

RIGOL

Programming Guide

DL3000 Series Programmable DC Electronic Load

Aug. 2017
RIGOL TECHNOLOGIES, INC.

Guaranty and Declaration

Copyright

© 2017 **RIGOL** TECHNOLOGIES, INC. All Rights Reserved.

Trademark Information

RIGOL is a registered trademark of **RIGOL** TECHNOLOGIES, INC.

Publication Number

PGJ01102-1110

Software Version

00.01.01

Software upgrade might change or add product features. Please acquire the latest version of the manual from **RIGOL** website or contact **RIGOL** to upgrade the software.

Notices

- **RIGOL** products are covered by P.R.C. and foreign patents, issued and pending.
- **RIGOL** reserves the right to modify or change parts of or all the specifications and pricing policies at the company's sole decision.
- Information in this publication replaces all previously released materials.
- Information in this publication is subject to change without notice.
- **RIGOL** shall not be liable for either incidental or consequential losses 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, photocopied, or rearranged without prior written approval of **RIGOL**.

Product Certification

RIGOL guarantees that this product conforms to the national and industrial standards in China as well as the ISO9001:2008 standard and the ISO14001:2004 standard. Other international standard conformance certifications are in progress.

Contact Us

If you have any problem or requirement when using our products or this manual, please contact **RIGOL**.

E-mail: service@rigol.com

Website: www.rigol.com

Document Overview

This manual is your guide to programming the **RIGOL** DL3000 series programmable DC power supply.

Main Topics in this Manual:

Chapter 1 Programming Overview

This chapter introduces how to set up remote communication between the electronic load and the PC, the remote control methods, the syntax, symbols, parameters, and abbreviation rules of the SCPI commands.

Chapter 2 Command System

This chapter introduces the syntax, function, parameters, and usage of each command.

Chapter 3 Application Instances

This chapter provides some application instances of the basic functions of the electronic load. Each application instance is composed of a series of commands, which are used to realize the specified basic function.

Chapter 4 Programming Instances

This chapter illustrates how to control the DL3000 series by programming in Excel, MATLAB, LabVIEW, Visual C++, and Visual C#.

Chapter 5 Appendix

This chapter provides default settings and warranty information for the DL3000 series.

Tip

For the latest version of this manual, download it from the official website of **RIGOL** (www.rigol.com).

Format Conventions in this Manual:

1. Key

The key at the front panel is denoted by the format of "Key Name (Bold) + Text Box". For example, **Utility** denotes the "Utility" key.

2. Menu Softkey

The menu softkey is denoted by the format of "Menu Word (Bold) + Character Shading". For example, **Interface** denotes the "Interface" menu softkey under **Utility**.

Content Conventions in this Manual:

DL3000 series programmable DC electronic load includes the following models. Unless otherwise specified, this manual takes DL3031A as an example to make a detailed introduction about the commands of the DL3000 series, and lists some application instances and programming examples of DL3031A.

Model	No. of Channels	Voltage	Current	Power
DL3021/DL3021A	1	DC 150 V	40 A	200 W
DL3031/DL3031A	1	DC 150 V	60 A	350 W

Contents

Guaranty and Declaration	I
Document Overview	II
Chapter 1 Programming Overview.....	1-1
To Build Remote Communication	1-2
Remote Control Method	1-3
SCPI Command Overview.....	1-4
Syntax.....	1-4
Symbol Description	1-4
Parameter Type.....	1-5
Command Abbreviation	1-5
SCPI Status Register	1-6
Questionable Status Register	1-7
Standard Event Status Register	1-8
Status Byte Register	1-8
Chapter 2 Command System.....	2-1
IEEE 488.2 Common Commands.....	2-2
*CLS	2-2
*ESE	2-3
*ESR?	2-4
*IDN?	2-4
*OPC.....	2-5
*OPT?	2-5
*PSC	2-6
*RST	2-6
*SRE	2-7
*STB?	2-7
*TRG	2-8
*TST?	2-8
*WAI.....	2-8
:STATus Commands	2-9
:STATus:QUEStionable:CONDition?	2-9
:STATus:QUEStionable:ENABle	2-10
:STATus:QUEStionable[:EVENT]?	2-10
:STATus:PRESet	2-11
:STATus:OPERation:CONDition?.....	2-11
:STATus:OPERation:ENABle.....	2-12
:STATus:OPERation[:EVENT]?	2-12
:MEASure commands and :FETCh commands	2-13
:FETCh:VOLTagE[:DC]?.....	2-13
:MEASure:VOLTagE[:DC]?	2-13
:FETCh:VOLTagE:MAX?	2-14
:MEASure:VOLTagE:MAX?	2-14
:FETCh:VOLTagE:MIN?	2-14
:MEASure:VOLTagE:MIN?.....	2-14
:FETCh:CURREnt[:DC]?	2-14
:MEASure:CURREnt[:DC]?.....	2-14
:FETCh:CURREnt:MAX?.....	2-14
:MEASure:CURREnt:MAX?	2-14
:FETCh:CURREnt:MIN?	2-14
:MEASure:CURREnt:MIN?	2-14
:FETCh:RESistance[:DC]?	2-15
:MEASure:RESistance[:DC]?	2-15
:FETCh:POWer[:DC]?	2-15

:MEASure:POWer[:DC]?	2-15
:FETCh:CAPability?	2-15
:MEASure:CAPability?	2-15
:FETCh:WATThours?	2-15
:MEASure:WATThours?	2-15
:FETCh:DISChargingTime?	2-16
:MEASure:DISChargingTime?	2-16
:FETCh:TIME?	2-16
:MEASure:TIME?	2-16
:FETCh:WAVedata?	2-16
:MEASure:WAVedata?	2-16
:TRIGger Commands	2-17
:TRIGger	2-17
:TRIGger:SOURce	2-17
:[SOURce] Commands	2-18
:[SOURce]:INPut	2-19
:[SOURce]:FUNction	2-19
:[SOURce]:FUNction:MODE	2-20
:[SOURce]:TRANsient	2-20
:[SOURce]:CURRent[:LEVel][:IMMediate]	2-21
:[SOURce]:CURRent:RANGe	2-21
:[SOURce]:CURRent:SLEW	2-22
:[SOURce]:CURRent:SLEW:POSitive	2-22
:[SOURce]:CURRent:SLEW:NEGative	2-23
:[SOURce]:CURRent:VON	2-23
:[SOURce]:CURRent:VLIMt	2-24
:[SOURce]:CURRent:ILIMt	2-24
:[SOURce]:CURRent:TRANsient:MODE	2-25
:[SOURce]:CURRent:TRANsient:ALEVel	2-25
:[SOURce]:CURRent:TRANsient:BLEVel	2-26
:[SOURce]:CURRent:TRANsient:AWIDth	2-26
:[SOURce]:CURRent:TRANsient:BWIDth	2-27
:[SOURce]:CURRent:TRANsient:FREQuency	2-27
:[SOURce]:CURRent:TRANsient:PERiod	2-28
:[SOURce]:CURRent:TRANsient:ADUTy	2-28
:[SOURce]:VOLTagE[:LEVel][:IMMediate]	2-29
:[SOURce]:VOLTagE:RANGe	2-29
:[SOURce]:VOLTagE:VLIMt	2-30
:[SOURce]:VOLTagE:ILIMt	2-30
:[SOURce]:RESistance[:LEVel][:IMMediate]	2-31
:[SOURce]:RESistance:RANGe	2-31
:[SOURce]:RESistance:VLIMt	2-32
:[SOURce]:RESistance:ILIMt	2-32
:[SOURce]:POWer[:LEVel][:IMMediate]	2-33
:[SOURce]:POWer:VLIMt	2-33
:[SOURce]:POWer:ILIMt	2-34
:[SOURce]:LIST:MODE	2-34
:[SOURce]:LIST:RANGe	2-35
:[SOURce]:LIST:COUNT	2-35
:[SOURce]:LIST:STEP	2-36
:[SOURce]:LIST:LEVel	2-36
:[SOURce]:LIST:WIDth	2-37
:[SOURce]:LIST:SLEW	2-37
:[SOURce]:LIST:END	2-38
:[SOURce]:WAVE:TIME	2-39
:[SOURce]:WAVE:TSTep	2-39
:SYSTem Commands	2-40
:SYSTem:KEY	2-40

:SYSTem:ERRor?	2-41
:SYSTem:VERSion?	2-41
:SYSTem:IDN	2-41
:LXI Commands	2-42
LXI:IDENtify[:STATE]	2-42
LXI:MDNS[:ENABLE]	2-42
LXI:RESet	2-43
LXI:REStart	2-43
:LIC Command	2-44
:LIC:SET	2-44
Chapter 3 Application Instances	3-1
LIST Function	3-1
Chapter 4 Programming Instances	4-1
Programming Preparations	4-1
Excel Programming Instance	4-3
MATLAB Programming Instance	4-6
LabVIEW Programming Instance	4-7
Visual C++ Programming Instance	4-12
Visual C# Programming Instance	4-16
Chapter 5 Appendix	5-1
Appendix A: Default Settings	5-1
Appendix B: Warranty	5-4

Chapter 1 Programming Overview

This chapter introduces how to set up remote communication between the electronic load and the PC, the remote control methods, the syntax, symbols, parameters, and abbreviation rules of the SCPI commands.

Contents in this chapter:

- ◆ [To Build Remote Communication](#)
- ◆ [Remote Control Method](#)
- ◆ [SCPI Command Overview](#)
- ◆ [SCPI Status Register](#)

To Build Remote Communication

DL3000 series electronic load can communicate with the PC via USB, LAN (option), RS232, or GPIB (extended using the USB-GPIB module) interface.

Note: The RS232 protocol command ends with "\r\n" for the DL3000 series.

Operation Procedures:

- 1. Install Ultra Sigma (PC) software.**
You can download Ultra Sigma from the official website of **RIGOL** (www.rigol.com) and install the software according to the installation wizard.
- 2. Connect the electronic load to the PC and configure the interface parameters for the load.**
Connect the load to the PC by using any of the four communication interfaces: USB, LAN, RS232, and GPIB (via the USB-GPIB interface converter), as shown in Figure 1-1.

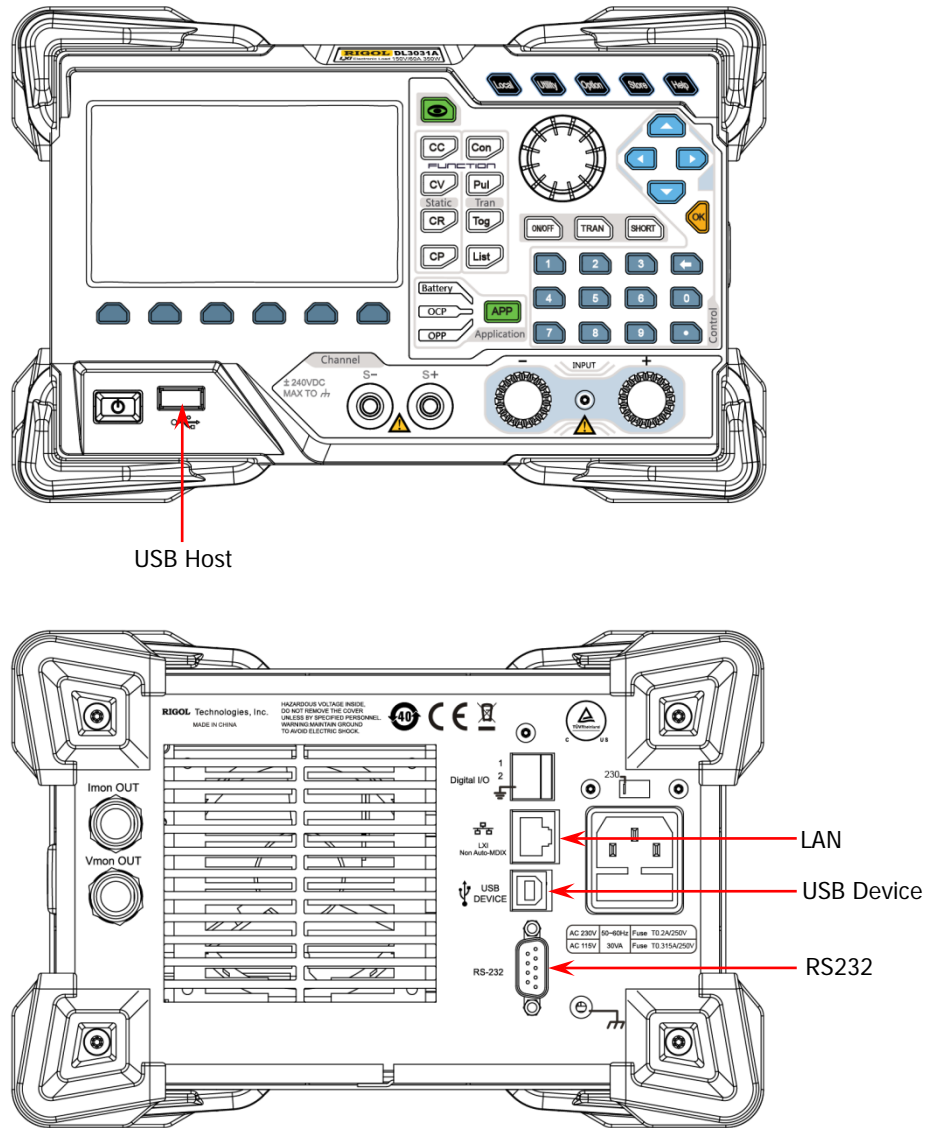


Figure 1-1 DL3000 Communication Interface

Note: For DL3021A and DL3031A, they have been configured with LAN network before leaving the factory, so they can remotely communicate with the PC via the LAN interface. For DL3021 and DL3031, LAN interface communication is an optional function. If you need the function, please purchase the option and install it properly ([:LIC:SET](#)).

- (1) Remote communication via USB: Use the USB cable to connect the USB DEVICE interface on the rear panel of DL3000 to the USB HOST interface of the PC.
- (2) Remote communication via LAN:
 - Make sure that your PC has been accessed to the local area network.
 - Check whether the local area network where your PC resides supports DHCP or auto IP mode. If not, you need to obtain the available network interface parameters, including the IP address, subnet mask, default gateway, and DNS.
 - Use the network cable to have DL3000 get access to the local area network.
 - Press **Utility** → **Interface** → **LAN** to configure the IP address, subnet mask, default gateway, and DNS for the instrument.
- (3) Remote communication via RS232:
 - Use the RS232 cable to connect the RS232 interface to the PC or the data terminal equipment (DTE).
 - Press **Utility** → **Interface** → **RS232** to set the interface parameters (baud rate, parity, etc.) that match the PC or DTE.
- (4) Remote communication via GPIB:
 - Use the USB-GPIB interface module to expand the GPIB interface by connecting the module to the USB HOST interface on the front panel of DL3000.
 - Use the GPIB cable to connect the instrument to your PC.
 - Press **Utility** → **Interface** → **GPIB** to configure the GPIB address for the instrument.

3. Check whether the connection is successful.

Run Ultra Sigma, and then search for the resources and right-click the resource name. Select "SCPI Panel Control" to open the SCPI command control panel. In the SCPI command control panel, input a correct command. After that, click **Send Command** first and then click **Read Response**. Or you can also directly click **Send & Read** to verify whether the connection works properly.

Remote Control Method

1. User-defined programming

You can refer to "**Command System**" of Chapter 2 in this manual to use the SCPI (Standard Commands for Programmable Instruments) commands to control DL3000 by programming in LabVIEW, Visual C#, and other development environments. For details, refer to "**Programming Instances**" of Chapter 4 in this manual.

2. Send SCPI commands via the PC software

You can use the PC software to send commands to control DL3000 remotely. **RIGOL** Ultra Sigma is recommended. You can download the software from **RIGOL** official website (www.rigol.com).

SCPI Command Overview

SCPI (Standard Commands for Programmable Instruments) is a standardized instrument programming language that is built upon the existing standard IEEE 488.1 and IEEE 488.2 and conforms to various standards, such as the floating point operation rule in IEEE 754 standard, ISO 646 7-bit coded character set for information interchange (equivalent to ASCII programming). This chapter introduces the syntax, symbols, parameters, and abbreviation rules of the SCPI commands.

Syntax

The SCPI commands provide a hierarchical tree structure, and consist of multiple subsystems. Each command subsystem consists of one root keyword and one or more sub-keywords. The command line usually starts with a colon; the keywords are separated by colons, and following the keywords are the parameter settings available. The keywords of the command and the first parameter is separated by a space. The command ending with a quotation mark indicates querying a certain function.

For example,

```
:STATus:QUESTionable:ENABle <enable value>
:STATus:QUESTionable:ENABle?
```

STATus is the root keyword of the command. QUESTionable and ENABle are the second and third keyword, respectively. The command line starts with a colon, and different levels of keywords are also separated by colons. <enable value> indicates a settable parameter. The command keywords ":STATus:QUESTionable:ENABle" and the parameter <enable value> are separated by a space. The quotation mark (?) indicates querying.

In some commands with multiple parameters, commas are often used to separate these parameters. For example,

```
:SYSTem:IDN:SET <manufacturer>,<model>,<sn>,<firmware>
```

Symbol Description

The following four symbols are not part of the command, and they are not sent with the commands, but taken as delimiters to better describe the parameters in the command.

1. Braces { }

The contents enclosed in the braces can contain multiple optional parameters. When sending the command, you must select one of the parameters.

