1308A 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.rigol.com> to download.

---

# Contents

|   |            |
|---|------------|
| <b>Guaranty and Declaration .....</b>         | <b>I</b>   |
| <b>Document Overview.....</b>                 | <b>II</b>  |
| <b>Chapter 1 Programming Overview .....</b>   | <b>1-1</b> |
| 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        |
| <b>Chapter 2 DP1308A Command Systems.....</b> | <b>2-1</b> |
| Common Commands.....                          | 2-2        |
| APPLy Commands.....                           | 2-4        |
| INSTrument Commands .....                     | 2-5        |
| MEASure Commands.....                         | 2-6        |
| OUTPut Commands .....                         | 2-7        |
| SOURce Commands.....                          | 2-12       |
| DISPlay Commands .....                        | 2-14       |
| SYSTEM Commands.....                          | 2-15       |
| STORe/RECAll Commands .....                   | 2-23       |
| <b>Chapter 3 Programming Examples.....</b>    | <b>3-1</b> |
| Prepare for Programming .....                 | 3-2        |
| Programming in Visual C++ 6.0.....            | 3-4        |
| Programming in Visual Basic 6.0 .....         | 3-7        |
| <b>Command Quick Reference A-Z.....</b>       | <b>1</b>   |



# Chapter 1 Programming Overview

This chapter introduces 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 DP1308A include:

- Setup the DP1308A
- Output programmable linear DC power supply

### Connection:

It is recommended to use the USB interface to connect the instrument to PC.

Please connect the USB Device port at the rear panel 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 based on ASCII for test and measurement instrument. The commands present a hierarchy structure (tree system) and contain different sub-systems that are distinguished by different root keywords. Each command consists of a "Root" keyword and one or multiple sub-keywords. The keywords are separated by ":" and are followed by the parameter settings available, "?" is added at the end of the command string to indicate query and the command and parameter are separated by "space".

For example,

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

```
SYSTem:LANGUage:TYPE?
```

**SYSTem** is the root keyword of this command, **LANGUage** and **TYPE** are second-level and third-level keywords, respectively, all of these keywords are separated by ":". Characters enclosed in the "{}" are the parameters available. "space" is used to divide the command keyword SYSTem:LANGUage:TYPE and the parameter {EN|CH}. SYSTem:LANGUage:TYPE? denotes 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 contained in a command 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}
```

Users 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 truning on (enable) the focus display, "OFF" denoets turning 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 parameter can be any value within the effective range and with the required precision. For example,