2. Vertical Bar |

The vertical bar is used to separate multiple parameter options. When sending the command, you must select one of the parameters.

3. Square Brackets []

The contents (keywords or parameters) in the square brackets can be omitted. If the parameter is omitted, it will be set to the default. For example, when sending the :MEASure[:VOLTage][:DC]? command, you can select any one of the following four commands, as they can achieve the same effects as the :MEASure[:VOLTage][:DC]? command.

```
:MEASure?
:MEASure:DC?
:MEASure:VOLTage?
:MEASure:VOLTage:DC?
```

4. Angle Brackets < >

The parameter enclosed in the angle brackets must be replaced by an effective value. For example, the :LIC:SET <sn> command must be sent in the form of :LIC:SET UVF2L3N3XXKYTB73PPRSA4XDMSRT.

Parameter Type

The parameters contained in this manual can be divided into the following five types: Bool, Integer, Real, Discrete, and ASCII String.

1. Bool

The parameter can be set to ON (1) or OFF (0).

2. Integer

Unless otherwise specified, the parameter can be any integer within the effective value range.

Note: Do not set the parameter to a decimal, otherwise, errors will occur.

3. Real

Unless otherwise specified, the parameter can be any real-value (in decimal form or in scientific notation) within the effective value range.

4. Discrete

The parameters can only be the specified numerical values or characters.

5. ASCII String

The parameter can be the combinations of ASCII characters.

Besides, many commands support the MINimum, MAXimum, or DEF parameter. MINimum indicates setting the parameter to a minimum value or querying the minimum value of the parameter; MAXimum indicates setting the parameter to a maximum value or querying the maximum value of the parameter; DEF indicates setting the parameter to a default value.

Command Abbreviation

The letters in the commands are case-insensitive. The commands can be input in uppercase letters or in lowercase letters. If abbreviation is used, you must enter all the uppercase letters that exist in the command syntax. For example, :SYSTem:ERRor? can be abbreviated as :SYST:ERR?

SCPI Status Register

All the SCPI commands to the instrument perform the status register operation in the same way. The status system records the instrument status into the following three registers: questionable status register, standard event status register, and status byte register. The status byte register is used to record advanced summary information, which is reported by other registers. The SCPI status system of the DL3000 is shown in Figure 1-2.

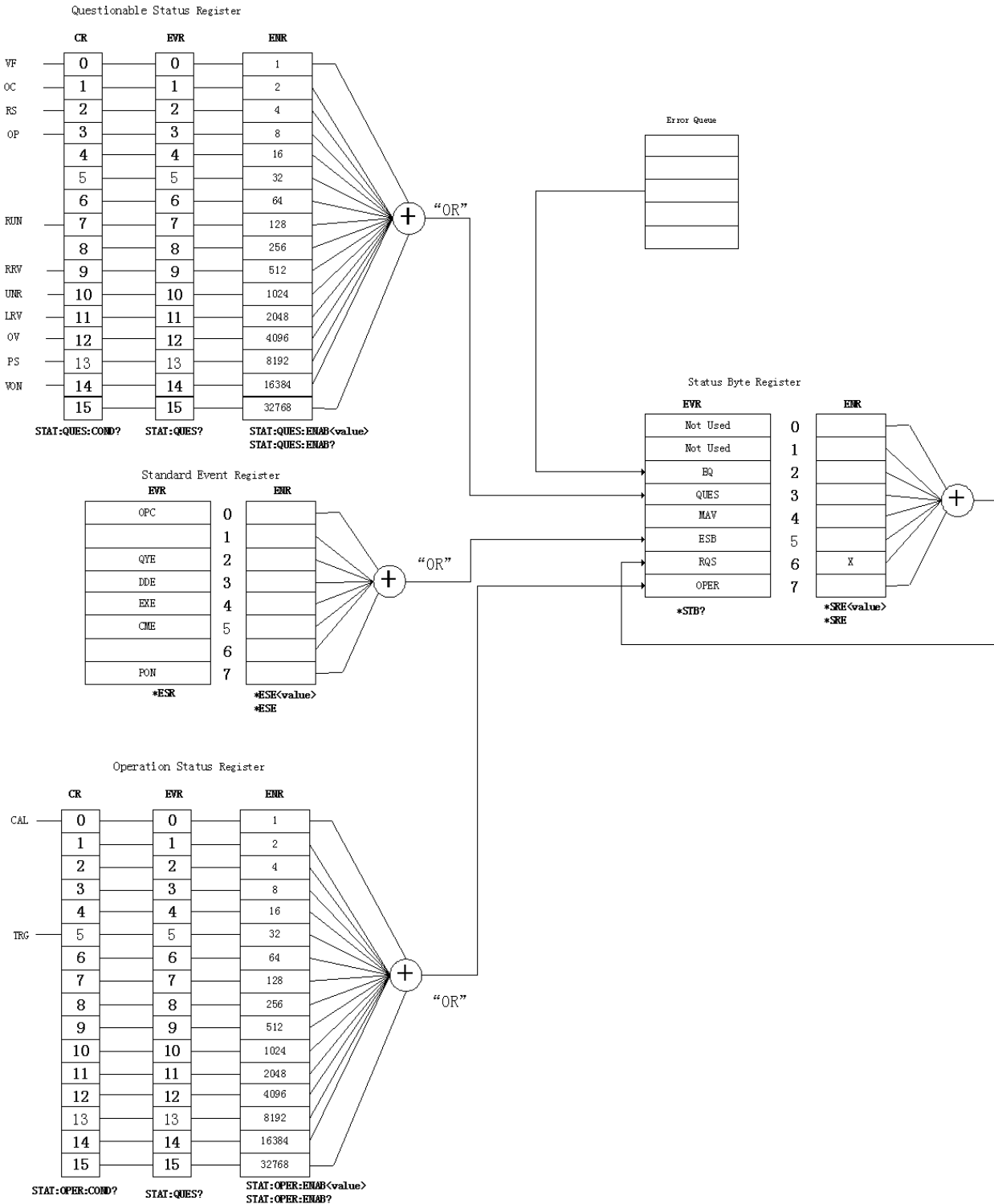


Figure 1-2 SCPI Status System of the DL3000 Series

Questionable Status Register

The SCPI status system of the DL3000 is shown in Figure 1-2. The questionable status register monitors the overall instrument condition, such as voltage/current control, OTP, OVP, OCP, and fan failure. You can send the `:STATus:QUEStionable[:EVENT]?` command to read the register. The definitions for the bits in the questionable status register and the decimal value that corresponds to the binary-weighted value are shown in Table 1-1.

Table 1-1 Definitions for the bits in the questionable status register and the decimal value that corresponds to the binary-weighted value

Bit	Weight	Name	Description
0	1	VF	Voltage fault. Overvoltage and reverse voltage occurred; the OV or the RV condition is removed.
1	2	OC	Overcurrent occurred.
2	4	RS	Remote Sense terminal connection
3	8	OP	Overpower occurred.
4	Not used	Not used	This bit is always set to 0.
5	Not used	Not used	This bit is always set to 0.
6	Not used	Not used	This bit is always set to 0.
7	128	RUN	Runs in List mode
8	Not used	Not used	This bit is always set to 0.
9	512	RRV	Remote Reverse Voltage. When reverse voltage occurs to the remote terminals, the bit and VF bit are set. When the reverse voltage is removed, the RRV bit is cleared, but the VF bit will not be cleared.
10	1024	UNR	Unregulated. The input is unregulated. When the input has been regulated, the bit is cleared.
11	2048	LRV	Local Reverse Voltage. When reverse voltage occurs to the input terminals, the bit and VF bit are set. When the reverse voltage is removed, the LRV bit is cleared, but the VF bit will not be cleared.
12	4096	OV	Overvoltage. When OV occurs, the OV bit and VF bit are set, and the load is turned off.
13	8192	PS	Protection Shutdown. When overcurrent, overpower, or overtemperature occurred, the load's input is turned off (protection shutdown).
14	16384	VON	Voltage of sink current on. When the input voltage exceeds the set Von value, the load starts to sink the current.
15	Not used	Not used	This bit is always set to 0.

● Event Register

The event register latches events from the condition register. Once the bit in the event register is set to 1, the subsequent event changes are ignored. The event register is read-only. Once the bit is set to 1, it will remain set and will not be cleared until cleared by running a query command (e.g. the `*ESR?` command) or the `*CLS` command. When you send a query command to query the event register, a decimal value (equivalent to the binary-weighted sum of all bits enabled in the register) will be returned.

● Enable Register

The enable register is both readable and writable. It defines whether to report the events in the event register to the status byte register. The `:STATus:PRESet` command will clear all bits in the enable register, and the `*PSC 1` command will make all bits in the enabled register cleared once the instrument is powered on. To enable the events in the event register to be reported to the status byte register, write a decimal value (equivalent to the binary-weighted sum of all bits enabled in the register) into the enable register.

Standard Event Status Register

The standard event status register records the following events that might occur on your instrument: power-on inspection, command syntax error, command execution error, self-test/calibration error, query error, or operation completed. All these events or any one of the events can be reported to Bit 5 (ESB, Event Summary Bit) of status byte register via the enable register. To set the enable register mask, you need to use the [*ESE](#) command to write a decimal value into the register. The definitions for the bits in the standard event status register and the decimal value that corresponds to the binary-weighted value are shown in Table 1-2.

Table 1-2 Definitions for the bits in the standard event status register and the decimal value that corresponds to the binary-weighted value

Bit	Weight	Name	Description
0	1	OPC (Operation complete)	The operation has been completed. All pending operations were completed following the execution of the *OPC command.
1	Not used	Not used	This bit is always set to 0.
2	4	QYE (Query error)	A query error has occurred.
3	8	DDE (Device error)	A device error (error caused by self-test or calibration) has occurred.
4	16	EXE (Execution error)	An execution error (trigger ignored, initialization ignored, setting conflict, data over-the-limit, data length too long, or invalid parameter value) has occurred.
5	32	CME (Command error)	A command error (command syntax error) has occurred.
6	Not used	Not used	This bit is always set to 0.
7	128	PON (Power on)	Power-on test. After the bit has been read from the last event of the register or has been cleared, turn off the instrument and then turn it on again.

Status Byte Register

The status byte register reports the status information about other status registers; queries the report of Bit 4 (MAV, Message Available) data to be queried in the output buffer. The summary bit in the status byte register will not be latched. Clearing the event register will clear the corresponding summary bits of the status byte register. Read all the information in the output cache (including any pending queries), and will clear Bit 4 (MAV, Message Available). The definitions for the bits in the status byte register and the decimal value that corresponds to the binary-weighted value are shown in Table 1-3.

Table 1-3 Definitions for the bits in the status byte register and the decimal value that corresponds to the binary-weighted value

Bit	Weight	Name	Description
0	Not used	Not used	This bit is always set to 0.
1	Not used	Not used	This bit is always set to 0.
2	4	EQ	Error queue. One or more errors were found in the error queue.
3	8	Questionable status summary	One or multiple bits of the questionable status register has/have been set (To enable the bit, refer to the :STATus:QUESTIONable:ENABLE command).
4	16	Message available	A message is available in the output buffer.
5	32	Standard event status summary	One or multiple bits of the standard event status register has/have been set (To enable the bit, refer to the *ESE command).
6	64	Master summary	One or multiple bits of the status byte register has/have been set.
7	128	Operation status summary	One or multiple bits of the operation status register has/have been set (To enable the bit, refer to the :STATus:OPERation:ENABLE command).

Chapter 2 Command System

This chapter introduces the syntax, function, parameters, and usage of each command.

Contents in this chapter:

- ◆ [IEEE 488.2 Common Commands](#)
- ◆ [:STATus Commands](#)
- ◆ [:MEASure commands and :FETCh commands](#)
- ◆ [:TRIGger Commands](#)
- ◆ [:\[SOURce\] Commands](#)
- ◆ [:SYSTem Commands](#)
- ◆ [:LXI Commands](#)
- ◆ [:LIC Command](#)

IEEE 488.2 Common Commands

Command List:

- ◆ [*CLS](#)
- ◆ [*ESE](#)
- ◆ [*ESR?](#)
- ◆ [*IDN?](#)
- ◆ [*OPC](#)
- ◆ [*OPT?](#)
- ◆ [*PSC](#)
- ◆ [*RST](#)
- ◆ [*SRE](#)
- ◆ [*STB?](#)
- ◆ [*TRG](#)
- ◆ [*TST?](#)
- ◆ [*WAI](#)

*CLS

Syntax *CLS

Description Clears all the event registers.

- Remarks**
- You can send the command for querying the event register ([:STATus:QUESTionable\[:EVENT\]?](#) or [*ESR?](#)) to clear the corresponding event register.
 - Sending the reset command ([*RST](#)) or device clearing command cannot clear the event register.

Related Command [:STATus:QUESTionable\[:EVENT\]?](#)
[*ESR?](#)
[*RST](#)

***ESE**

Syntax *ESE <enable value>

*ESE?

Description Enables the bit in the enable register part of the standard event status register.

Queries the enabled bit in the enable register part of the standard event status register.

Parameter

Name	Type	Range	Default
<enable value>	String	Refer to "Remarks".	None

Remarks

- The parameter <enable value> is a decimal value. It corresponds to the binary-weighted sum of the bit enabled in the enable register part of the standard event status register. For the definitions for the bits in the standard event status register and the decimal value that corresponds to the binary-weighted value, refer to Table 1-2. For example, to enable Bit 2 (query error) and Bit 4 (execution error) in the enable register part of the standard event status register, set the parameter <enable value> to 20 (based on the formula: $2^2 + 2^4 = 20$).
- After the bit in the enable register part of the standard event status register is enabled, the system will report the status of the bit to the status byte register.
- When the parameter <enable value> is set to 0, you can run the command to clear the enable register part of the standard event status register.
- You can also send the [*PSC](#) command (*PSC 1) to clear the enable register part of the standard event status register once the instrument is turned on next time.

Return Format

Returns a decimal value. It corresponds to the binary-weighted sum of the bit enabled in the enable register part of the standard event status register. For example, 20.

Example

*ESE 20 /*Enables Bit 2 (query error) and Bit 4 (execution error) in the enable register part of the standard event status register*/

*ESE? /*Queries the bit enabled in the enable register part of the current standard event status register, returns 20*/

Related Command

[*PSC](#)

***ESR?**

Syntax *ESR?

Description Queries the event status register of the standard event status register.

- Remarks**
- The query returns a decimal value. It corresponds to the binary-weighted sum of all the bits in the enable register part of the standard event status register. The command also clears the status of the register. For the definitions for the bits in the standard event status register and the decimal value that corresponds to the binary-weighted value, refer to Table 1-2.

For example, if query error and execution error have occurred on the current instrument, Bit 2 (query error) and Bit 4 (execution error) in the event register part of the standard event status register will be set to 1. At this time, when you run the command, it returns 20 (based on the formula: $2^2 + 2^4 = 20$).

- The bit in the event register part of the standard event status register is latched, and reading the event register part will clear it. You can also send the [*CLS](#) command to clear the register.

Return Format Returns a decimal value. It corresponds to the binary-weighted sum of all the bits in the event register part of the standard event status register. For example, 20.

Example *ESR? /*Queries the event register part of the standard event status register. The query returns 20*/

Related Command [*CLS](#)

***IDN?**

Syntax *IDN?

Description Queries the instrument information.

Return Format Returns the instrument information in strings, which includes four parts: manufacturer, model number, serial number, and the version number of the digital board. The four parts are separated by commas.

OPC*Syntax** *OPC

*OPC?

Description After you run the command, Bit 0 (operation complete bit) in the event register part of the standard event status register is set to 1.

Queries whether the *OPC command has been executed. If yes, it returns "1" in the output buffer.

- Remarks**
- "Operation complete" indicates that all pending operations were completed following the execution of the *OPC command.
 - Sending the query command *OPC? and reading the results can ensure that all the commands can be executed one by one.
 - When configuring the instrument by programming commands (executing a series of command), running the command as the last one can determine when the command queue can all be executed. After all the commands in a queue have been executed, Bit 0 (OPC, operation complete bit) in the event register part of the standard event status register will be set to 1.
 - If you first run the query command (querying data of the output buffer) and then send the *OPC command, you can determine when the data is available through the "OPC" bit.

Return Format Returns 1.**Example** *OPC /*After you run the command, Bit 0 (operation complete bit) in the event register part of the standard event status register is set to 1.*/

*OPC? /*Queries whether the current operation is finished. The command returns 1.*/

OPT?*Syntax** *OPT?**Description** Queries the installation status of the option.

- Remarks**
- The options include high slew rate, high frequency, high readback resolution, LAN, and Digital I/O.
 - For DL3021A and DL3031A, they have been installed with the above five options when leaving the factory. For DL3021 and DL3031, if you need any of the options, please purchase the desired option and install it properly ([:LIC:SET](#)).

Return Format Returns the installation status of the option. Different options are separated by commas. If the option has been installed, it returns the name of the option; if not, it returns 0.

PSC*Syntax** *PSC {0|1}

*PSC?

Description Enables (1) or disables (0) clearing of the enable register part of the status byte register and the standard event status register at power-on.

Queries the state of enable register part when clearing of the enable register part of the status byte register and the standard event status register at power-on.

- Remarks**
- The *PSC 1 command indicates enabling power-on clearing of affected registers; the *PSC 0 command indicates disabling power-on clearing of affected registers.
 - You can also send the [*SRE](#) command (*SRE 0) or the [*ESE](#) command (*ESE 0) to disable clearing of the enable register part of the status byte register and the standard event status register at power-on.

Return Format 0 or 1**Example** *PSC 1 /*Enables clearing of the enable register part of the status byte register and the standard event status register at power-on.*/

*PSC? /*Queries whether to enable or disable the power-on clearing of the affected registers. It returns 1.*/

Related Command [*SRE](#)
[*ESE](#)***RST****Syntax** *RST**Description** Restores the instrument to factory default settings (refer to "**Appendix A: Default Settings**") and clears the error queue.

***SRE**

Syntax *SRE <enable value>

*SRE?

Description Enables bits in the enable register part of the status byte register.

Queries the bit enabled in the enable register part of the status byte register.

Remarks

- The parameter <enable value> is a decimal value. It corresponds to the binary-weighted sum of the bit enabled in the enable register part of the status byte register. For the definitions for the bits in the status byte register and the decimal value that corresponds to the value, refer to Table 1-3.

For example, to enable Bit 3 (QUES, questionable data summary bit) and Bit 4 (MAV, message available) in the enable register part of the standard event status register, set the parameter <enable value> to 24 (based on the formula: $2^3 + 2^4 = 24$).

- After the bits are enabled, the system sends the service request through Bit 6 (RQS, request service bit) of the status byte register.
- When the parameter <enable value> is set to 0, you can run the command to clear the enable register part of the standard event status register. You can also send the [*PSC](#) command (*PSC 1) to clear the enable register part of the status byte register once the instrument is turned on next time.

Return Format The query returns a decimal value. It corresponds to the binary-weighted sum of the bit enabled in the enable register part of the status byte register. For example, 24.

Example *SRE 24 /*Enables Bit 3 (QUES) and Bit 4 (MAV) in the enable register part of the status byte register. Enables the service request.*/

*SRE? /*Queries the bit enabled in the enable register part of the status byte register. The query returns 24.*/

Related Command [*PSC](#)

***STB?**

Syntax *STB?

Description Queries the condition register of the current status byte register.

Remarks The query returns a decimal value. It corresponds to the binary-weighted sum of all the bits in the register, but the command does not clear the register. For the definitions for the bits in the status byte register and the decimal value that corresponds to the value, refer to Table 1-3.

For example, if the current instrument produces questionable results and requests service for disruption, then Bit3 (QUES) and Bit6 (RQS) in the condition register of the status byte register will be set to 1. Then, run the command and the query command returns 72. (It is calculated based on the formula: $2^3 + 2^6 = 72$).

Return Format The query returns a decimal value. It corresponds to the binary-weighted sum of all the bits in the condition register of the status byte register. For example, 72.

Example *STB? /*Queries the condition register of the current status byte register. The query returns 72.*/

***TRG**

Syntax *TRG

Description Generates a trigger action.

- Remarks**
- The command is only applicable to the trigger system with the bus (software) trigger source ([:STATUS:QUESTIONABLE:CONDITION?](#)) being selected.
 - After you select the bus (software) trigger source, sending the command will trigger the instrument.

Example :TRIG:SOUR BUS /*Selects the bus (software) trigger source.*/

Related Command [:TRIGger](#)
[:TRIGger:SOURce](#)

***TST?**

Syntax *TST?

Description Queries the self-test results of the instrument.

Remarks After the load is powered on, it starts self-test operation. The query command queries the self-test results (such as Vmon trig reference voltage, Imon trig reference voltage, and radiator temperature).

Return Format The query returns the self-test results such as Vmon trig reference voltage, Imon trig reference voltage, and radiator temperature. The return format is shown below:
OppRef: PASS,VmonTrig: PASS,ImonTrig: PASS,OcpRef: PASS,OvpRef: PASS,Temp1: PASS,Temp2: PASS

***WAI**

Syntax *WAI

Description Configures the instrument to wait for all pending operations to complete before executing any additional commands.

Remarks When you select "Bus" (bus (software) trigger) as the trigger source, sending this command can ensure that your application is synchronized with the instrument. After you run the command, the instrument will wait for all pending operations to complete before executing any additional commands.

Example *WAI /*Configures the instrument to wait for all pending operations to complete before executing any additional commands.*/

:STATus Commands

Command List:

- ◆ [:STATus:QUEStionable:CONDition?](#)
- ◆ [:STATus:QUEStionable:ENABle](#)
- ◆ [:STATus:QUEStionable\[:EVENT\]?](#)
- ◆ [:STATus:PRESet](#)
- ◆ [:STATus:OPERation:CONDition?](#)
- ◆ [:STATus:OPERation:ENABle](#)
- ◆ [:STATus:OPERation\[:EVENT\]?](#)

:STATus:QUEStionable:CONDition?

Syntax :STATus:QUEStionable:CONDition?

Description Queries the condition register of the questionable status register.

Remarks The query returns a decimal value. It corresponds to the binary-weighted sum of all the bits in the register. The definitions for the bits in the questionable status register and the decimal value that corresponds to the binary-weighted value are shown in Table 1-1.

Return Format The query returns a decimal value. It corresponds to the binary-weighted sum of all the bits in the register. For example, the query returns 1.

Example :STAT:QUES:COND? /*Queries the condition register of the questionable status register.
The query returns 1.*/*

:STATus:QUESTionable:ENABLE

Syntax :STATus:QUESTionable:ENABLE <enable value>

:STATus:QUESTionable:ENABLE?

Description Enables bits in the enable register part of the questionable status register.

Queries the bit enabled in the enable register part of the questionable status register.

Parameter

Name	Type	Range	Default
<enable value>	Discrete	Refer to "Remarks".	None

Remarks

- The parameter <enable value> is a decimal value. It corresponds to the binary-weighted sum of the bit enabled in the enable register part of the questionable status register. The definitions for the bits in the questionable status register and the decimal value that corresponds to the binary-weighted value are shown in Table 1-1.
- After the bit in the enable register part of the questionable status register is enabled, the system will report the status of the bit to the status byte register.
- When the parameter <enable value> is set to 0, you can run the command to clear the enable register part of the questionable status register.

Return Format

The query returns a decimal value. It corresponds to the binary-weighted sum of the bit enabled in the enable register part of the questionable status register. For example, 17.

Example

```
:STAT:QUES:ENAB 17 /*Enables Bit 0 and Bit 4 in the enable register part of the
questionable status register.*/

:STAT:QUES:ENAB? /*Queries the bit enabled in the enable register part of the
questionable status register. The query returns 17.*/
```

:STATus:QUESTionable[:EVENT]?

Syntax :STATus:QUESTionable[:EVENT]?

Description Queries the event register of the questionable status register.

Remarks

- The query returns a decimal value. It corresponds to the binary-weighted sum of all the bits in the enable register part of the standard event status register. The command also clears the status of the register. The definitions for the bits in the questionable status register and the decimal value that corresponds to the binary-weighted value are shown in Table 1-1.
- The bit in the event register part of the questionable status register is latched, and reading the event register part will clear the register. You can also send the [*CLS](#) command to clear the register.

Return Format

The query returns a decimal value. It corresponds to the binary-weighted sum of all the bits in the event register part of the questionable status register. For example, 17.

Example

```
:STAT:QUES? /*Queries the event register part of the questionable status register.
The query returns 17.*/
```

Related Command

[*CLS](#)

:STATus:PRESet

Syntax :STATus:PRESet

Description Clears all the bits in the event register part of the questionable status register.

Remarks The event registers that have been cleared include query event enable register, channel summary event enable register, and operation event enable register. Other registers will not be affected by the command.

Example :STAT:PRES /*Clears all the bits in the event register part of the questionable status register.*/*

:STATus:OPERation:CONDition?

Syntax :STATus:OPERation:CONDition?

Description Queries the operation condition register of the questionable status register.

Remarks

- The query returns a decimal value. It corresponds to the binary-weighted sum of all the bits in the register. The definitions for the bits in the questionable status register and the decimal value that corresponds to the binary-weighted value are shown in Table 1-1.
- It is read-only, and keeps the real-time (unlatched) operation state of the load.
- The command is not exclusive to the channel. It is applicable to the mainframe.

Return Format The query returns a decimal value. It corresponds to the binary-weighted sum of all the bits in the register. For example, the query returns 1.

Example :STAT:OPER:COND? /*Queries the operation condition register of the questionable status register. The query returns 1.*/*

:STATus:OPERation:ENABLE

Syntax :STATus:OPERation:ENABLE <enable value>

:STATus:OPERation:ENABLE?

Description Enables bits in the operation enable register of the questionable status register.

Queries the bit enabled in the operation enable register of the questionable status register.

Parameter

Name	Type	Range	Default
<enable value>	Discrete	0-65535	0

Remarks

- The parameter <enable value> is a decimal value. It corresponds to the binary-weighted sum of the bit to be enabled in the enable register part of the questionable status register. The definitions for the bits in the questionable status register and the decimal value that corresponds to the binary-weighted value are shown in Table 1-1.
- After the bit in the operation enable register of the questionable status register is enabled, the system will report the status of the bit to the status byte register.
- When the parameter <enable value> is set to 0, you can run the command to clear the operation enable register of the questionable status register.

Return Format The query returns a decimal value. It corresponds to the binary-weighted sum of the bit to be enabled in the operation enable register of the questionable status register. For example, 17.

Example :STAT:OPER:ENAB 17 /*Enables Bit 0 and Bit 4 in the operation enable register of the questionable status register.*/
 :STAT:OPER:ENAB? /*Queries the bit to be enabled in the operation enable register of the questionable status register. The query returns 17.*/*

:STATus:OPERation[:EVENT]?

Syntax :STATus:OPERation[:EVENT]?

Description Queries the operation event register of the questionable status register.

Remarks

- The query returns a decimal value. It corresponds to the binary-weighted sum of all the bits in the enable register part of the standard event status register. The command also clears the status of the register. The definitions for the bits in the questionable status register and the decimal value that corresponds to the binary-weighted value are shown in Table 1-1.
- The bit in the operation event register of the questionable status register is latched, and reading the operation event register will clear the register. You can also send the [*CLS](#) command to clear the register.

Return Format The query returns a decimal value. It corresponds to the binary-weighted sum of all the bits in the operation event register of the questionable status register. For example, 17.

Example :STAT:OPER? /*Queries the operation event register of the questionable status register. The query returns 17.*/*

Related Command [*CLS](#)

:MEASure commands and :FETCh commands

Command List:

- ◆ [:FETCh:VOLTag\[:DC\]?](#)
- ◆ [:MEASure:VOLTag\[:DC\]?](#)
- ◆ [:FETCh:VOLTag:MAX?](#)
- ◆ [:MEASure:VOLTag:MAX?](#)
- ◆ [:FETCh:VOLTag:MIN?](#)
- ◆ [:MEASure:VOLTag:MIN?](#)
- ◆ [:FETCh:CURRent\[:DC\]?](#)
- ◆ [:MEASure:CURRent\[:DC\]?](#)
- ◆ [:FETCh:CURRent:MAX?](#)
- ◆ [:MEASure:CURRent:MAX?](#)
- ◆ [:FETCh:CURRent:MIN?](#)
- ◆ [:MEASure:CURRent:MIN?](#)
- ◆ [:FETCh:RESistance\[:DC\]?](#)
- ◆ [:MEASure:RESistance\[:DC\]?](#)
- ◆ [:FETCh:POWer\[:DC\]?](#)
- ◆ [:MEASure:POWer\[:DC\]?](#)
- ◆ [:FETCh:CAPability?](#)
- ◆ [:MEASure:CAPability?](#)
- ◆ [:FETCh:WATThours?](#)
- ◆ [:MEASure:WATThours?](#)
- ◆ [:FETCh:DISChargingTime?](#)
- ◆ [:MEASure:DISChargingTime?](#)
- ◆ [:FETCh:TIME?](#)
- ◆ [:MEASure:TIME?](#)
- ◆ [:FETCh:WAVedata?](#)
- ◆ [:MEASure:WAVedata?](#)

:FETCh:VOLTag[:DC]?

:MEASure:VOLTag[:DC]?

Syntax :FETCh:VOLTag[:DC]?
:MEASure:VOLTag[:DC]?

Description Reads the input voltage of the instrument.

Return The query returns a real number.

Format

Related [:\[SOURce\]:VOLTag\[:LEVel\]\[:IMMediate\]](#)
Command

:FETCh:VOLTage:MAX?
:MEASure:VOLTage:MAX?

Syntax :FETCh:VOLTage:MAX?
:MEASure:VOLTage:MAX?

Description Reads the maximum input voltage of the instrument.

Return The query returns a real number.
Format

:FETCh:VOLTage:MIN?
:MEASure:VOLTage:MIN?

Syntax :FETCh:VOLTage:MIN?
:MEASure:VOLTage:MIN?

Description Reads the minimum input voltage of the instrument.

Return The query returns a real number.
Format

:FETCh:CURRent[:DC]?
:MEASure:CURRent[:DC]?

Syntax :FETCh:CURRent[:DC]?
:MEASure:CURRent[:DC]?

Description Reads the input current of the instrument.

Return The query returns a real number.
Format

Related [:\[SOURce\]:CURRent\[:LEVel\]\[:IMMediate\]](#)
Command

:FETCh:CURRent:MAX?
:MEASure:CURRent:MAX?

Syntax :FETCh:CURRent:MAX?
:MEASure:CURRent:MAX?

Description Reads the maximum input current of the instrument.

Return The query returns a real number.
Format

:FETCh:CURRent:MIN?
:MEASure:CURRent:MIN?

Syntax :FETCh:CURRent:MIN?
:MEASure:CURRent:MIN?

Description Reads the minimum input current of the instrument.

Return The query returns a real number.
Format

**:FETCh:RESistance[:DC]?
:MEASure:RESistance[:DC]?**

Syntax :FETCh:RESistance[:DC]?
:MEASure:RESistance[:DC]?

Description Reads the resistance of the instrument.

**Return
Format** The query returns a real number.

**Related
Command** [:\[SOURce\]:RESistance\[:LEVel\]\[:IMMediate\]](#)

**:FETCh:POWer[:DC]?
:MEASure:POWer[:DC]?**

Syntax :FETCh:POWer[:DC]?
:MEASure:POWer[:DC]?

Description Reads the input power of the instrument.

**Return
Format** The query returns a real number.

**Related
Command** [:\[SOURce\]:POWer\[:LEVel\]\[:IMMediate\]](#)

**:FETCh:CAPability?
:MEASure:CAPability?**

Syntax :FETCh:CAPability?
:MEASure:CAPability?

Description Reads the battery capacity.

**Return
Format** The query returns a real number.

**:FETCh:WATThours?
:MEASure:WATThours?**

Syntax :FETCh:WATThours?
:MEASure:WATThours?

Description Reads the battery energy.

**Return
Format** The query returns a real number.

:FETCh:DISChargingTime?
:MEASure:DISChargingTime?

Syntax :FETCh:DISChargingTime?
:MEASure:DISChargingTime?

Description Reads the discharge time of the battery.

Return Format The query returns a real number.

:FETCh:TIME?
:MEASure:TIME?

Syntax :FETCh:TIME?
:MEASure:TIME?

Description Reads the integration time of the instrument.

Return Format It is the integration time of 10 PLC. It is 200, and its unit is ms.

:FETCh:WAVedata?
:MEASure:WAVedata?

Syntax :FETCh:WAVedata?
:MEASure:WAVedata?

Description Reads the data points (400 data points) in the data cache area in the waveform display interface.

Return Format The query returns 400 consecutive data points.

:TRIGger Commands

Command List:

- ◆ [:TRIGger](#)
- ◆ [:TRIGger:SOURce](#)

:TRIGger

Syntax :TRIGger[:IMMediate]

Description When SCPI command trigger (Bus) is selected to be the trigger source, running the command will immediately initiate a trigger.

Related Command [:TRIGger:SOURce](#)

:TRIGger:SOURce

Syntax :TRIGger:SOURce {BUS|EXTeRnal|MANUal}
:TRIGger:SOURce?

Description Selects the trigger source.
Queries trigger source that you set for the instrument.

- Remarks**
- The default trigger source is MANUal.
 - The trigger sources include the following types:
 - BUS: When the load receives the remote triggering command ([:TRIGger](#)) via the interface, the load will perform one trigger operation.
 - EXTeRnal: The digital I/O interface on the rear panel of the load can be used to receive the external trigger signal. When the external trigger is in effect, the load will perform one trigger operation once a low pulse is received over the trigger terminal. The trigger output signal over the digital I/O interface can be used to trigger an external device, such as a digital oscilloscope, an electronic load, and other products.
 - MANUal: In the local operation mode, press the **TRAN** key on the front panel, and then one trigger operation is executed.