```
CURRent {<current>|MINimum|MAXimum}
```

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

### 5. Discrete

The parameter 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 the combinations of ASCII characters. For example,  
STORe:LOCAl {1|2|3|4} <name>.

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

## Command Abbreviation

According to SCPI syntax, most of commands consists of both uppercase and lowercase letters, 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 abbreviation is used, all 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 of each command supported by DP1308A, including command syntax, function description, using considerations as well as some application examples.

DP1308A contains following commands subsystems:

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

## Common Commands

IEEE standard defines common commands for querying basic instrument information or executing basic operations of the instrument. Those commands usually begin with "\*" and holds 3-character long command keyword.

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

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

Detailed information of each **IEEE488.2** command:

| <b>1. *IDN?</b> |   |
|-----------------|---|
| Syntax          | *IDN?   |
| Function        | Query ID character string of the instrument, including 4 character segments separated by commas ",": manufacturer, model number, serial number and the edition number that consists of numbers separated by "." . |
| Return Value    | Rigol Technologies,DP1308A,DP1A110300105,00.01.00.00.01.02.01.01.03.00  |
| <b>2. *RST</b>  |   |
| Syntax          | *RST  |
| Function        | Reset the system to the internally specified state.   |
| <b>3. *TST?</b> |   |
| Syntax          | *TST?   |
| Function        | Query the results of self-test.   |
| Return Value    | The query returns Pass or Error.  |
| <b>4. *SAV</b>  |   |
| Syntax          | *SAV {1 2 3 4},<name>   |
| Function        | Save the current system state to the nonvolatile memory with the name assigned by <name>.   |
| Explanation     | 4 store locations are available for storing the state of the  |

|                |  |
|----------------|--|
|                | instrument. (number "1", "2", "3" and "4") |
| <b>5. *RCL</b> |  |
| Syntax         | *RCL {1 2 3 4}                             |
| Function       | Recall the stored instrument state.        |

## APPLY Commands

**APPLY** commands are used for quickly choosing the channel to be measured and its voltage and current values, making the control over remote interface more easier.

DP1308A supports following **APPLY** commands:

1. APPLY
2. APPLY?

Detailed information of each command:

| <b>1. APPLY</b>  |  |
|------------------|--|
| Syntax           | APPLY {P6V P25V N25V}{[,<volt> DEFault MINimum MAXimum}<br>[,<curr> DEFault MINimum MAXimum}]  |
| Function         | Set the voltage and current for a channel.   |
| Explanations     | If only one parameter is set, <volt> is set.<br>If no parameters is set, the command selects the channel (the same with the <b>INSTrument[:SELEct]</b> command).   |
| Example          | <ol style="list-style-type: none"> <li>1. APPL P25V,10,0.5 Set the voltage and current on +25V channel as 10 V and 0.5 A.</li> <li>2. APPL P6V,6 set the voltage on +6V channel as +6V.</li> <li>3. APPL N25V Select -25V channel as the current channel.</li> </ol> |
| <b>2. APPLY?</b> |  |
| Syntax           | APPLY? [{P6V P25V N25V}]   |
| Function         | Query the setting values of voltage and current of the specified channel.  |
| Explanations     | If no channel is specified, the query returns the voltage and current of the current channel.  |
| Example          | P25V,Limit,10.0000V,0.5000A  |



## INSTRument Commands

Being different with **Apply** commands, **INSTRument** commands provides more flexibility in channel selection, set 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?

Detailed information of each command:

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

Detailed information of each command:

| <b>1. MEASure:CURRent[:DC]?</b>   |  |
|-----------------------------------|--|
| Syntax                            | 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 the channel under operation. |
| Return Value                      | Such as: 0.0530  |
| <b>2. MEASure[:VOLTage][:DC]?</b> |  |
| Syntax                            | 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: 3.3310  |
| <b>3. MEASure:POWEr[:DC]?</b>     |  |
| Syntax                            | 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: 0.0600  |

## 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?

Detailed information of each command:

| <b>1. OUTPut[:STATe]</b>  |   |
|---------------------------|---|
| Syntax                    | OUTPut[:STATe] {P6V P25V N25V},{OFF ON}           |
| Function                  | Turn on or off the specified channel.             |