Example :TRIG:SOUR BUS /*Selects the Bus trigger source.*/
:TRIG:SOUR? /*Queries the selected trigger source.*/*

:**[SOURce]** Commands

Command List:

- ◆ [:\[SOURce\]:INPut](#)
- ◆ [:\[SOURce\]:FUNction](#)
- ◆ [:\[SOURce\]:FUNction:MODE](#)
- ◆ [:\[SOURce\]:TRANsient](#)
- ◆ [:\[SOURce\]:CURRent\[:LEVel\]\[:IMMediate\]](#)
- ◆ [:\[SOURce\]:CURRent:RANGe](#)
- ◆ [:\[SOURce\]:CURRent:SLEW](#)
- ◆ [:\[SOURce\]:CURRent:SLEW:POSitive](#)
- ◆ [:\[SOURce\]:CURRent:SLEW:NEGative](#)
- ◆ [:\[SOURce\]:CURRent:VON](#)
- ◆ [:\[SOURce\]:CURRent:VLIMt](#)
- ◆ [:\[SOURce\]:CURRent:ILIMt](#)
- ◆ [:\[SOURce\]:CURRent:TRANsient:MODE](#)
- ◆ [:\[SOURce\]:CURRent:TRANsient:ALEVel](#)
- ◆ [:\[SOURce\]:CURRent:TRANsient:BLEVel](#)
- ◆ [:\[SOURce\]:CURRent:TRANsient:AWIDth](#)
- ◆ [:\[SOURce\]:CURRent:TRANsient:BWIDth](#)
- ◆ [:\[SOURce\]:CURRent:TRANsient:FREQuency](#)
- ◆ [:\[SOURce\]:CURRent:TRANsient:PERiod](#)
- ◆ [:\[SOURce\]:CURRent:TRANsient:ADUTy](#)
- ◆ [:\[SOURce\]:VOLTage\[:LEVel\]\[:IMMediate\]](#)
- ◆ [:\[SOURce\]:VOLTage:RANGe](#)
- ◆ [:\[SOURce\]:VOLTage:VLIMt](#)
- ◆ [:\[SOURce\]:VOLTage:ILIMt](#)
- ◆ [:\[SOURce\]:RESistance\[:LEVel\]\[:IMMediate\]](#)
- ◆ [:\[SOURce\]:RESistance:RANGe](#)
- ◆ [:\[SOURce\]:RESistance:VLIMt](#)
- ◆ [:\[SOURce\]:RESistance:ILIMt](#)
- ◆ [:\[SOURce\]:POWER\[:LEVel\]\[:IMMediate\]](#)
- ◆ [:\[SOURce\]:POWER:VLIMt](#)
- ◆ [:\[SOURce\]:POWER:ILIMt](#)
- ◆ [:\[SOURce\]:LIST:MODE](#)
- ◆ [:\[SOURce\]:LIST:RANGe](#)
- ◆ [:\[SOURce\]:LIST:COUNT](#)
- ◆ [:\[SOURce\]:LIST:STEP](#)
- ◆ [:\[SOURce\]:LIST:LEVel](#)

- ◆ [:\[SOURce\]:LIST:WIDTH](#)
- ◆ [:\[SOURce\]:LIST:SLEW](#)
- ◆ [:\[SOURce\]:LIST:END](#)
- ◆ [:\[SOURce\]:WAVE:TIME](#)
- ◆ [:\[SOURce\]:WAVE:TSTep](#)

:[SOURce]:INPut

Syntax :[SOURce]:INPut[:STATe] {0|1|ON|OFF}
:[SOURce]:INPut[:STATe]?

Description Sets the input of the electronic load to be on or off.

Queries whether the input of the electronic load is on or off.

- Remarks**
- Once you power on the load, it will not sink the current immediately, and you need to press the ON/OFF key to turn on the input of the load first. Then, when it reaches the starting voltage (Von), it starts to sink the current.
 - Enabling or disabling the input does not affect the set values of the current parameters.
 - When the input is off, the load has a high impedance.

Return Format The query returns 0 or 1. 0 indicates that the input is off, and 1 indicates that the input is on.

Example :SOUR:INP:STAT 1 /*Sets the input of the electronic load to be on.*/
:SOUR:INP:STAT? /*Queries whether the input of the electronic load is on or off.*/

:[SOURce]:FUNctIon

Syntax :[SOURce]:FUNctIon {CURRent|RESistance|VOLTage|POWer}
:[SOURce]:FUNctIon?

Description Sets the static operation mode of the electronic load.

Queries the static operation mode of the electronic load.

- Remarks**
- CURRent: indicates the constant current (CC) mode.
RESistance: indicates the constant resistance (CR) mode.
VOLTage: indicates the constant voltage (CV) mode.
POWer: indicates the constant power (CP) mode.

Return Format The query returns CC, CV, CR, or CP.

Example :SOUR:FUNC RES /*Sets the static operation mode of the electronic load to be CR mode.*/
:SOUR:FUNC? /*Queries the static operation mode of the electronic load.*/

Related Command [:\[SOURce\]:CURRent:TRANSient:MODE](#)

:[SOURce]:FUNCTION:MODE

Syntax :[SOURce]:FUNCTION:MODE {FIXed|LIST|WAVE|BATTery}
:[SOURce]:FUNCTION:MODE?

Description The input regulation mode setting is controlled by the FUNCTION command, the list value, the waveform display command, or the battery discharge command.

Queries what controls the input regulation mode.

- Remarks**
- FIXed: indicates that the input regulation mode is determined by the FUNCTION command. LIST: indicates that the input regulation mode is determined by the activated list. WAVE: indicates that the input regulation mode is determined by the waveform display command. BATTery: indicates that the input regulation mode is determined by the battery discharge command.
 - The query returns LIST in List mode; returns FIX or BATT in waveform display interface; and returns FIX in other modes.

Return Format The query returns FIX, LIST, or BATT.

Example :SOUR:FUNC:MODE FIX /*Sets the input regulation mode to be controlled by the FUNCTION command.*/
:SOUR:FUNC:MODE? /*Queries what controls the input regulation mode.*/

Related Command [:\[SOURce\]:FUNCTION](#)

:[SOURce]:TRANSient

Syntax :[SOURce]:TRANSient[:STATe] {0|1|ON|OFF}
:[SOURce]:TRANSient[:STATe]?

Description Sets the trigger function to be on or off.

Queries the state of the transient generator.

- Remarks**
- Running this command produces the same effect as pressing the **TRAN** key.
 - In the local operation mode, press the **TRAN** key on the front panel, and then one trigger operation is executed.

Return Format The query returns 0 or 1. 0 indicates that the trigger function is off, and 1 indicates that the trigger function is on.

Example :SOUR:TRAN:STAT 1 /*Sets the trigger function to be on.*/
:SOUR:TRAN:STAT? /*Queries the state of the trigger function.*/

:[SOURce]:CURRent[:LEVel][:IMMediate]

Syntax :[SOURce]:CURRent[:LEVel][:IMMediate] {<value>|MINimum|MAXimum|DEFault}
:[SOURce]:CURRent[:LEVel][:IMMediate]? [MINimum|MAXimum|DEFault]

Description Sets the load's regulated current in CC mode.

Queries the load's regulated current set in CC mode.

- Remarks**
- The load's regulated current refers to the constant current value in CC mode. By default, the load's regulated current is 0 A.
 - In CC mode, when the set load current is greater than the current output from the DUT, short circuit occurs to the DUT.
 - In the setting command, the parameter "MINimum" indicates setting the load's minimum regulated current; the parameter "MAXimum" indicates setting the load's maximum regulated current; and the parameter "DEFault" indicates setting the load's default regulated current.
 - In the query command, the parameter "MINimum" indicates querying the minimum value of the load's regulated current; the parameter "MAXimum" indicates querying the maximum value of the load's regulated current; and the parameter "DEFault" indicates querying the default value of the load's regulated current.

Return Format The query returns a real number.

Example :SOUR:CURR:LEV:IMM 3 /*Sets the load's regulated current in CC mode to be 3 A.*/
:SOUR:CURR:LEV:IMM? /*Queries the load's regulated current set in CC mode.*/

:[SOURce]:CURRent:RANGe

Syntax :[SOURce]:CURRent:RANGe {<value>|MINimum|MAXimum|DEFault}
:[SOURce]:CURRent:RANGe? [MINimum|MAXimum|DEFault]

Description Sets the current range in CC mode and transient operation mode to be a high range or a low one.

Queries the current range set in CC mode and transient operation mode.

- Remarks**
- The default range is a high range.
 - In the setting command, the parameter "MINimum" indicates setting a low range; the parameter "MAXimum" indicates setting a high range; and the parameter "DEFault" indicates setting the default range.
 - In the query command, the parameter "MINimum" indicates querying a low range; the parameter "MAXimum" indicates querying a high range; and the parameter "DEFault" indicates querying the default range.

Return Format The query returns a real number.

Example :SOUR:CURR:RANG 60 /*Sets the current range in CC mode to be 60 A (a high range).*/
:SOUR:CURR:RANG? /*Queries the current range set in CC mode.*/

:[SOURce]:CURRent:SLEW

Syntax :[SOURce]:CURRent:SLEW[:BOTH] {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:CURRent:SLEW[:BOTH]? [MINimum|MAXimum|DEFAULT]

Description Sets the rising and falling slew rate in CC mode.
Queries the rising and falling rate set in CC mode.

- Remarks**
- The rising and falling rate of the load set in CC mode are called the slew rate of the load in CC mode. Its unit is A/μs.
 - In the setting command, the parameter "MINimum" indicates setting the minimum rate; the parameter "MAXimum" indicates setting the maximum rate; and the parameter "DEFAULT" indicates setting the default rate.
 - In the query command, the parameter "MINimum" indicates querying the minimum rate; the parameter "MAXimum" indicates querying the maximum rate; and the parameter "DEFAULT" indicates querying the default rate.

Return Format The query returns a real number.

Example :SOUR:CURR:SLEW:BOTH 0.5 /*Sets the slew rate in CC mode to be 0.5 A/μs.*/
:SOUR:CURR:SLEW:BOTH? /*Queries the slew rate set in CC mode.*/*

:[SOURce]:CURRent:SLEW:POSitive

Syntax :[SOURce]:CURRent:SLEW:POSitive {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:CURRent:SLEW:POSitive? [MINimum|MAXimum|DEFAULT]

Description Sets the rising rate in transient operation mode.
Queries the rising rate set in transient operation mode.

- Remarks**
- The rising rate of the load set in transient operation mode is the slew rate of the load in transient operation mode. Its unit is A/μs.
 - In the setting command, the parameter "MINimum" indicates setting the minimum rising rate; the parameter "MAXimum" indicates setting the maximum rising rate; and the parameter "DEFAULT" indicates setting the default rising rate.
 - In the query command, the parameter "MINimum" indicates querying the minimum rising rate; the parameter "MAXimum" indicates querying the maximum rising rate; and the parameter "DEFAULT" indicates querying the default rising rate.

Return Format The query returns a real number.

Example :SOUR:CURR:SLEW:POS 0.5 /*Sets the rising rate in continuous mode to be 0.5 A/μs.*/
:SOUR:CURR:SLEW:POS? /*Queries the rising rate in continuous mode.*/*

Related Command [:\[SOURce\]:CURRent:SLEW:NEGative](#)

:[SOURce]:CURRent:SLEW:NEGative

Syntax :[SOURce]:CURRent:SLEW:NEGative {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:CURRent:SLEW:NEGative? [MINimum|MAXimum|DEFAULT]

Description Sets the falling rate in transient operation mode.

Queries the falling rate set in transient operation mode.

- Remarks**
- The rising rate of the load set in continuous mode is the slew rate of the load in continuous mode. Its unit is A/μs.
 - In the setting command, the parameter "MINimum" indicates setting the minimum falling rate; the parameter "MAXimum" indicates setting the maximum falling rate; and the parameter "DEFAULT" indicates setting the default falling rate.
 - In the query command, the parameter "MINimum" indicates querying the minimum falling rate; the parameter "MAXimum" indicates querying the maximum falling rate; and the parameter "DEFAULT" indicates querying the default falling rate.

Return Format The query returns a real number.

Example :SOUR:CURR:SLEW:NEG 0.5 /*Sets the falling rate in continuous mode to be 0.5 A/μs.*/
:SOUR:CURR:SLEW:NEG? /*Queries the falling rate in continuous mode.*/*

Related Command [:\[SOURce\]:CURRent:SLEW:POSitive](#)

:[SOURce]:CURRent:VON

Syntax :[SOURce]:CURRent:VON {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:CURRent:VON? [MINimum|MAXimum|DEFAULT]

Description Sets the starting voltage in CC mode.

Queries the starting voltage set in CC mode.

- Remarks**
- When the input voltage increases and reaches a value above the set starting voltage (Von), the load starts to sink the current; when the input voltage decreases and reaches a value below the set starting voltage, the load stops sinking. The unit is V.
 - In the setting command, the parameter "MINimum" indicates setting the minimum Von value; the parameter "MAXimum" indicates setting the maximum Von; and the parameter "DEFAULT" indicates setting the default Von value.
 - In the query command, the parameter "MINimum" indicates querying the minimum Von value; the parameter "MAXimum" indicates querying the maximum Von value; and the parameter "DEFAULT" indicates querying the default Von value.

Return Format The query returns a real number.

Example :SOUR:CURR:VON 5 /*Sets the starting voltage (Von) in CC mode to be 5 V.*/
:SOUR:CURR:VON? /*Queries the starting voltage (Von) set in CC mode.*/*

:[SOURce]:CURRENT:VLIMt

Syntax :[SOURce]:CURRENT:VLIMt {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:CURRENT:VLIMt? [MINimum|MAXimum|DEFAULT]

Description Sets the voltage limit in CC mode.

Queries the voltage limit set in CC mode.

- Remarks**
- The voltage limit refers to the upper limit of the voltage working in CC mode. The unit is V.
 - In the setting command, the parameter "MINimum" indicates setting the minimum voltage; the parameter "MAXimum" indicates setting the maximum voltage; and the parameter "DEFAULT" indicates setting the default voltage limit.
 - In the query command, the parameter "MINimum" indicates querying the minimum voltage; the parameter "MAXimum" indicates querying the maximum voltage; and the parameter "DEFAULT" indicates querying the default voltage limit value.

Return Format The query returns a real number.

Example :SOUR:CURRE:VLIM 5 /*Sets the voltage limit in CC mode to be 5 V.*/
:SOUR:CURRE:VLIM? /*Queries the voltage limit set in CC mode.*/*

:[SOURce]:CURRENT:ILIMt

Syntax :[SOURce]:CURRENT:ILIMt {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:CURRENT:ILIMt? [MINimum|MAXimum|DEFAULT]

Description Sets the current limit in CC mode.

Queries the current limit set in CC mode.

- Remarks**
- The current limit refers to the upper limit of the current working in CC mode. Its unit is A.
 - In the setting command, the parameter "MINimum" indicates setting the minimum current; the parameter "MAXimum" indicates setting the maximum current; and the parameter "DEFAULT" indicates setting the default current limit value.
 - In the query command, the parameter "MINimum" indicates querying the minimum current; the parameter "MAXimum" indicates querying the maximum current; and the parameter "DEFAULT" indicates querying the default current limit value.

Return Format The query returns a real number.

Example :SOUR:CURRE:ILIM 5 /*Sets the current limit in CC mode to be 5 A.*/
:SOUR:CURRE:ILIM? /*Queries the current limit set in CC mode.*/*

:[SOURce]:CURRent:TRANsient:MODE

Syntax :[SOURce]:CURRent:TRANsient:MODE {CONTInuous|PULSe|TOGGle}
:[SOURce]:CURRent:TRANsient:MODE?

Description Sets the transient operation mode in CC mode.
Queries the transient operation mode in CC mode.

Remarks CONTInuous: generates a repetitive pulse stream after the transient generator receives a trigger signal.
PULSe: generates a single pulse after the transient generator receives a trigger signal.
TOGGle: generates a repetitive pulse stream that toggles between two load levels after the transient generator receives a trigger signal.

Return Format The query returns CONTInuous, PULSe, or TOGGle.

Example :SOUR:CURR:TRAN:MODE TOGG /*Sets the transient operation mode in CC mode to be TOGG.*/
:SOUR:CURR:TRAN:MODE? /*Queries the transient operation mode in CC mode.*/*

:[SOURce]:CURRent:TRANsient:ALEVel

Syntax :[SOURce]:CURRent:TRANsient:ALEVel {<value>|MINimum|MAXimum|DEFault}
:[SOURce]:CURRent:TRANsient:ALEVel? [MINimum|MAXimum|DEFault]

Description Sets Level A in transient operation mode.
Queries Level A set in transient operation mode.

Remarks

- In transient operation, the sink current toggles between a high value and a low value. Level A indicates a high value. The unit for Level A is Amphere (A).
- The input value of Level A should be within the set range.
- In the setting command, the parameter "MINimum" indicates setting the minimum Level A; the parameter "MAXimum" indicates setting the maximum Level A; and the parameter "DEFault" indicates setting the default value of Level A.
- In the query command, the parameter "MINimum" indicates querying the minimum Level A; the parameter "MAXimum" indicates querying the maximum Level A; and the parameter "DEFault" indicates querying the default value of Level A.

Return Format The query returns a real number.

Example :SOUR:CURR:TRAN:ALEV 5 /*Sets Level A in transient operation mode to be 5 A.*/
:SOUR:CURR:TRAN:ALEV? /*Queries Level A set in transient operation mode.*/*

Related Command [:\[SOURce\]:CURRent:TRANsient:BLEVel](#)

:[SOURce]:CURRent:TRANSient:BLEVel

Syntax :[SOURce]:CURRent:TRANSient:BLEVel {<value>|MINimum|MAXimum|DEFault}
:[SOURce]:CURRent:TRANSient:BLEVel? [MINimum|MAXimum|DEFault]

Description Sets Level B in transient operation mode.

Queries Level B set in transient operation mode.