| Example                   | OUTP:STAT N25V, ON                                |
|                           |   |
| <b>2. OUTPut[:STATe]?</b> |   |
| Syntax                    | OUTPut[:STATe]? {P6V P25V N25V}                   |
| Function                  | Query whether the specified channel is turned on. |
| Return Value              | The query returns ON or OFF.                      |

| <b>3. OUTPut:OVP:STATe</b>  |   |
|-----------------------------|---|
| Syntax                      | OUTPut:OVP:STATe {P6V P25V N25V},{ON OFF}   |
| Function                    | Enable or disable the overvoltage protection function (OVP) of the specified channel.   |
| Explanation                 | This command is only available for channels currently turned on.  |
| Example                     | OUTP:OVP:STAT N25V,ON   |
| <b>4. OUTPut:OVP:STATe?</b> |   |
| Syntax                      | 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.  |
| <b>5. OUTPut:OVP</b>        |   |
| Syntax                      | OUTPut:OVP {P6V P25V N25V},<value>  |
| Function                    | Set the overvoltage protection value (OVP) of the specified channel.  |
| Explanations                | <value> denotes the overvoltage protection value, the range of which varies on different channels:<br>1. On +6V channel: 0.1V~6.5V;<br>2. On +25V channel: 0.1V~27V;<br>3. On -25V channel: -0.1V~-27V. |
| Example                     | OUTP:OVP P6V,2  |
| <b>6. OUTPut:OVP?</b>       |   |
| Syntax                      | OUTPut:OVP? {P6V P25V N25V}   |
| Function                    | Query the overvoltage protection value (OVP) of the specified channel.  |
| Return Value                | Such as: -27.0  |
| <b>7. OUTPut:OCP:STATe</b>  |   |
| Syntax                      | OUTPut:OCP:STATe {P6V P25V N25V},{ON OFF}   |
| Function                    | Enable or disable the overcurrent protection function (OCP) of the specified channel.   |
| Explanation                 | This command is only available for channels currently turned on.  |
| Example                     | OUTP:OCP:STAT P25V,ON   |

| <b>8. OUTPut:OCP:STATe?</b> |   |
|-----------------------------|---|
| Syntax                      | 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.  |
| <b>9. OUTPut:OCP</b>        |   |
| Syntax                      | OUTPut:OCP {P6V P25V N25V},<value>  |
| Function                    | Set the overcurrent protection value (OCP) of the specified channel.  |
| Explanations                | <value> denotes the overcurrent protection value, the range of which varies on different channels: <ol style="list-style-type: none"> <li>1. On +6V channel: 0.1A~5.5A;</li> <li>2. On +25V channel: 0.1A~1.2A;</li> <li>3. On -25V channel: 0.1A~1.2A.</li> </ol>  |
| Example                     | OUTP:OCP N25V,1   |
| <b>10. OUTPut:OCP?</b>      |   |
| Syntax                      | OUTPut:OCP? {P6V P25V N25V}   |
| Function                    | Query the overcurrent protection value (OCP) of the specified channel.  |
| Return Value                | Such as: 1.0  |
| <b>11. OUTPut:TRACk</b>     |   |
| Syntax                      | OUTPut:TRACk {P25V N25V OFF}  |
| Function                    | Set the track state of the specified channel.   |
| Explanations                | <ul style="list-style-type: none"> <li>● Track function is available only on +25V and -25V channels.</li> <li>● The voltages of <math>\pm 25V</math> channels would be the same when the track function is enabled for the first time. In track mode, the voltage of the corresponding tracking channel varies with the voltage of the channel being tracked, which also applies to the voltage output value if both +25V and -25V channel are in CV mode.</li> <li>● To exit track mode, please select "OFF".</li> </ul> |
| Example                     | OUTP:TRAC P25V  |
| <b>12. OUTPut:TRACk?</b>    |   |
| Syntax                      | OUTPut:TRACk?   |

|                               |  |
|-------------------------------|--|
| Function                      | Query the track state of the specified channel.  |
| Return Value                  | Such as: TRACK_P25_ON  |
| <b>13. OUTPut:WAVE</b>        |  |
| Syntax                        | 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  |
| <b>14. OUTPut:WAVE?</b>       |  |
| Syntax                        | 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.   |
| <b>15. OUTPut:TIMER</b>       |  |
| Syntax                        | OUTPut:TIMER<br>{P6V P25V N25V},<secnum>,<volt>,<curr>,<time>  |
| Function                      | Set 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 timing outputs could be set per channel, the range of <secnum> is 1~5.    |
| Example                       | OUTP:TIME P25V,2,5,0.5,2   |
| <b>16. OUTPut:TIMER?</b>      |  |
| Syntax                        | 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:<br><secnum>, <volt>, <curr>, <time>; such as:<br>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; |