- Remarks**
- In transient operation, the sink current toggles between a high value and a low value. Level B indicates a low value. The unit for Level B is Ampere (A).
 - The input value of Level B should be within the set range.
 - In the setting command, the parameter "MINimum" indicates setting the minimum Level B; the parameter "MAXimum" indicates setting the maximum Level B; and the parameter "DEFault" indicates setting the default value of Level B.
 - In the query command, the parameter "MINimum" indicates querying the minimum Level B; the parameter "MAXimum" indicates querying the maximum Level B; and the parameter "DEFault" indicates querying the default value of Level B.

Return Format The query returns a real number.

Example :SOUR:CURR:TRAN:BLEV 5 /*Sets Level B in transient operation mode to be 5 A.*/
:SOUR:CURR:TRAN:BLEV? /*Queries Level B set in transient operation mode.*/

Related Command [:\[SOURce\]:CURRent:TRANSient:ALEVel](#)

:[SOURce]:CURRent:TRANSient:AWIDth

Syntax :[SOURce]:CURRent:TRANSient:AWIDth {<value>|MINimum|MAXimum|DEFault}
:[SOURce]:CURRent:TRANSient:AWIDth? [MINimum|MAXimum|DEFault]

Description Sets the width of Level A in continuous and pulsed transient operation.

Sets the width of Level A in continuous and pulsed transient operation.

- Remarks**
- The width of Level A refers to the time during which the sink current stays at Level A when it switches to Level A in continuous and pulsed transient operation mode.
 - In the setting command, the parameter "MINimum" indicates setting the minimum width of Level A; the parameter "MAXimum" indicates setting the maximum width of Level A; and the parameter "DEFault" indicates setting the default width of Level A.
 - In the query command, the parameter "MINimum" indicates querying the minimum width of Level A; the parameter "MAXimum" indicates querying the maximum width of Level A; and the parameter "DEFault" indicates querying the default width of Level A.

Return Format The query returns a real number.

Example :SOUR:CURR:TRAN:AWID 1 /*Sets the width of Level A in continuous or pulsed transient operation to be 1 ms.*/
:SOUR:CURR:TRAN:AWID? /*Queries the width of Level A in continuous or pulsed transient operation.*/

Related Command [:\[SOURce\]:CURRent:TRANSient:BWIDth](#)

:[SOURce]:CURRent:TRANsient:BWIDth

Syntax :[SOURce]:CURRent:TRANsient:BWIDth {<value>|MINimum|MAXimum|DEFault}
:[SOURce]:CURRent:TRANsient:BWIDth? [MINimum|MAXimum|DEFault]

Description Sets the width of Level B in continuous and pulsed transient operation.
Sets the width of Level B in continuous and pulsed transient operation.

- Remarks**
- The width of Level B refers to the time during which the sink current stays at Level B when it switches to Level B in continuous transient operation mode.
 - In the setting command, the parameter "MINimum" indicates setting the minimum width of Level B; the parameter "MAXimum" indicates setting the maximum width of Level B; and the parameter "DEFault" indicates setting the default width of Level B.
 - In the query command, the parameter "MINimum" indicates querying the minimum width of Level B; the parameter "MAXimum" indicates querying the maximum width of Level B; and the parameter "DEFault" indicates querying the default width of Level B.

Return Format The query returns a real number.

Example :SOUR:CURR:TRAN:BWID 1 /*Sets the width of Level B in continuous or pulsed transient operation to be 1 ms.*/
:SOUR:CURR:TRAN:BWID? /*Queries the width of Level B in continuous or pulsed transient operation.*/

Related Command [:\[SOURce\]:CURRent:TRANsient:AWIDth](#)

:[SOURce]:CURRent:TRANsient:FREQuency

Syntax :[SOURce]:CURRent:TRANsient:FREQuency {<value>|MINimum|MAXimum|DEFault}
:[SOURce]:CURRent:TRANsient:FREQuency? [MINimum|MAXimum|DEFault]

Description Sets the frequency in continuous mode.
Queries the frequency set in continuous mode.

- Remarks**
- Frequency is the reciprocal of period. The unit of frequency is kHz.
 - In the setting command, the parameter "MINimum" indicates setting minimum frequency; the parameter "MAXimum" indicates setting maximum frequency; and the parameter "DEFault" indicates setting the default frequency.
 - In the query command, the parameter "MINimum" indicates querying the minimum frequency; the parameter "MAXimum" indicates querying the maximum frequency; and the parameter "DEFault" indicates querying the default frequency.

Return Format The query returns a real number.

Example :SOUR:CURR:TRAN:FREQ 5 /*Sets the frequency in continuous mode to be 5 kHz.*/
:SOUR:CURR:TRAN:FREQ? /*Queries the frequency set in continuous mode.*/

Related Command [:\[SOURce\]:CURRent:TRANsient:PERiod](#)

:[\[SOURce\]:CURRent:TRANsient:PERiod](#)

Syntax :[\[SOURce\]:CURRent:TRANsient:PERiod](#) {<value>|MINimum|MAXimum|DEFAULT}
 :[\[SOURce\]:CURRent:TRANsient:PERiod?](#) [MINimum|MAXimum|DEFAULT]

Description Sets the period in continuous mode.

Queries the period in continuous mode.

- Remarks**
- Period refers to the sum of time during which the sink current stays at Level A and Level B in continuous mode. The unit for the period is ms.
 - In the setting command, the parameter "MINimum" indicates setting the minimum period; the parameter "MAXimum" indicates setting the maximum period; and the parameter "DEFAULT" indicates setting the default period.
 - In the query command, the parameter "MINimum" indicates querying the minimum period; the parameter "MAXimum" indicates querying the maximum period; and the parameter "DEFAULT" indicates querying the default period.

Return Format The query returns a real number.

Example :SOUR:CURR:TRAN:PER 1 /*Sets the period in continuous mode to be 1 ms.*/
 :SOUR:CURR:TRAN:PER? /*Queries the period set in continuous mode.*/

Related Command :[\[SOURce\]:CURRent:TRANsient:FREQuency](#)

:[\[SOURce\]:CURRent:TRANsient:ADUTy](#)

Syntax :[\[SOURce\]:CURRent:TRANsient:ADUTy](#) {<value>|MINimum|MAXimum|DEFAULT}
 :[\[SOURce\]:CURRent:TRANsient:ADUTy?](#) [MINimum|MAXimum|DEFAULT]

Description Sets the duty cycle in continuous mode.

Queries the duty cycle in continuous mode.

- Remarks**
- Duty cycle refers to the ratio of duration of Level A to the period when the sink current switches to Level A in continuous mode. The command returns an integer ranging from 1 to 100.
 - In the setting command, the parameter "MINimum" indicates setting the minimum duty cycle; the parameter "MAXimum" indicates setting the maximum duty cycle; and the parameter "DEFAULT" indicates setting the default duty cycle.
 - In the query command, the parameter "MINimum" indicates querying the minimum duty cycle; the parameter "MAXimum" indicates querying the maximum duty cycle; and the parameter "DEFAULT" indicates querying the default duty cycle.

Return Format The query returns a real number.

Example :SOUR:CURR:TRAN:ADUT 50 /*Sets the duty cycle in continuous mode to be 50%.*/
 :SOUR:CURR:TRAN:ADUT? /*Queries the duty cycle set in continuous mode.*/

Related Command :[\[SOURce\]:CURRent:TRANsient:AWIDth](#)

:[SOURce]:VOLTage[:LEVel][:IMMediate]

Syntax :[SOURce]:VOLTage[:LEVel][:IMMediate] {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:VOLTage[:LEVel][:IMMediate]? [MINimum|MAXimum|DEFAULT]

Description Sets the load voltage in CV mode.

Queries the load voltage set in CV mode.

- Remarks**
- The load voltage refers to the constant voltage in CV mode. The unit is V.
 - In CV mode, when the set load voltage is greater than the voltage output from the DUT, open circuit occurs to the DUT.
 - In the setting command, the parameter "MINimum" indicates setting the minimum load voltage; the parameter "MAXimum" indicates setting the maximum load voltage; and the parameter "DEFAULT" indicates setting the default load voltage.
 - In the query command, the parameter "MINimum" indicates querying the minimum load voltage; the parameter "MAXimum" indicates querying the maximum load voltage; and the parameter "DEFAULT" indicates querying the default load voltage.

Return Format The query returns a real number.

Example :SOUR:VOLT:LEV:IMM 5 /*Sets the load voltage in CV mode to be 5 V.*/
:SOUR:VOLT:LEV:IMM? /*Queries the load voltage set in CV mode.*/

:[SOURce]:VOLTage:RANGe

Syntax :[SOURce]:VOLTage:RANGe {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:VOLTage:RANGe? [MINimum|MAXimum|DEFAULT]

Description Sets the voltage range in CV mode to be a high range or a low one.

Queries the voltage range set in CV mode.

- Remarks**
- The default range is a high range.
 - In the setting command, the parameter "MINimum" indicates setting a low range; the parameter "MAXimum" indicates setting a high range; and the parameter "DEFAULT" indicates setting the default range.
 - In the query command, the parameter "MINimum" indicates querying a low range; the parameter "MAXimum" indicates querying a high range; and the parameter "DEFAULT" indicates querying the default range.

Return Format The query returns a real number.

Example :SOUR:VOLT:RANG 150 /*Sets the voltage range in CV mode to be 150 V (a high range).*/
:SOUR:VOLT:RANG? /*Queries the voltage range set in CV mode.*/

:[SOURce]:VOLTage:VLIMt

Syntax :[SOURce]:VOLTage:VLIMt {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:VOLTage:VLIMt? [MINimum|MAXimum|DEFAULT]

Description Sets the voltage limit in CV mode.

Queries the voltage limit set in CV mode.

- Remarks**
- The voltage limit refers to the upper limit of the voltage working in CV mode. The unit is V.
 - In the setting command, the parameter "MINimum" indicates setting the minimum voltage; the parameter "MAXimum" indicates setting the maximum voltage; and the parameter "DEFAULT" indicates setting the default voltage limit.
 - In the query command, the parameter "MINimum" indicates querying the minimum voltage; the parameter "MAXimum" indicates querying the maximum voltage; and the parameter "DEFAULT" indicates querying the default voltage limit value.

Return Format The query returns a real number.

Example :SOUR:VOLT:VLIM 5 /*Sets the voltage limit in CV mode to be 5 V.*/
:SOUR:VOLT:VLIM? /*Queries the voltage limit set in CV mode.*/

:[SOURce]:VOLTage:ILIMt

Syntax :[SOURce]:VOLTage:ILIMt {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:VOLTage:ILIMt? [MINimum|MAXimum|DEFAULT]

Description Sets the current limit in CV mode.

Queries the current limit set in CV mode.

- Remarks**
- The current limit refers to the upper limit of the current working in CV mode. Its unit is A.
 - In the setting command, the parameter "MINimum" indicates setting the minimum current; the parameter "MAXimum" indicates setting the maximum current; and the parameter "DEFAULT" indicates setting the default current limit value.
 - In the query command, the parameter "MINimum" indicates querying the minimum current; the parameter "MAXimum" indicates querying the maximum current; and the parameter "DEFAULT" indicates querying the default current limit value.

Return Format The query returns a real number.

Example :SOUR:VOLT:ILIM 5 /*Sets the current limit in CV mode to be 5 A.*/
:SOUR:VOLT:ILIM? /*Queries the current limit set in CV mode.*/

:[SOURCE]:RESistance[:LEVel][:IMMediate]

Syntax :[SOURCE]:RESistance[:LEVel][:IMMediate] {<value>|MINimum|MAXimum|DEFAULT}
:[SOURCE]:RESistance[:LEVel][:IMMediate]? [MINimum|MAXimum|DEFAULT]

Description Sets the load resistance in CR mode.

Queries the load resistance set in CR mode.

- Remarks**
- The load resistance refers to the constant resistance in CR mode. Its unit is Ω .
 - In the setting command, the parameter "MINimum" indicates setting the minimum load resistance; the parameter "MAXimum" indicates setting the maximum load resistance; and the parameter "DEFAULT" indicates setting the default load resistance.
 - In the query command, the parameter "MINimum" indicates querying the minimum load resistance; the parameter "MAXimum" indicates querying the maximum load resistance; and the parameter "DEFAULT" indicates querying the default load resistance.

Return Format The query returns a real number.

Example :SOUR:RES:LEV:IMM 5 /*Sets the load resistance in CR mode to be 5 Ω .*/
:SOUR:RES:LEV:IMM? /*Queries the load resistance set in CR mode.*/

:[SOURCE]:RESistance:RANGe

Syntax :[SOURCE]:RESistance:RANGe {<value>|MINimum|MAXimum|DEFAULT}
:[SOURCE]:RESistance:RANGe? [MINimum|MAXimum|DEFAULT]

Description Sets the resistance range in CR mode to be a high range or a low one.

Queries the resistance range set in CR mode.

- Remarks**
- The default range is a high range.
 - In the setting command, the parameter "MINimum" indicates setting a low range; the parameter "MAXimum" indicates setting a high range; and the parameter "DEFAULT" indicates setting the default range.
 - In the query command, the parameter "MINimum" indicates querying a low range; the parameter "MAXimum" indicates querying a high range; and the parameter "DEFAULT" indicates querying the default range.

Return Format The query returns a real number.

Example :SOUR:RES:RANG 15000 /*Sets the resistance range in CR mode to be 15000 Ω (a high range).*/
:SOUR:RES:RANG? /*Queries the resistance range set in CR mode.*/

:[SOURce]:RESistance:VLIMt

Syntax :[SOURce]:RESistance:VLIMt {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:RESistance:VLIMt? [MINimum|MAXimum|DEFAULT]

Description Sets the voltage limit in CR mode.

Queries the voltage limit set in CR mode.

- Remarks**
- The voltage limit refers to the upper limit of the voltage working in CR mode. The unit is V.
 - In the setting command, the parameter "MINimum" indicates setting the minimum voltage; the parameter "MAXimum" indicates setting the maximum voltage; and the parameter "DEFAULT" indicates setting the default voltage limit.
 - In the query command, the parameter "MINimum" indicates querying the minimum voltage; the parameter "MAXimum" indicates querying the maximum voltage; and the parameter "DEFAULT" indicates querying the default voltage limit value.

Return Format The query returns a real number.

Example :SOUR:RES:VLIM 5 /*Sets the voltage limit in CR mode to be 5 V.*/
:SOUR:RES:VLIM? /*Queries the voltage limit set in CR mode.*/

:[SOURce]:RESistance:ILIMt

Syntax :[SOURce]:RESistance:ILIMt {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:RESistance:ILIMt? [MINimum|MAXimum|DEFAULT]

Description Sets the current limit in CR mode.

Queries the current limit set in CR mode.

- Remarks**
- The current limit refers to the upper limit of the current working in CR mode. Its unit is A.
 - In the setting command, the parameter "MINimum" indicates setting the minimum current; the parameter "MAXimum" indicates setting the maximum current; and the parameter "DEFAULT" indicates setting the default current limit value.
 - In the query command, the parameter "MINimum" indicates querying the minimum current; the parameter "MAXimum" indicates querying the maximum current; and the parameter "DEFAULT" indicates querying the default current limit value.

Return Format The query returns a real number.

Example :SOUR:RES:ILIM 5 /*Sets the current limit in CR mode to be 5 A.*/
:SOUR:RES:ILIM? /*Queries the current limit set in CR mode.*/

:[SOURce]:POWER[:LEVel][:IMMediate]

Syntax :[SOURce]:POWER[:LEVel][:IMMediate] {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:POWER[:LEVel][:IMMediate]? [MINimum|MAXimum|DEFAULT]

Description Sets the load power in CP mode.
Queries the load power set in CP mode.

- Remarks**
- The load power refers to the constant power value in CP mode. The unit of power is W.
 - In the setting command, the parameter "MINimum" indicates setting the minimum load power; the parameter "MAXimum" indicates setting the maximum load power; and the parameter "DEFAULT" indicates setting the default load power.
 - In the query command, the parameter "MINimum" indicates querying the minimum load power; the parameter "MAXimum" indicates querying the maximum load power; and the parameter "DEFAULT" indicates querying the default load power.

Return Format The query returns a real number.

Example :SOUR:POW:LEV:IMM 5 /*Sets the load power in CP mode to be 5 W.*/
:SOUR:POW:LEV:IMM? /*Queries the load resistance set in CP mode.*/

:[SOURce]:POWER:VLIMt

Syntax :[SOURce]:POWER:VLIMt {<value>|MINimum|MAXimum|DEFAULT}
:[SOURce]:POWER:VLIMt? [MINimum|MAXimum|DEFAULT]

Description Sets the voltage limit in CP mode.
Queries the voltage limit set in CP mode.

- Remarks**
- The voltage limit refers to the upper limit of the voltage working in CP mode. The unit is V.
 - In the setting command, the parameter "MINimum" indicates setting the minimum voltage; the parameter "MAXimum" indicates setting the maximum voltage; and the parameter "DEFAULT" indicates setting the default voltage limit.
 - In the query command, the parameter "MINimum" indicates querying the minimum voltage; the parameter "MAXimum" indicates querying the maximum voltage; and the parameter "DEFAULT" indicates querying the default voltage limit value.

Return Format The query returns a real number.

Example :SOUR:POW:VLIM 5 /*Sets the voltage limit in CP mode to be 5 V.*/
:SOUR:POW:VLIM? /*Queries the voltage limit set in CP mode.*/

:[SOURce]:POWER:ILIMt

Syntax :[SOURce]:POWER:ILIMt {<value>|MINimum|MAXimum|DEFault}
:[SOURce]:POWER:ILIMt? [MINimum|MAXimum|DEFault]

Description Sets the current limit in CP mode.

Queries the current limit set in CP mode.

- Remarks**
- The current limit refers to the upper limit of the current working in CP mode. Its unit is A.
 - In the setting command, the parameter "MINimum" indicates setting the minimum current; the parameter "MAXimum" indicates setting the maximum current; and the parameter "DEFault" indicates setting the default current limit value.
 - In the query command, the parameter "MINimum" indicates querying the minimum current; the parameter "MAXimum" indicates querying the maximum current; and the parameter "DEFault" indicates querying the default current limit value.

Return Format The query returns a real number.

Example :SOUR:POW:ILIM 5 /*Sets the current limit in CP mode to be 5 A.*/
:SOUR:POW:ILIM? /*Queries the current limit set in CP mode.*/

:[SOURce]:LIST:MODE

Syntax :[SOURce]:LIST:MODE {CC|CV|CR|CP}
:[SOURce]:LIST:MODE?

Description Sets the running mode of the load in List operation.

Queries the running mode of the load set in List operation.

- Remarks**
- In list operation mode, you can perform the complex current/voltage/resistance/power modes accurately and rapidly, which may be synchronized with internal or external signals. In this way, you can complete the sophisticated test.
 - The list function supports CC, CV, CR, and CP modes.

Return Format The query returns CC, CV, CR, or CP.

Example :SOUR:LIST:MODE CC /*Sets the load to run in CC mode in List operation.*/
:SOUR:LIST:MODE? /*Queries the running mode of the load in List operation.*/

:[SOURce]:LIST:RANGe

Syntax :[SOURce]:LIST:RANGe <value>
:[SOURce]:LIST:RANGe?

Description Sets the range for each running mode in List operation.
Queries the range of the mode set in List operation.

- Remarks**
- Selects a different range based on the current working mode of List. There is no range available in CP mode.
 - Ranges for different modes:
CC: 0-6 A/0-60 A
CV: 0-15 V/0-150 V
CR: 0-15 Ω /0-15000 Ω

Return Format The query returns a real number.

Example :SOUR:LIST:RANG 60 /*Sets the current range for CC mode in List operation to be 60 A.*/
:SOUR:LIST:RANG? /*Queries the current range for CC mode in List operation.*/

Related Command [:\[SOURce\]:LIST:MODE](#)

:[SOURce]:LIST:COUNT

Syntax :[SOURce]:LIST:COUNT {<value>|MINimum|MAXimum}
:[SOURce]:LIST:COUNT? [MINimum|MAXimum]

Description Sets the number of times the list is cycled.
Queries the number of times the list is cycled.

- Remarks**
- The number of times the list is cycled (or the execution times) refers to the times when timing input is completed based on the preset current/voltage/resistance/power.
 - Its range is from 0 to 99999.
 - When the value is set to 0, the execution time is displayed to be "Infinite", and then it will be switched to the infinite mode.

Return Format The query returns a real number.

Example :SOUR:LIST:COUN 3 /*Sets the number of times the list is cycled to be 3.*/
:SOUR:LIST:COUN? /*Queries the number of times the list is cycled.*/

:[SOURce]:LIST:STEP

Syntax :[SOURce]:LIST:STEP {<value>|MINimum|MAXimum}
:[SOURce]:LIST:STEP? [MINimum|MAXimum]

Description Sets the number of steps executed in each cycle.
Queries the number of steps executed in each cycle.

- Remarks**
- Its range is from 2 to 512.
 - The step value starts from 0.
 - Total Steps = Steps*Cycles
 - You can edit up to 512 groups of data.

Return Format The query returns a real number.

Example :SOUR:LIST:STEP 2 /*Sets the number of steps executed in each cycle to be 3.*/
:SOUR:LIST:STEP? /*Queries the number of steps executed in each cycle.*/

:[SOURce]:LIST:LEVel

Syntax :[SOURce]:LIST:LEVel <step>,<value>
:[SOURce]:LIST:LEVel? <step>

Description Sets the input value for each step.
Queries the set input value for each step.

- Remarks**
- The step value starts from 0.
 - In CC mode, the set value is for current, and its default unit is A.
In CV mode, the set value is for voltage, and its default unit is V.
In CR mode, the set value is for resistance, and its default unit is Ω .
In CP mode, the set value is for power, and its default unit is W.

Return Format The query returns a real number.

Example :SOUR:LIST:LEV 3,1.5 /*Sets the input value for Step 4 to 1.5 A.*/
:SOUR:LIST:LEV? 3 /*Queries the input value set for Step 4.*/

:[SOURce]:LIST:WIDth

Syntax :[SOURce]:LIST:WIDth <step>,<value>
:[SOURce]:LIST:WIDth? <step>

Description Sets the width for each step.
Queries the width set for each step.

- Remarks**
- Width refers to the dwell time for each step. Its unit is s.
 - The step value starts from 0.
 - Its settable range is from 0.00005 s to 3600 s.

Return Format The query returns a real number.

Example :SOUR:LIST:WID 3,3 /*Sets the width for Step 4 to 3 s.*/
:SOUR:LIST:WID? 3 /*Queries the width for Step 4.*/*

:[SOURce]:LIST:SLEW

Syntax :[SOURce]:LIST:SLEW <step>,<value>
:[SOURce]:LIST:SLEW? <step>

Description Sets the slew rate for CC mode in List operation.
Queries the set slew rate.

- Remarks**
- It is only applicable to CC mode. The default unit for the slew rate is A/μs.
 - The step value starts from 0.

Return Format The query returns a real number.

Example :SOUR:LIST:SLEW 3,0.5 /*Sets the slew rate for Step 4 in CC mode to 0.5 A/μs.*/
:SOUR:LIST:SLEW? 3 /*Queries the slew rate set for Step 4.*/*

Related Command [:\[SOURce\]:LIST:MODE](#)

:[SOURce]:LIST:END

Syntax :[SOURce]:LIST:END {LAST|OFF}

: [SOURce]:LIST:END?

Description Sets the end state when input is completed in list operation.

Queries the set end rate.

- Remarks**
- End state: when the number of execution times is a finite value, last state refers to the state the load stays after the load completes the total steps of current/voltage/resistance/power inputs.
 - OFF: indicates that the input will be turned off automatically after the input is completed.
- LAST: keeps the input state of the last group after the input is completed.

Return The query returns LAST or OFF.

Format

Example :SOUR:LIST:END LAST /*Sets the end state to be "LAST" when input is completed in list operation.*/

:SOUR:LIST:END? /*Queries the set end state.*/

:[SOURce]:WAVe:TIME

Syntax :[SOURce]:WAVe:TIME{ADD|SUB}
:[SOURce]:WAVe:TIME?

Description Sets the window time in the waveform display interface.
Queries the set window time.

- Remarks**
- ADD: indicates increasing the time value. SUB: indicates decreasing the time value.
 - When the time that you set is greater than 120 s, it will be automatically displayed in minute; when greater than 120 min, it will be automatically displayed in hour.

Return Format The query returns a real number.

Example :SOUR:WAV:TIM ADD /*Increases the window time in the waveform display interface.*/
:SOUR:WAV:TIM? /*Queries the set window time.*/*

Related Command [:\[SOURce\]:FUNCTION:MODE](#)

:[SOURce]:WAVe:TSTep

Syntax :[SOURce]:WAVe:TSTep{1|10}
:[SOURce]:WAVe: TSTep?

Description Sets the time step scale in the waveform display interface.
Queries the set time step scale.

- Remarks**
- The time step scale includes *1 and *10. When the time step is set to "*1", it indicates that the window time increases or decreases with a step size of 1; when the time step is set to "*10", it indicates that the window time increases or decreases with a step size of 10.
 - When the window time unit is s, the time step scale *1 is not supported.

Return Format 1 or 10

Example :SOUR:WAV:TST 1 /*Sets the time step scale in the waveform display interface to *1.*/
:SOUR:WAV:TST? /*Queries the set time step scale.*/*

Related Command [:\[SOURce\]:FUNCTION:MODE](#)

:SYSTem Commands

Command List:

- ◆ [:SYSTem:KEY](#)
- ◆ [:SYSTem:ERRor?](#)
- ◆ [:SYSTem:VERsion?](#)
- ◆ [:SYSTem:IDN](#)

:SYSTem:KEY

Syntax :SYSTem:KEY <keyval>

Description Simulates the keys on the front panel remotely.

Remarks ● keyval is the key value. For its detailed key name, refer to the following table.

Key Value	Key Name	Key Value	Key Name
0	CC	22	Numeric key 2
1	CV	23	Numeric key 3
2	CR	24	Numeric key 4
3	CP	25	Numeric key 5
4	Con	26	Numeric key 6
5	Pul	27	Numeric key 7
6	Tog	28	Numeric key 8
7	List	29	Numeric key 9
8	Local	30	Decimal point (.)
9	Utility	31	Backspace key
10	Option	32	ON/OFF
11	Store	33	SHORT
12	Help	34	TRAN
13	APP	35	Rotate the knob counterclockwise
14	First menu key (from left to right)	36	Rotate the knob clockwise
15	Second menu key	37	Left arrow key
16	Third menu key	38	Right arrow key
17	Fourth menu key	39	Up arrow key
18	Fifth menu key	40	Down arrow key
19	Sixth menu key	41	OK
20	Numeric key 0	42	Waveform display key
21	Numeric key 1		

- When you use the virtual panel, first run the :DEBUg:KEY {0|1|ON|OFF} command to enable the virtual panel (when the parameter is set to 1 or ON), and then you can operate on the virtual panel. If you do not need to use the virtual panel, run the :DEBUg:KEY {0|1|ON|OFF} command to disable the virtual panel (when the parameter is set to 0 or OFF).

Example :SYST:KEY 2 /*Simulates the CR key on the front panel remotely.*/

:SYSTem:ERRor?

Syntax :SYSTem:ERRor?

Description Queries the last error message in the error queue and clears the error message.

Remarks You can also send the [*RST](#) command to restore the instrument to factory default settings and clear the error queue.

Return Format The query returns the number and contents of the error message, such as -113,"Undefined header; keyword cannot be found".

:SYSTem:VERSion?

Syntax :SYSTem:VERSion?

Description Queries the SCPI version of the current system.

Return Format The query returns the SCPI version of the current system in strings, and its format is YYYY.V. Of which, YYYY indicates the year, and V indicates the version number for the year. For example, 1999.0.

Example :SYST:VERS? /*Queries the SCPI version of the current system.*/

:SYSTem:IDN

Syntax :SYSTem:IDN:SET <manufacturer>,<model>,<sn>,<firmware>

Description Self-defines the required identification string to be returned through this command.

Remarks Identification string contains four comma separated fields and conforms to the IEEE-488 standards. After the system is reset, the identification string will be cleared.

Example SYSTem:IDN:SET RIGOL,DL3031A,DL3000A000001,00.01.00.04.05
/*Defines the manufacturer, the product model, the serial number, and the firmware version.*/

:LXI Commands

Command List:

- ◆ [LXI:IDENtify\[:STATE\]](#)
- ◆ [LXI:MDNS\[:ENABLE\]](#)
- ◆ [LXI:RESet](#)
- ◆ [LXI:REStart](#)

LXI:IDENtify[:STATE]

Syntax LXI:IDENtify[:STATE] {0|1|ON|OFF}
LXI:IDENtify[:STATE]?

Description Sets the LXI identifier in the interface to be on or off.
Queries the state of the LXI identifier in the interface.

- Remarks**
- Only when DL3000 is connected to the Internet via the LAN cable, can this command be effective in setting the LXI identifier to be on.
 - Send the [*RST](#) command to restore the instrument to factory defaults, and then the LXI identifier can be turned off.

Return Format The query returns 0 or 1. 0 indicates that the input is off, and 1 indicates that the input is on.

Example LXI:IDEN:STATE 1 /*Sets the LXI identifier in the interface to be on.*/
LXI:IDEN:STATE? /*Queries the state of the LXI identifier in the interface.*/

LXI:MDNS[:ENABLE]

Syntax LXI:MDNS[:ENABLE] {0|1|ON|OFF}
LXI:MDNS[:ENABLE]?

Description Sets MDNS to be on or off.
Queries the state of MDNS.

- Remarks**
- Only when DL3000 is connected to the Internet via the LAN cable, can this command be effective in setting the LXI identifier to be on.
 - Send the [*RST](#) command to restore the instrument to factory defaults, and then the MDNS can be disabled.

Return Format The query returns 0 or 1. 0 indicates that the input is off, and 1 indicates that the input is on.

Example LXI:MDNS:ENAB 1 /*Sets MDNS to be on.*/
LXI:MDNS:ENAB? /*Queries the state of MDNS.*/

LXI:RESet

Syntax LXI:RESet

Description Restores the LAN settings to defaults.

Remarks The default state of LAN is as follows:

DHCP:ON

AutoIP:ON

ManualIP:OFF

LXI:REStart

Syntax LXI:REStart

Description Applies the currently set parameters.

:LIC Command

The :LIC command is used to install the option, and it is applicable to DL3021 and DL3031. If you need any options, please purchase them and install them properly. For DL3021A and DL3031A, they have been installed with the following options: high frequency, high slew rate, LAN, high readback resolution, and Digital I/O before leaving the factory. You do not need to install them by yourself.

- LAN: Connect the load to the PC or the local area network (LAN) where the PC resides to realize remote control. The order number is LAN-DL3.
- Digital I/O: provides the trigger input and output function; and the order number is DIGITALIO-DL3.
- High readback resolution: improves the resolution of the instrument; and the order number is HIRES-DL3.
- High slew rate: provides the high slew rate option function, and the order number is SLEWRATE-DL3.
- High frequency: provides the high frequency option function, and the order number is FREQ-DL3.

:LIC:SET

Syntax :LIC:SET <sn>

Description Installs an option.

Parameter

Name	Type	Range	Default
<sn>	ASCII String	Refer to "Remarks".	None

Remarks

- To install an option, you need an option license. <sn> is the option license, and each instrument has one unique license. It is a 28-character string, which can only contain uppercase letters and numbers.
- To obtain the option license, first purchase the required option to obtain the key, and then use the key to generate the option license according to the following steps.
 - Log in to the **RIGOL** official website (www.rigol.com), click **SERVICE** → **Software License Register** to enter the "Registered product license code" interface.
 - Enter the correct key, serial number (press **Utility** → **System Info** to get the instrument serial number), and verification code in the product license register interface, click **Generate** to acquire the option license. (**Note:** The hyphens in the license should be omitted.)

Example :LIC:SET UVF2L3N3XXKYTB73PPRSA4XDMSRT

Related Command [*OPT?](#)

Chapter 3 Application Instances

This chapter provides the application instances of the SCPI commands. The main functions of the load can be realized through a series of SCPI commands.

Note:

1. The instances in this chapter take DL3031A as an example. The range of certain parameters for other models may be different. Therefore, you need to adjust the parameter range for the model that you use if necessary.
2. Before using the instances in this chapter, refer to "To Build Remote Communication" to set up remote communication between the electronic load and the PC. In addition, you have to install Ultra Sigma or other PC software that can be used to send commands.
3. In each instance, every command is followed by contents enclosed by two slashes (*"/** and **/*). They are the descriptions of the command and not part of the command, which help you understand the command better.

LIST Function

Requirement

Functions to be realized:

1. Parameters: mode (CC), range (6 A); circle (2); steps (3); trigger source (MANUal), end state (LAST). Three groups of list parameters are shown below:

Steps	Input Current Setting Value	Duration Time	Slew Rate
1	1 A	3 s	0.1 A/μs
2	1.2 A	5 s	0.3 A/μs
3	1.8 A	3.5 s	0.2 A/μs

2. Enable trigger input.

Method

- | | | |
|------|-----------------------|--|
| (1) | *IDN? | <i>/*Queries the ID string of the load to test whether the remote communication works normally.*/*</i> |
| (2) | :SOUR:LIST:MODE CC | <i>/*Sets the operation mode of the load to be CC mode.*/*</i> |
| (3) | :SOUR:LIST:RANG 6 | <i>/*Sets the load's current range in CC mode to be 6 A.*/*</i> |
| (4) | :SOUR:LIST:COUN 2 | <i>/*Sets the number of times the list is cycled to be 2.*/*</i> |
| (5) | :SOUR:LIST:STEP 2 | <i>/*Sets the total steps to be 3.*/*</i> |
| (6) | :SOUR:LIST:END LAST | <i>/*Sets the end state of the load to be Last.*/*</i> |
| (7) | :SOUR:LIST:LEV 0,1 | <i>/*Sets the input current setting at Step 1 to be 1 A.*/*</i> |
| (8) | :SOUR:LIST:WID 0,3 | <i>/*Sets the dwell time at Step 1 to be 3 s.*/*</i> |
| (9) | :SOUR:LIST:SLEW 0,0.1 | <i>/*Sets the slew rate at Step 1 to be 0.1 A/μs.*/*</i> |
| (10) | :SOUR:LIST:LEV 1,1.2 | <i>/*Sets the input current setting at Step 2 to be 1.2 A.*/*</i> |
| (11) | :SOUR:LIST:WID 1,5 | <i>/*Sets the dwell time at Step 2 to be 5 s.*/*</i> |
| (12) | :SOUR:LIST:SLEW 1,0.3 | <i>/*Sets the slew rate at Step 2 to be 0.3 A/μs.*/*</i> |
| (13) | :SOUR:LIST:LEV 2,1.8 | <i>/*Sets the input current setting at Step 3 to be 1.8 A.*/*</i> |
| (14) | :SOUR:LIST:WID 2,3.5 | <i>/*Sets the dwell time at Step 3 to be 3.5 s.*/*</i> |

-
- (15) :SOUR:LIST:SLEW 2,0.2 /*Sets the slew rate at Step 3 to be 0.2 A/ μ s.*/
(16) :TRIG:SOUR MANU /*Sets the trigger source of the load to be manual. Pressing the **TRAN** key can enable the trigger.*/
(17) :SOUR:INP:STAT 1 /*Sets the input of the load to be on.*/>

Chapter 4 Programming Instances

This chapter illustrates how to program DL3000 series with SCPI commands based on NI-VISA in Excel, MATLAB, LabVIEW, Visual C++, and Visual C#.