| <b>17. OUTPut:TIMER:STATE</b> |  |
| Syntax                        | OUTPut:TIMER:STATE {P6V P25V N25V},{OFF ON}  |
| Function                      | Turn off or on the timing output of the specified channel.   |
| Explanation                   | This command is available only when <secnum> in the  |

|                                |  |
|--------------------------------|--|
|                                | <b>OUTPut:TIMER</b> command starts from 1.   |
| Example                        | OUTP:TIME:STAT P6V,ON  |
| <b>18. OUTPut:TIMER:STATE?</b> |  |
| Syntax                         | 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. <ul style="list-style-type: none"> <li>● PAUSE: timing output is enabled but the corresponding channel is not enabled.</li> <li>● RUN: both the timing output and the corresponding channel are enabled.</li> <li>● OFF: neither of the two is enabled or the timing finishes.</li> </ul> |

## SOURce Commands

**SOURce** commands are used for setting the current and voltage 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]?

Detailed information of each command:

| <b>1. [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]</b>  |   |
|---|---|
| Syntax  | [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]<br>{<current> MINimum MAXimum}   |
| Function  | Set the current value of the current channel.   |
| Explanations  | <ul style="list-style-type: none"> <li>● The current of the current channel would change to the setting value as soon as this command was received successfully.</li> <li>● Select "&lt;current&gt;" to set the current value manually; select "MINimum" to set the minimum and "MAXimum" to set the maximum.</li> <li>● Different channels have different current ranges.<br/>On +6V channel: 0~5.25A;<br/>On ±25V channel: 0~1.05A</li> </ul> |
| Example   | SOUR:CURR:LEV:IMM:AMPL 0.5  |
| <b>2. [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?</b> |   |
| Syntax  | [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]? [MINimum   MAXimum]   |
| Function  | Query the setting value of the current of the channel under operation.  |
| Explanations  | <ul style="list-style-type: none"> <li>● The query returns the setting value of the current of the channel under operation once the command is received.</li> <li>● Select "MINimum" to query and return the minimum value and "MAXimum" to query and return the maximum.</li> </ul>  |



|   |   |
|---|---|
| Return Value  | Such as: P6V,Limit Current,5.0000A  |
| <b>3. [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]</b>  |   |
| Syntax  | [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]<br>{<voltage> MINimum  MAXimum}  |
| Function  | Set voltage value of the current channel.   |
| Explanations  | <ul style="list-style-type: none"> <li>● The voltage of the current channel would change to the setting value as soon as this command is received successfully.</li> <li>● Select "&lt;voltage&gt;" to set the voltage manually; select "MINimum" to set the minimum and "MAXimum" to set the maximum.</li> <li>● Different channels have different voltage ranges.<br/>On +6V: 0~+6.3V;<br/>On +25V: 0~+26.25V;<br/>On -25V: 0~-26.25V.</li> </ul> |
| Example   | SOUR:VOLT:LEV:IMM:AMPL MIN  |
| <b>4. [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]?</b> |   |
| Syntax  | [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]?<br>[MINimum   MAXimum]  |
| Function  | Query the setting value of the voltage of the current channel.  |
| Explanations  | <ul style="list-style-type: none"> <li>● The query returns the setting value of the voltage of the current channel once the command is received.</li> <li>● Select "MINimum" to query and return the minimum value and "MAXimum" to query and return the maximum.</li> </ul>  |
| Return Value  | Such as: P6V,Limit Voltage,0.0000V  |

## DISPlay Commands

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

DP1308A supports following **DISPlay** command:

1. DISPlay:FOCUS

| <b>1. DISPlay:FOCUS</b> |  |
|-------------------------|--|
| Syntax                  | 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:LAN:APPLY
26. SYSTem:COMMunicate:GPIB:ADDRes
27. SYSTem:COMMunicate:GPIB:ADDRes?
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?

Detailed information of each command:

|                                       |   |
|---------------------------------------|---|
| <b>1. SYSTem:LANGUage:TYPE</b>        |   |
| Syntax                                | SYSTem:LANGUage:TYPE {EN CH}  |
| Function                              | Choose the type of system language.   |
| Example                               | SYST:LANG:TYPE CH   |
| <b>2. SYSTem:LANGUage:TYPE?</b>       |   |
| Syntax                                | SYSTem:LANGUage:TYPE?   |
| Function                              | Query the system language selected.   |
| Return Value                          | The query returns English or Chinese.   |
| <b>3. SYSTem:BEEPPer[:IMMEDIATE]</b>  |   |
| Syntax                                | SYSTem:BEEPPer[: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  |
| <b>4. SYSTem:BEEPPer[:IMMEDIATE]?</b> |   |
| Syntax                                | SYSTem:BEEPPer[:IMMEDIATE]?   |
| Function                              | Query whether the beeper is enabled.  |
| Return Value                          | The query returns ON or OFF.  |
| <b>5. SYSTem:BRIGht</b>               |   |
| Syntax                                | SYSTem:BRIGht <brightness>  |
| Function                              | Adjust the LCD brightness of the instrument.  |
| Explanations                          | The available range of <brightness> is: 1~8.  |
| Example                               | SYST:BRIG 5   |
| <b>6. SYSTem:BRIGht?</b>              |   |

|  |   |
|--|---|
| Syntax   | SYSTem:BRIGht?  |
| Function                                       | Query the LCD brightness of the instrument.   |
| Return Value                                   | Such as: 6  |
| <b>7. SYSTem:POWEron</b>                       |   |
| Syntax   | SYSTem:POWEron {DEF LAST}   |
| Function                                       | Set the power-on mode of system.  |
| Explanations                                   | <ul style="list-style-type: none"> <li>● "DEF": denotes the system will be restored to factory defaults when restart the instrument.</li> <li>● "LAST": denotes the system will be restored to the settings before last power-off when restart the instrument.</li> </ul> |
| Example  | SYST:POWE DEF   |
| <b>8. SYSTem:POWEron?</b>                      |   |
| Syntax   | SYSTem:POWEron?   |
| Function                                       | Query the power-on mode of system.  |
| Return Value                                   | The query returns Last or Default.  |
| <b>9. SYSTem:LOCal</b>                         |   |
| Syntax   | SYSTem:LOCal  |
| Function                                       | Set the instrument as local mode.   |
| Explanations                                   | All keys on the front panel could be used normally in local mode.   |
| Example  | SYST:LOC  |
| <b>10. SYSTem:REMOte</b>                       |   |
| Syntax   | SYSTem:REMOte   |
| Function                                       | Set the instrument as remote mode.  |
| Explanations                                   | All keys on the front panel are disabled in remote mode except the "Local" key (numeric key 7).   |
| Example  | SYST:REM  |
| <b>11. SYSTem:COMMunicate:LAN:DHCP[:STATe]</b> |   |
| Syntax   | SYSTem:COMMunicate:LAN:DHCP[:STATe] {ON OFF}  |
| Function                                       | Set the state of DHCP mode.   |
| Explanation                                    | Execute the <b>SYSTem:COMMunicate:LAN:APPLY</b> command after sending this command to apply the newly set state.  |
| Example  | SYST:COMM:LAN:DHCP:STAT ON  |

| <b>12. SYSTem:COMMunicate:LAN:DHCP[:STATe]?</b>     |   |
|---|---|
| Syntax  | SYSTem:COMMunicate:LAN:DHCP[:STATe]?  |
| Function  | Query whether the DHCP mode is enabled or not.  |
| Return Value  | The query returns ON or OFF.  |
| <b>13. SYSTem:COMMunicate:LAN:AUTOip[:STATe]</b>    |   |
| Syntax  | SYSTem:COMMunicate:LAN:AUTOip[:STATe] {ON OFF}  |
| Function  | Set the state of AUTOip mode.   |
| Explanation   | Execute the <b>SYSTem:COMMunicate:LAN:APPLY</b> command after sending this command to apply the newly set state.  |
| Example   | SYST:COMM:LAN:AUTO:STAT ON  |
| <b>14. SYSTem:COMMunicate:LAN:AUTOip[:STATe]?</b>   |   |
| Syntax  | SYSTem:COMMunicate:LAN:AUTOip[:STATe]?  |
| Function  | Query whether the AUTOip mode is enabled or not.  |
| Return Value  | The query returns ON or OFF.  |
| <b>15. SYSTem:COMMunicate:LAN:MANualip[:STATe]</b>  |   |
| Syntax  | SYSTem:COMMunicate:LAN:MANualip[:STATe] {ON OFF}  |
| Function  | Set the state of MANualip mode.   |
| Explanation   | Execute the <b>SYSTem:COMMunicate:LAN:APPLY</b> command after sending this command to apply the newly set state.  |
| Example   | SYST:COMM:LAN:MAN:STAT ON   |
| <b>16. SYSTem:COMMunicate:LAN:MANualip[:STATe]?</b> |   |
| Syntax  | SYSTem:COMMunicate:LAN:MANualip[:STATe]?  |
| Function  | Query whether the MANualip mode is enabled or not.  |
| Return Value  | The query returns ON or OFF.  |
| <b>17. SYSTem:COMMunicate:LAN:IPADdress</b>         |   |
| Syntax  | SYSTem:COMMunicate:LAN:IPADdress <ip>   |
| Function  | Set the IP address of the LAN.  |
| Explanations  | <ul style="list-style-type: none"> <li>● This command is only available under manual IP mode.</li> <li>● &lt;ip&gt; is ASCII character string and its range is from 1.0.0.0 to 223.255.255.255 (except 127.nnn.nnn.nnn).</li> <li>● The new IP address does not go into effect until the command</li> </ul> |

|  |   |
|--|---|
|  | <b>SYSTem:COMMunicate:LAN:APPLY</b> has been executed after this command.   |
| Example                                      | SYST:COMM:LAN:IPAD 172.16.3.32  |