NI-VISA (National Instrument-Virtual Instrument Software Architecture), developed by NI (National Instrument), provides an advanced programming interface to communicate with various instruments through their bus lines. NI-VISA enables you to communicate with the instrument in the same way, without considering the interface type of the instrument (such as GPIB, USB, LAN/Ethernet, or RS232).

The instruments that communicate with NI-VISA through various interfaces are called "Resources". The VISA descriptor (i.g. resource name) describes the accurate name and location of the VISA resource. For example, when you use the LAN interface to communicate with the instrument, the VISA descriptor is displayed as ":TCPIP::172.16.3.93::INSTR". Before programming, please obtain the correct VISA descriptor.

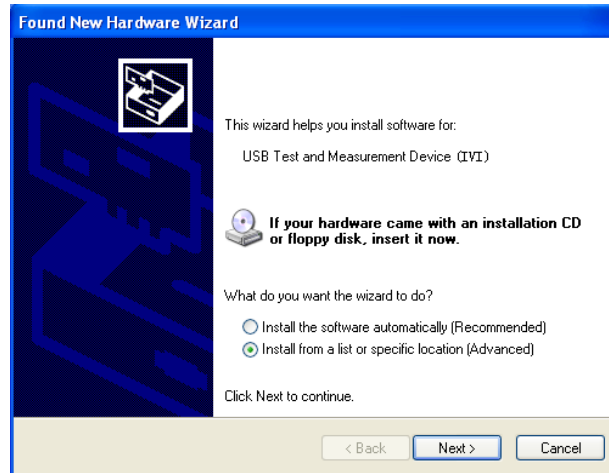
Contents in this chapter:

- ◆ [Programming Preparations](#)
- ◆ [Excel Programming](#)
- ◆ [MATLAB Programming Instance](#)
- ◆ [LabVIEW Programming Instance](#)
- ◆ [Visual C++ Programming Instance](#)
- ◆ [Visual C# Programming Instance](#)

Programming Preparations

Before programming, you need to prepare the following tasks:

1. Install Ultra Sigma (PC) software. You can log in to **RIGOL** official website (www.rigol.com) to download the software. Then install the software according to the installation wizard. After Ultra Sigma is installed successfully, NI-VISA library will be completely installed automatically. In this manual, the default installation path is C:\Program Files\IVI Foundation\VISA.
2. In this manual, the electronic load communicates with the PC via the USB interface. Use the USB cable to connect the load to the PC via the USB DEVICE interface on the rear panel of the load. You can also use LAN, RS232, or GPIB (expanded with **RIGOL**'s optional USB-GPIB interface converter) to remotely communicate with the PC. Note: The RS232 protocol command ends with "\r\n" for the DL3000 series.
3. After the load is properly connected to the PC, connect the load to power source and turn it on.
4. In this case, "Found New Hardware Wizard" dialog box appears on the PC. Please install "USB Test and Measurement Device (IVI)" according to the instructions.



5. Obtain the USB VISA descriptor of the load. Press **Utility**, and then the VISA descriptor is displayed at the bottom of the interface. The VISA descriptor of the load used in this instance is `USB0::0x1AB1::0x0E11::DL3000A000001::INSTR`.

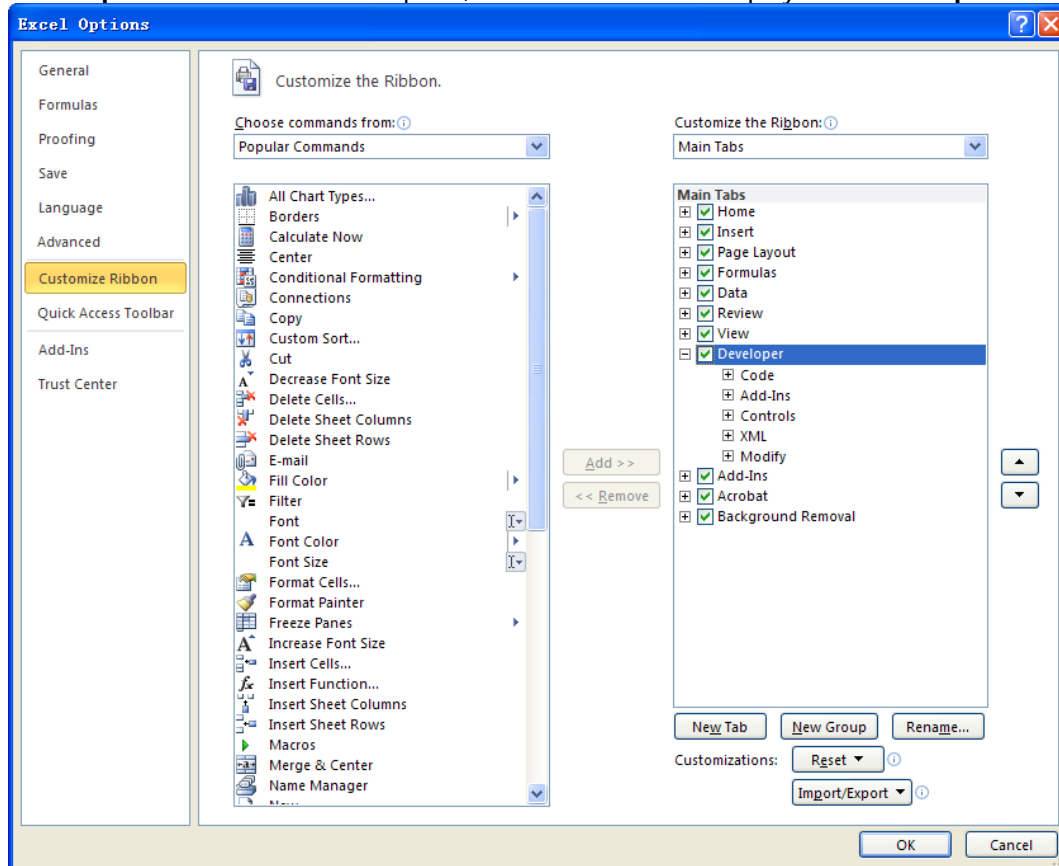
By now, the programming preparations are finished.

Excel Programming Instance

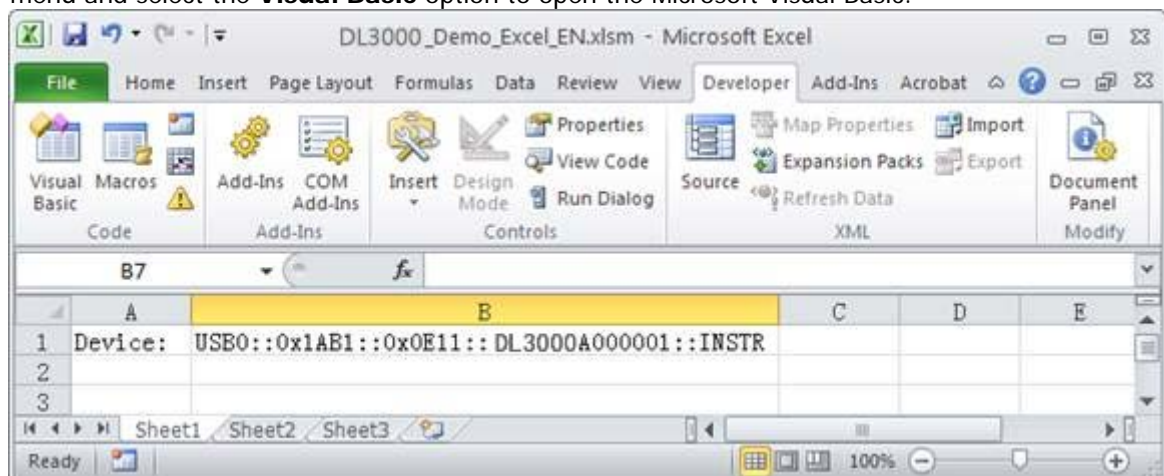
Program used in this instance: Microsoft Excel 2007

Function realized in this example: sending the *IDN? command and reading the instrument information.

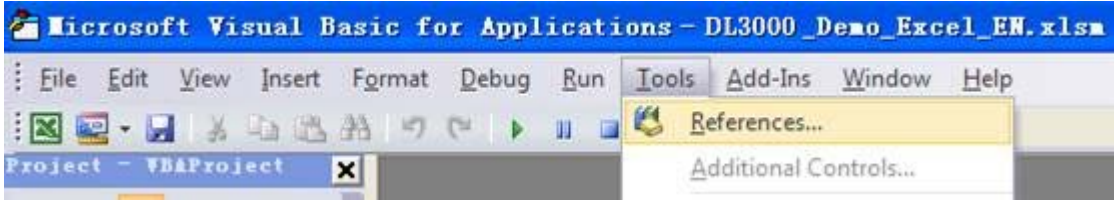
1. Create a new Excel file that enables the Macros, and name it "DL3000_Demo_Excel.xlsm".
2. Run the DL3000_Demo_Excel.xlsm file. Click **File**→**Options** at the upper-left corner of the Excel file to open the interface as shown in the figure below. Click **Customize Ribbon** at the left, check **Developer** and click **OK**. At this point, the Excel menu bar displays the **Developer** menu.



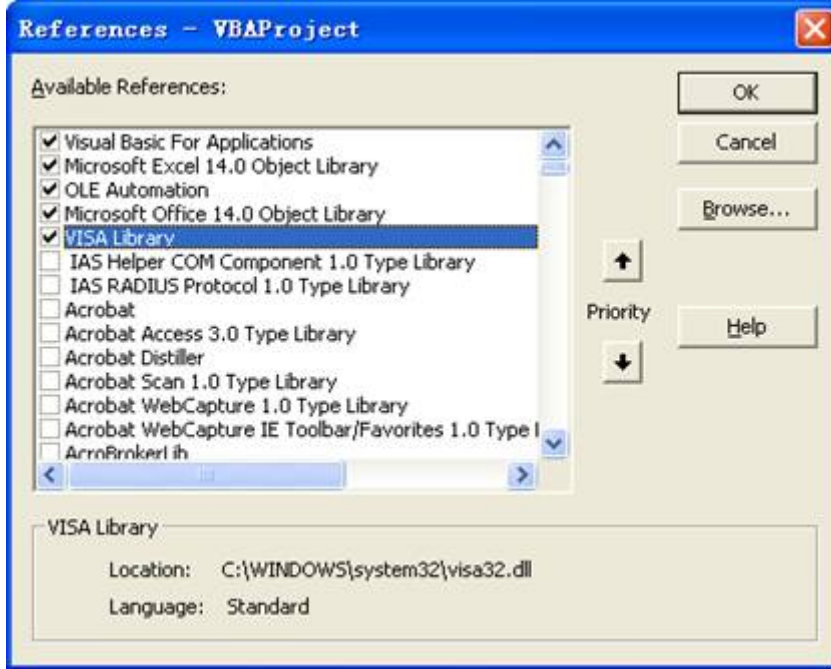
3. Input the VISA descriptor into a cell of the file, as shown in the figure below. Click the **Developer** menu and select the **Visual Basic** option to open the Microsoft Visual Basic.



- 4. Select **Tools(T)** in the Microsoft Visual Basic menu bar and click **References**.



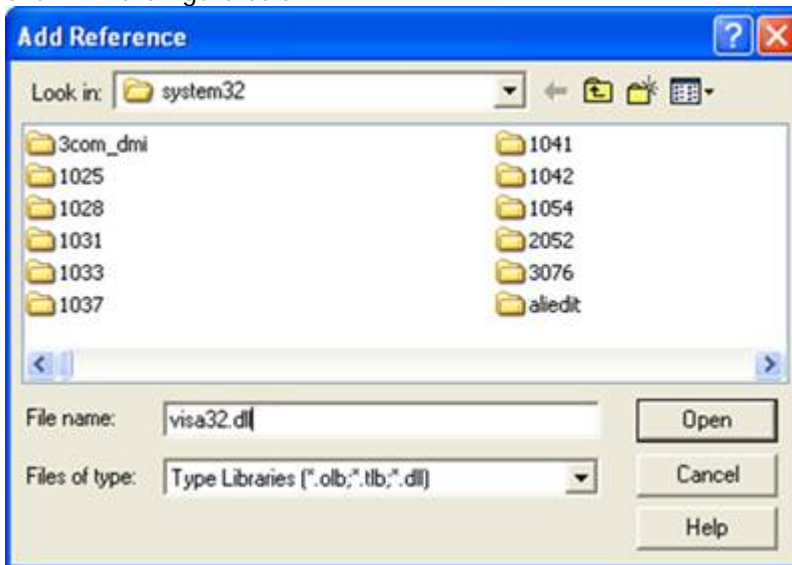
In the displayed dialog box, select **VISA Library**, and click **OK** to refer to the VISA Library.



Remarks:

If you cannot find VISA Library in the left section of the above dialog box, please follow the method below to find it.

- (1) Make sure that your PC has installed the NI-VISA library.
- (2) Click **Browser...** at the right section to search "visa32.dll" from "C:\WINDOWS\system32", as shown in the figure below.



- Click **View Code** under **Developer** menu to enter the interface of Microsoft Visual Basic. Add the following codes and save the file.

Note: If the Excel file created at Step 2 does not enable the Macros, a prompt message "The following features cannot be saved in macro-free workbooks" will be displayed. In this case, please save the file as a macro-enabled file type (filename with a suffix of ".xlsm").

Sub QueryIdn()

```
Dim viDefRm As Long
Dim viDevice As Long
Dim viErr As Long
Dim cmdStr As String
Dim idnStr As String * 128
Dim ret As Long
```

'Turn on the device, and the device resource descriptor is in CELLS(1,2) of SHEET1'

```
viErr = visa.viOpenDefaultRM(viDefRm)
viErr = visa.viOpen(viDefRm, Sheet1.Cells(1, 2), 0, 5000, viDevice)
```

'Send request, read the data, and the return value is in CELLS(2,2) of SHEET1'

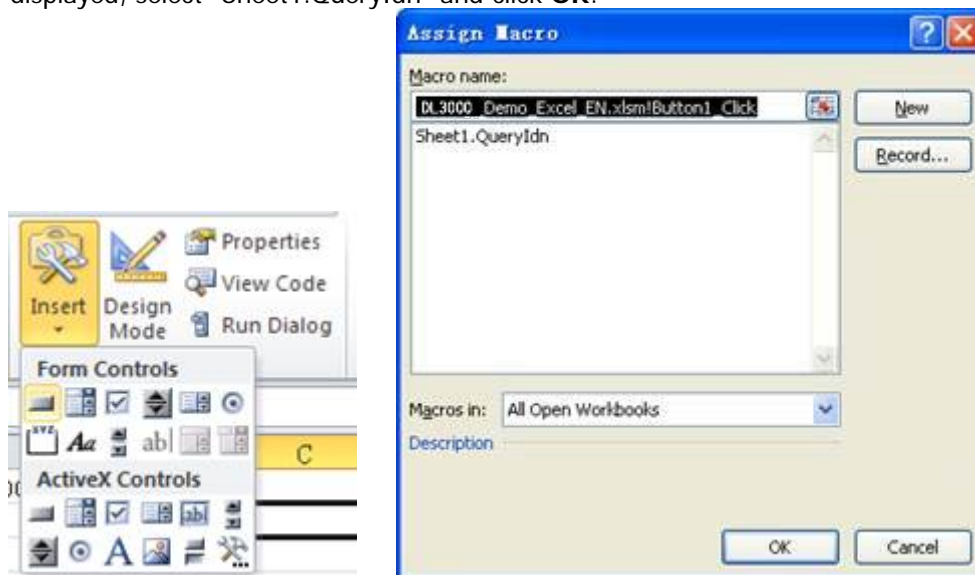
```
cmdStr = "*IDN?"
viErr = visa.viWrite(viDevice, cmdStr, Len(cmdStr), ret)
viErr = visa.viRead(viDevice, idnStr, 128, ret)
Sheet1.Cells(2, 2) = idnStr
```

'Turn off the device'

```
visa.viClose (viDevice)
visa.viClose (viDefRm)
```

End Sub

- Add the button control, and click "Insert" under the Developer menu. Select a button control under the "Form Control" menu item and put it into the Excel cell. At this time, the **Assign Macro** dialog box is displayed, select "Sheet1.QueryIdn" and click **OK**.



The default name of the button is "Button1". Right-click the button and select **Edit Text** in the pop-up menu to change the button name to "*IDN?".