| <b>18. SYSTem:COMMunicate:LAN:IPADdress?</b> |   |
| Syntax                                       | SYSTem:COMMunicate:LAN:IPADdress?   |
| Function                                     | Query the IP address of the LAN.  |
| Return Value                                 | Such as: 255.0.5.255  |
| <b>19. SYSTem:COMMunicate:LAN:SMASK</b>      |   |
| Syntax                                       | SYSTem:COMMunicate:LAN:SMASK <submask>  |
| Function                                     | Set the subnet mask of the LAN.   |
| Explanations                                 | <ul style="list-style-type: none"> <li>● This command is only available under manual IP mode.</li> <li>● &lt;submask&gt; is ASCII character string and its range is from 1.0.0.0 to 255.255.255.255.</li> <li>● The new subnet mask does not go into effect until the command <b>SYSTem:COMMunicate:LAN:APPLY</b> has been executed after this command.</li> </ul>                      |
| Example                                      | SYST:COMM:LAN:SMAS 255.255.255.0  |
| <b>20. SYSTem:COMMunicate:LAN:SMASK?</b>     |   |
| Syntax                                       | SYSTem:COMMunicate:LAN:SMASK?   |
| Function                                     | Query the subnet mask of the LAN.   |
| Return Value                                 | Such as: 255.255.255.0  |
| <b>21. SYSTem:COMMunicate:LAN:GATEway</b>    |   |
| Syntax                                       | SYSTem:COMMunicate:LAN:GATEway <gateway>  |
| Function                                     | Set the gateway of the LAN.   |
| Explanations                                 | <ul style="list-style-type: none"> <li>● This command is only available under manual IP mode.</li> <li>● &lt;gateway&gt; is ASCII character string and its range is from 1.0.0.0 to 223.255.255.255 (except 127.nnn.nnn.nnn).</li> <li>● The new gateway does not go into effect until the command <b>SYSTem:COMMunicate:LAN:APPLY</b> has been executed after this command.</li> </ul> |
| Example                                      | SYST:COMM:LAN:GATE 172.16.3.1   |
| <b>22. SYSTem:COMMunicate:LAN:GATEway?</b>   |   |

|   |  |
|---|--|
| Syntax                                      | SYSTem:COMMunicate:LAN:GATEway?  |
| Function                                    | Query the gateway of the LAN.  |
| Return Value                                | Such as: 172.16.3.1  |
| <b>23. SYSTem:COMMunicate:LAN:DNS</b>       |  |
| Syntax                                      | SYSTem:COMMunicate:LAN:DNS <dns>   |
| Function                                    | Set the DNS server of the LAN.   |
| Explanations                                | <ul style="list-style-type: none"> <li>● This command is only available under manual IP mode.</li> <li>● &lt;dns&gt; is ASCII character string and its range is from 1.0.0.0 to 223.255.255.255 (except 127.nnn.nnn.nnn).</li> <li>● The new DNS server setting does not go into effect until the command <b>SYSTem:COMMunicate:LAN:APPLY</b> has been executed after this command.</li> </ul> |
| Example                                     | SYST:COMM:LAN:DNS 172.16.2.3   |
| <b>24. SYSTem:COMMunicate:LAN:DNS?</b>      |  |
| Syntax                                      | SYSTem:COMMunicate:LAN:DNS?  |
| Function                                    | Query the DNS server of the LAN.   |
| Return Value                                | Such as: 172.16.2.3  |
| <b>25. SYSTem:COMMunicate:LAN:APPLY</b>     |  |
| Syntax                                      | SYSTem:COMMunicate:LAN:APPLY   |
| Function                                    | Apply the selected parameters about LAN.   |
| Example                                     | SYST:COMM:LAN:APPL   |
| <b>26. SYSTem:COMMunicate:GPIB:ADDRess</b>  |  |
| Syntax                                      | SYSTem:COMMunicate:GPIB:ADDRess <gpib address>   |
| Function                                    | Set the GPIB address of the instrument.  |
| Explanations                                | The available range of <gpib address> is: 1~30.  |
| Example                                     | SYST:COMM:GPIB:ADDR 10   |
| <b>27. SYSTem:COMMunicate:GPIB:ADDRess?</b> |  |
| Syntax                                      | SYSTem:COMMunicate:GPIB:ADDRess?   |
| Function                                    | Query the GPIB address of the instrument.  |
| Return Value                                | Such as: GPIB Address:10   |
| <b>28. SYSTem:SELF:TEST:AD?</b>             |  |



|                                   |  |
|-----------------------------------|--|
| Syntax                            | 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. |
| <b>29. SYSTem:SELF:TEST:DA?</b>   |  |
| Syntax                            | 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. |
| <b>30. SYSTem:SELF:TEST:FPGA?</b> |  |
| Syntax                            | 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. |
| <b>31. SYSTem:SELF:TEST:USB?</b>  |  |
| Syntax                            | 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. |
| <b>32. SYSTem:SELF:TEST:LAN?</b>  |  |
| Syntax                            | SYSTem:SELF:TEST:LAN?                            |
| Function                          | Query the self-test result of the LAN.           |
| Return Value                      | The query returns Pass or Error after self-test. |
| <b>33. SYSTem:SELF:TEST:FAN?</b>  |  |
| Syntax                            | SYSTem:SELF:TEST:FAN?                            |
| Function                          | Query the self-test result of the fan.           |
| Return Value                      | The query returns Pass or Error after self-test. |
| <b>34. SYSTem:OTP</b>             |  |
| Syntax                            | SYSTem:OTP {ON OFF}                              |
| Function                          | Enable or disable OTP function.                  |
| Example                           | SYST:OTP OFF                                     |
| <b>35. SYSTem:OTP?</b>            |  |
| Syntax                            | SYSTem:OTP?                                      |

|              |  |
|--------------|--|
| Function     | Query whether the OTP is enabled or not. |
| Return Value | The query returns On or OFF.             |

## STORE/RECALL Commands

**STORE/RECALL** commands save the system state into the internal or external memory or recall these states from assigned position in memory.

DP1308A supports following **STORE/RECALL** commands:

1. STORE:LOCAL
2. STORE:EXTERNAL
3. RECALL:LOCAL
4. RECALL:EXTERNAL

Detailed information of each command:

|                           |  |
|---------------------------|--|
| <b>1. STORE:LOCAL</b>     |  |
| Syntax                    | STORE:LOCAL {1 2 3 4},<name>   |
| Function                  | Save the current system state into the internal nonvolatile memory with the name specified in <name>. This command has the same function as the <b>*SAV</b> command. |
| Explanations              | 4 memory locations are provided by DP1308A for saving the instrument state (number "1", "2", "3", "4").  |
| Example                   | STOR:LOC 1,RIGOL   |
| <b>2. STORE:EXTERNAL</b>  |  |
| Syntax                    | STORE:EXTERNAL <name>  |
| Function                  | Save the current system state into root directory of the external memory with the name specified in <name>.  |
| Example                   | STOR:EXTE RIGOL  |
| <b>3. RECALL:LOCAL</b>    |  |
| Syntax                    | RECALL:LOCAL {1 2 3 4}   |
| Function                  | Recall the file of system state from the assigned position of internal memory. This function has the same function as the <b>*RCL</b> command.                       |
| Example                   | RECA:LOC 2   |
| <b>4. RECALL:EXTERNAL</b> |  |
| Syntax                    | 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 of how to realize common functions of DP1308A 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" mentioned here is NI (National Instrument)-VISA. NI-VISA is an API written by NI based on VISA standard. Users 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

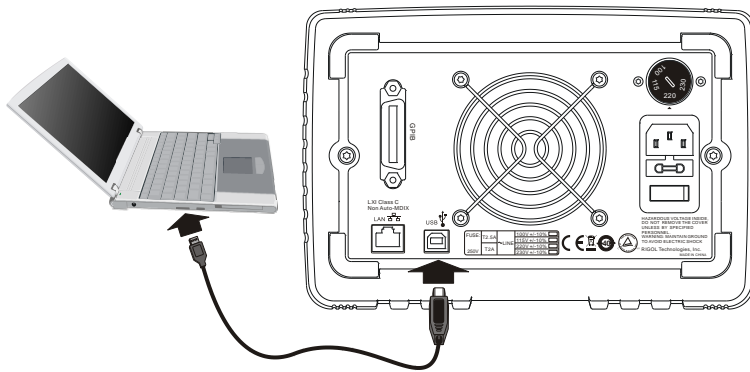
This chapter contains following topics:

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

## Prepare for Programming

First verify your computer has installed VISA library of NI (download it from <http://www.ni.com>). Here, the default installation path is C:\Program Files\IVI Foundation\VISA.

Here, the USB interface of DP1308A is used to communicate with the PC and please use the USB data cable to connect the USB Device interface at the rear panel of the instrument to the USB interface of the PC. See the figure below.



After successful connection, turn on the instrument and a "Found New Hardware Wizard" dialog box appears. Please follow the instructions to install the "USB Test and Measurement Device (Agilent)" on the PC. See the figure below.

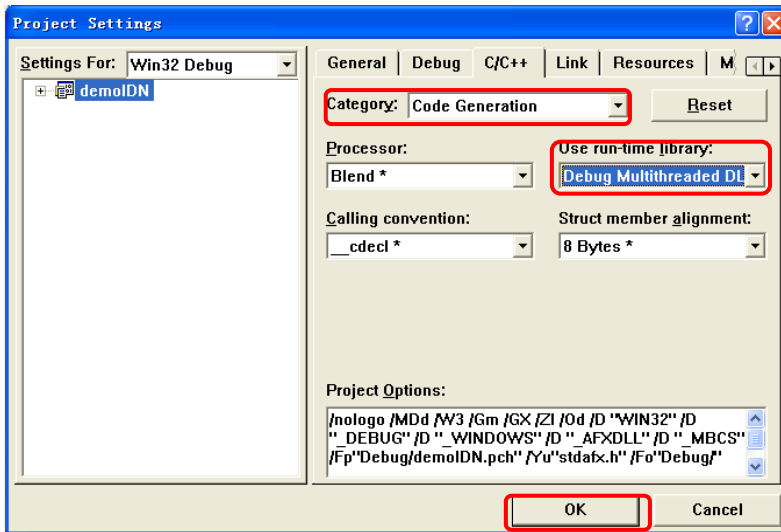


By now, the preparations for programming are finished. In the following part, detailed introductions are given about the programming demos in the Visual C++ 6.0 and Visual Basic 6.0 development environment.

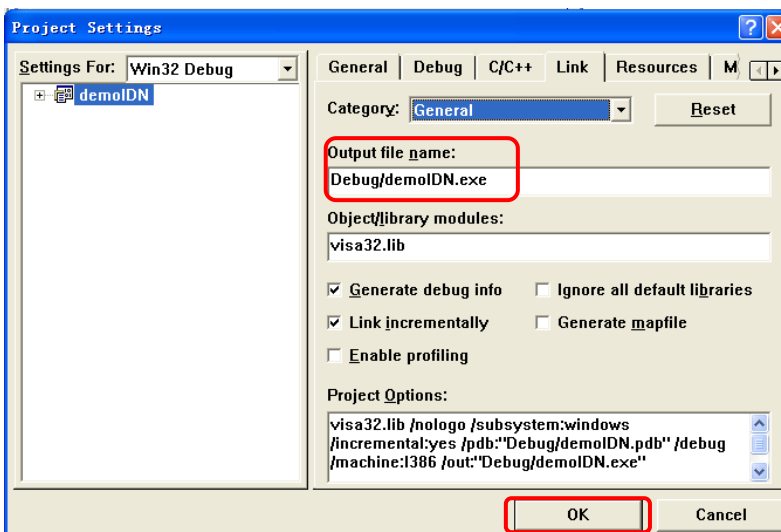
# Programming in Visual C++ 6.0

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**.

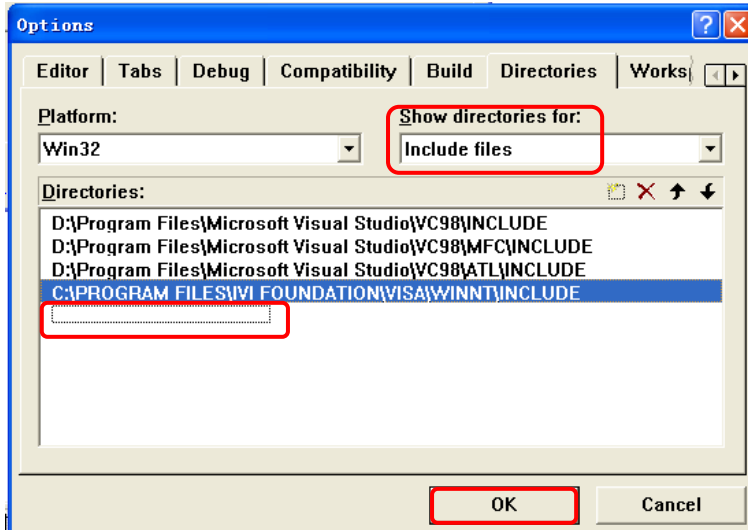




4. Choose **Tools** → **Options** → **Directories**; select **"Include files"** in **Show directories for**, and then dblclick the blank in **Directories** to add the path of **"Include"**: C:\Program Files\IVI Foundation\VISA\WinNT\include.

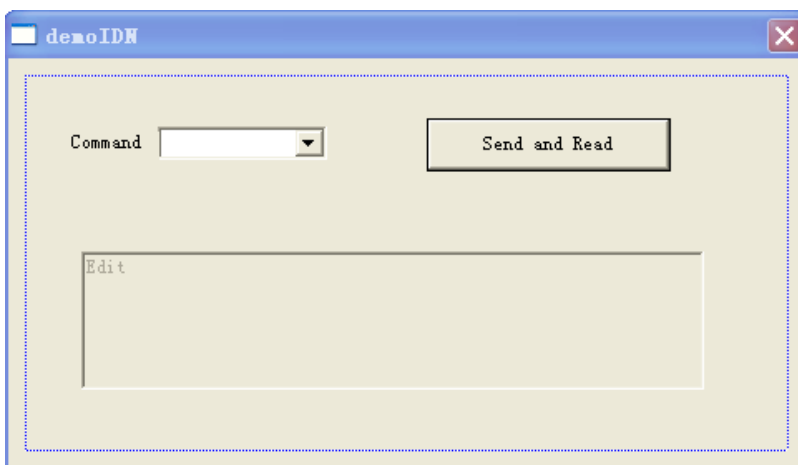
Select **"Library files"** in **Show directories for**, and then dblclick the blank in **Directories** to add the path of **"Lib"**:

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 the \*IDN? command 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);

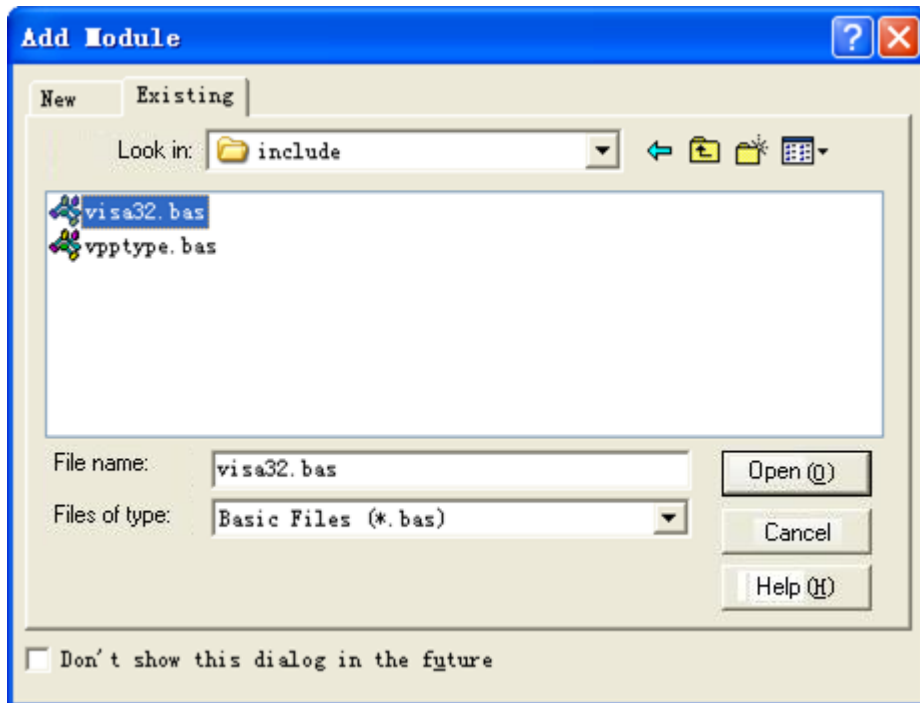
```

6. Save, compile 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.

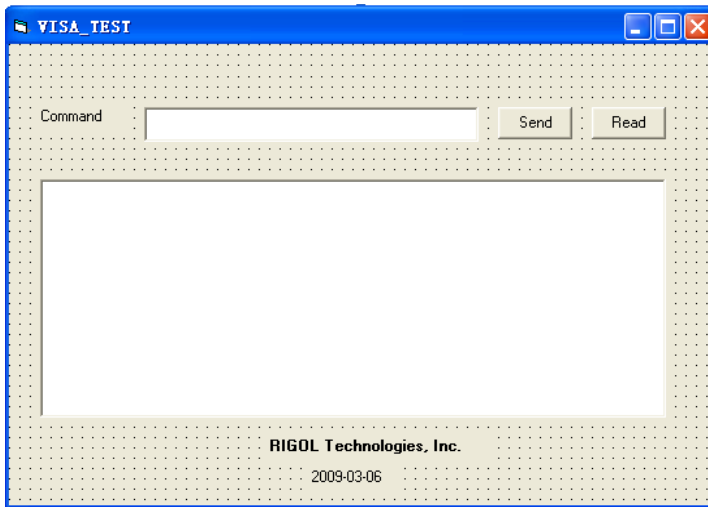
## Programming in Visual Basic 6.0

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 it;



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
    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 2-14

## I

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

## M

MEASure:CURRent[:DC]? 2-6  
 MEASure:POWEr[:DC]? 2-6  
 MEASure[:VOLTage][:DC]? 2-6

## O

OUTPut:OCP 2-9  
 OUTPut:OCP:STATe 2-8  
 OUTPut:OCP:STATe? 2-9  
 OUTPut:OCP? 2-9  
 OUTPut:OVP 2-8  
 OUTPut:OVP:STATe 2-8  
 OUTPut:OVP:STATe? 2-8  
 OUTPut:OVP? 2-8

OUTPut:TIMEr 2-10  
 OUTPut:TIMEr:STATe 2-10  
 OUTPut:TIMEr:STATe? 2-11  
 OUTPut:TIMEr? 2-10  
 OUTPut:TRACK 2-9  
 OUTPut:TRACK? 2-9  
 OUTPut:WAVE 2-10  
 OUTPut:WAVE? 2-10  
 OUTPut[:STATe] 2-7  
 OUTPut[:STATe]? 2-7

## R

RECALL:EXTErnal 2-23  
 RECALL:LOCal 2-23

## S

[SOURce:]CURRent[:LEVel][:IMMEDIATE][:AMP  
 Litude] 2-12  
 [SOURce:]CURRent[:LEVel][:IMMEDIATE][:AMP  
 Litude]? 2-12  
 [SOURce:]VOLTage[:LEVel][:IMMEDIATE][:AMP  
 Litude] 2-13  
 [SOURce:]VOLTage[:LEVel][:IMMEDIATE][:AMP  
 Litude]? 2-13  
 STORE:EXTErnal 2-23  
 STORE:LOCal 2-23  
 SYSTem:BEEPer[:IMMEDIATE] 2-16  
 SYSTem:BEEPer[:IMMEDIATE]? 2-16  
 SYSTem:BRIGht 2-16  
 SYSTem:BRIGht? 2-16  
 SYSTem:COMMunicate:GPIB:ADDReSS 2-20  
 SYSTem:COMMunicate:GPIB:ADDReSS? 2-20  
 SYSTem:COMMunicate:LAN:APPLY 2-20  
 SYSTem:COMMunicate:LAN:AUTOip[:STATe]

---

|  |                                    |
|--|------------------------------------|
| 2-18                                     | SYSTem:COMMunicate:LAN:SMASK? 2-19 |
| SYSTem:COMMunicate:LAN:AUTOip[:STATe]?   | SYSTem:LANGuage:TYPE 2-16          |
| 2-18                                     | SYSTem:LANGuage:TYPE? 2-16         |
| SYSTem:COMMunicate:LAN:DHCP[:STATe]      | SYSTem:LOCal 2-17                  |
| 2-17                                     | SYSTem:OTP 2-21                    |
| SYSTem:COMMunicate:LAN:DHCP[:STATe]?     | SYSTem:OTP? 2-21                   |
| 2-18                                     | SYSTem:POWEron 2-17                |
| SYSTem:COMMunicate:LAN:DNS 2-20          | SYSTem:POWEron? 2-17               |
| SYSTem:COMMunicate:LAN:DNS? 2-20         | SYSTem:REMote 2-17                 |
| SYSTem:COMMunicate:LAN:GATEway 2-19      | SYSTem:SELF:TEST:AD? 2-20          |
| SYSTem:COMMunicate:LAN:GATEway? 2-19     | SYSTem:SELF:TEST:DA? 2-21          |
| SYSTem:COMMunicate:LAN:IPADdress 2-18    | SYSTem:SELF:TEST:FAN? 2-21         |
| SYSTem:COMMunicate:LAN:IPADdress? 2-19   | SYSTem:SELF:TEST:FPGA? 2-21        |
| SYSTem:COMMunicate:LAN:MANualip[:STATe]  | SYSTem:SELF:TEST:USB? 2-21         |
| 2-18                                     | SYSTem:SELF:TEST:LAN? 2-21         |
| SYSTem:COMMunicate:LAN:MANualip[:STATe]? |                                    |
| 2-18                                     |                                    |
| SYSTem:COMMunicate:LAN:SMASK 2-19        |                                    |