- Click ***IDN?** to run the program. The return value is displayed in CELLS(2,2) of SHEET1.

MATLAB Programming Instance

Program used in this example: MATLAB R2009a

Function realized in this example: sending the *IDN? command and reading the instrument information.

1. Run the MATLAB software and modify the current path (modify the **Current Directory** at the top of the software). In this example, modify the current path to E:\DL3000_Demo.
2. Click **File → New → Blank M-File** in the MATLAB interface to create an empty M file.
3. Add the following codes to the M file:

```
dl3000= visa('ni','USB0::0x1AB1::0x0E11::DL3000A000001::INSTR'); %create VISA object
```

```
fopen(dl3000); %open the created VISA object
```

```
fprintf(dl3000, '*IDN?'); %send request
```

```
meas_CH1 = fscanf(dl3000); %read data
```

```
fclose(dl3000); %close the VISA object
```

```
display(IDN) %display the device information already read
```

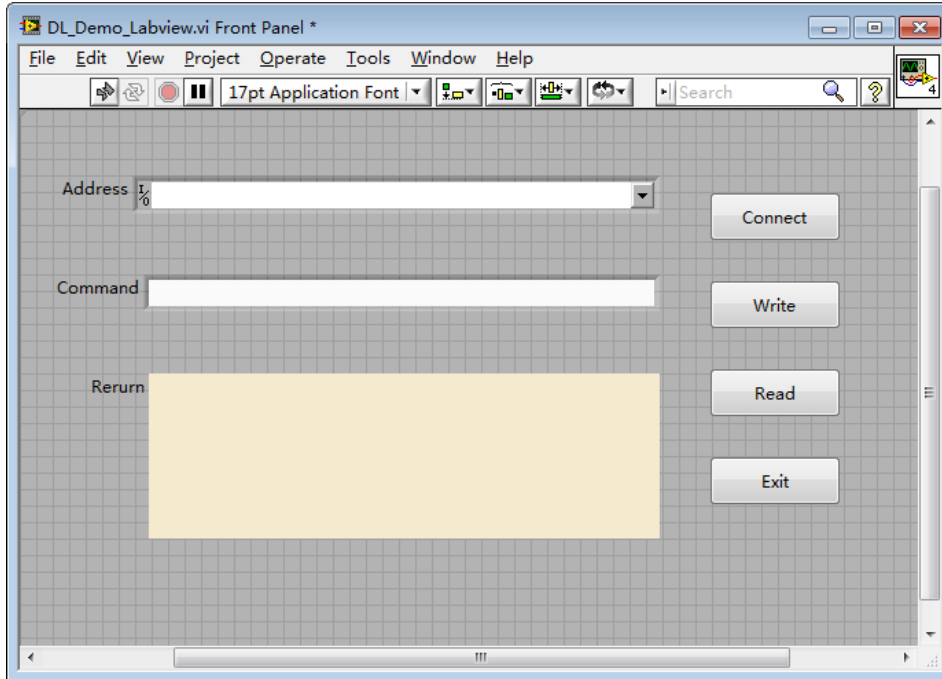
4. Save the M file to the current path. In this instance, the M file is named as "dl3000_Demo_MATLAB.m".
5. Run the M file and the running results are displayed as follows:
IDN=
RIGOL TECHNOLOGIES,DL3031A,LS000001,00.01.00.04.05

LabVIEW Programming Instance

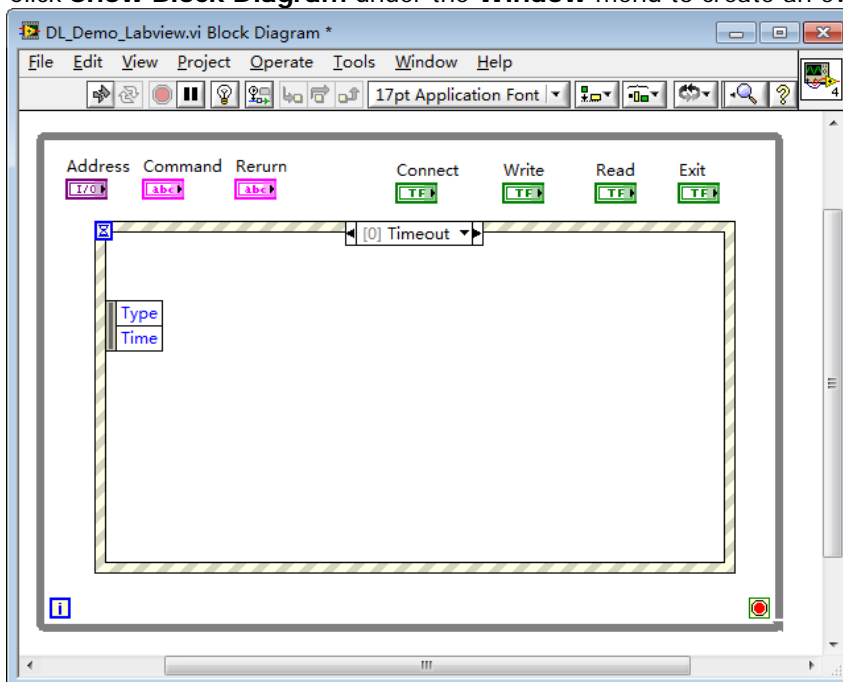
Program used in this example: LabVIEW 2010

Function realized in this instance: search for the instrument address, connect to the instrument, send and read commands.

1. Run LabVIEW 2010, and then create a VI file named DL_Demo_LABVIEW.
2. Add controls to the front panel interface, including the **Address** field, **Command** field, and **Return** field, the **Connect** button, the **Write** button, the **Read** button, and the **Exit** button.

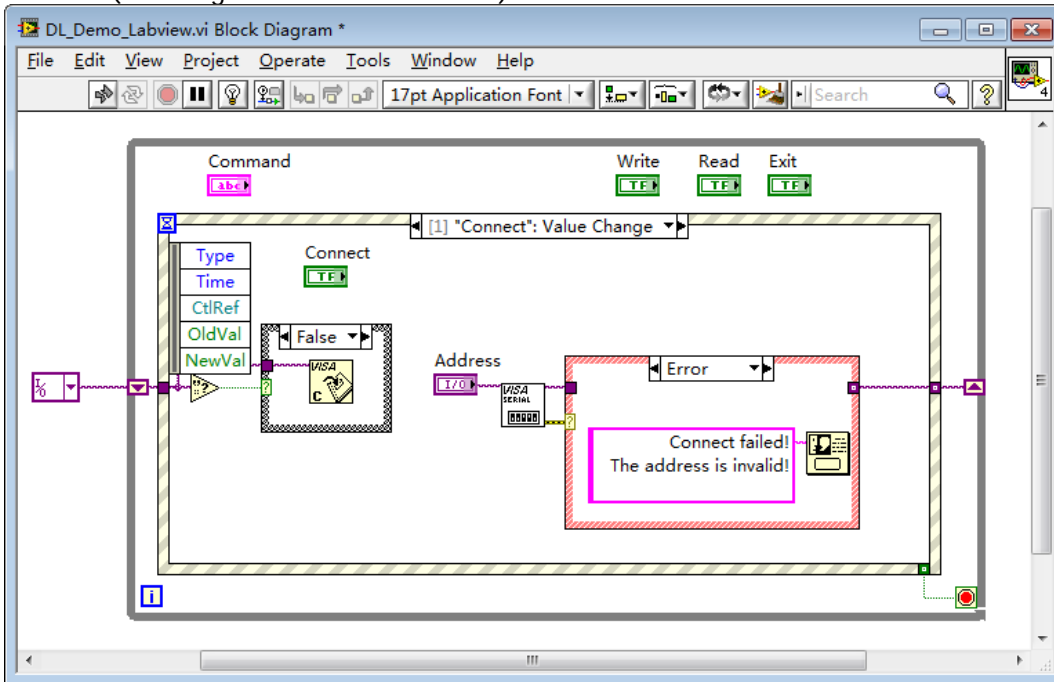


3. Click **Show Block Diagram** under the **Window** menu to create an event structure.

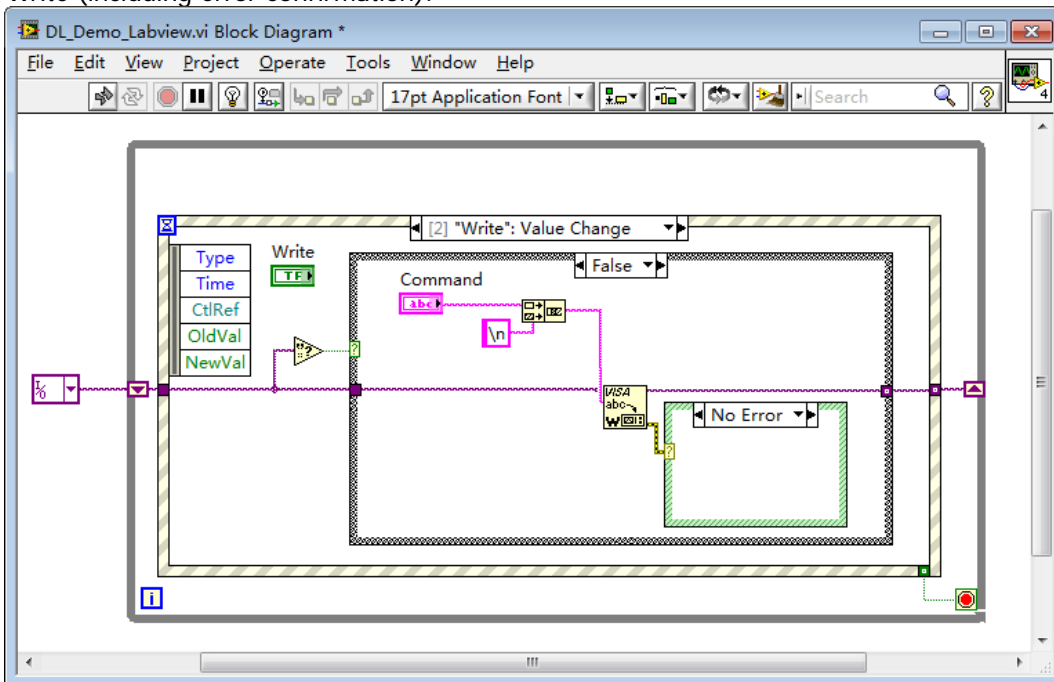


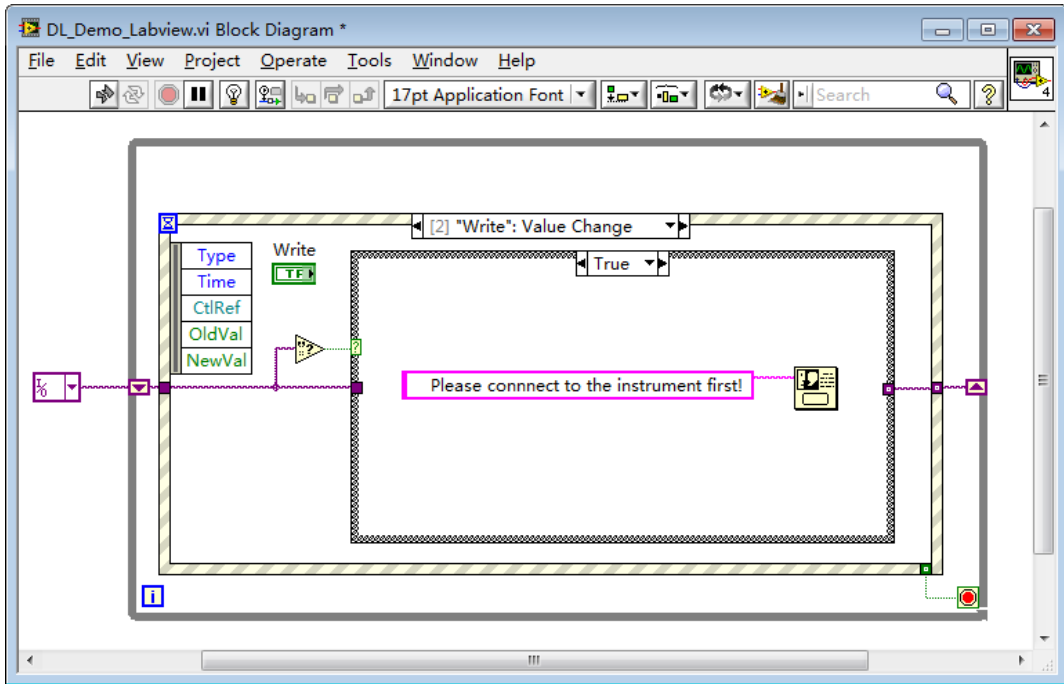
4. Add an event, including **Connect**, **Write**, **Read**, and **Exit**.

(1) Connect (including error correction advice):

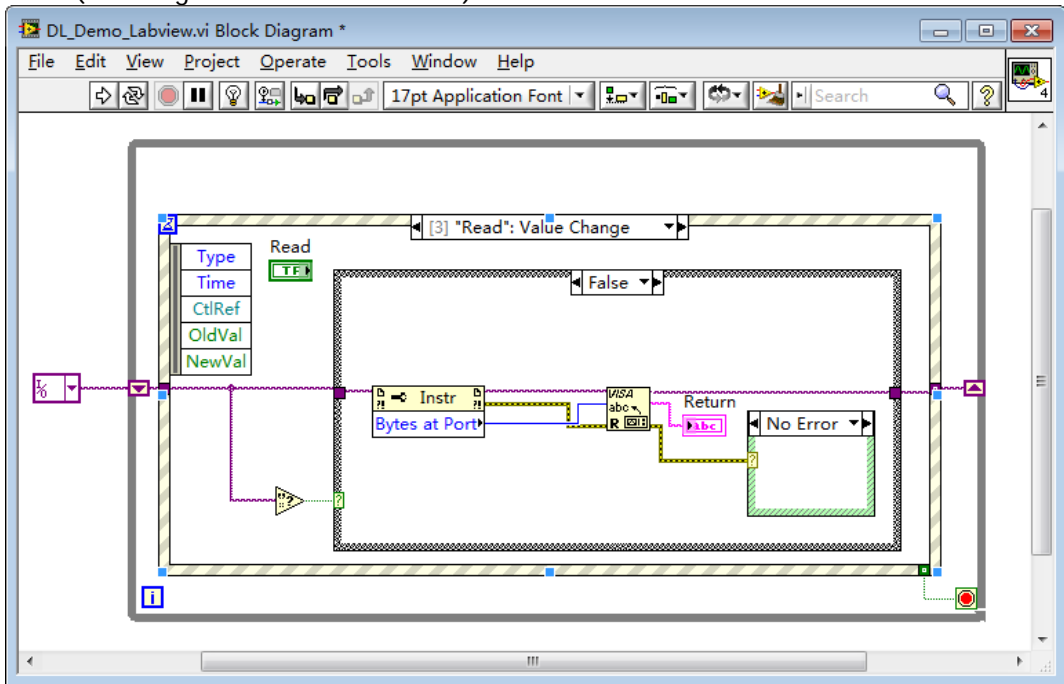


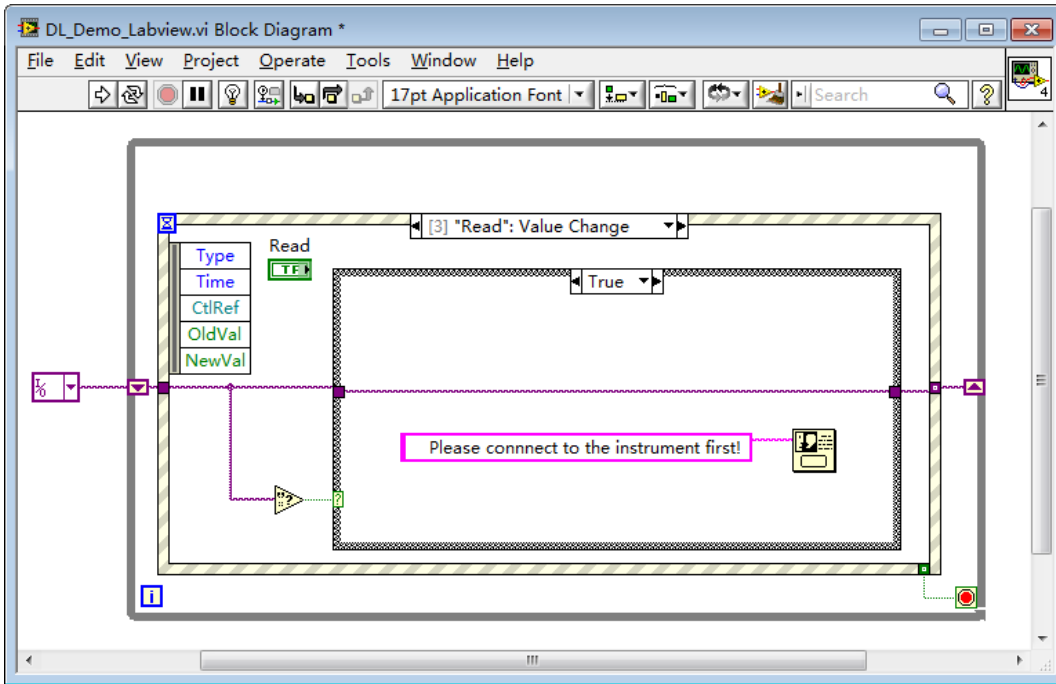
(2) Write (including error confirmation):



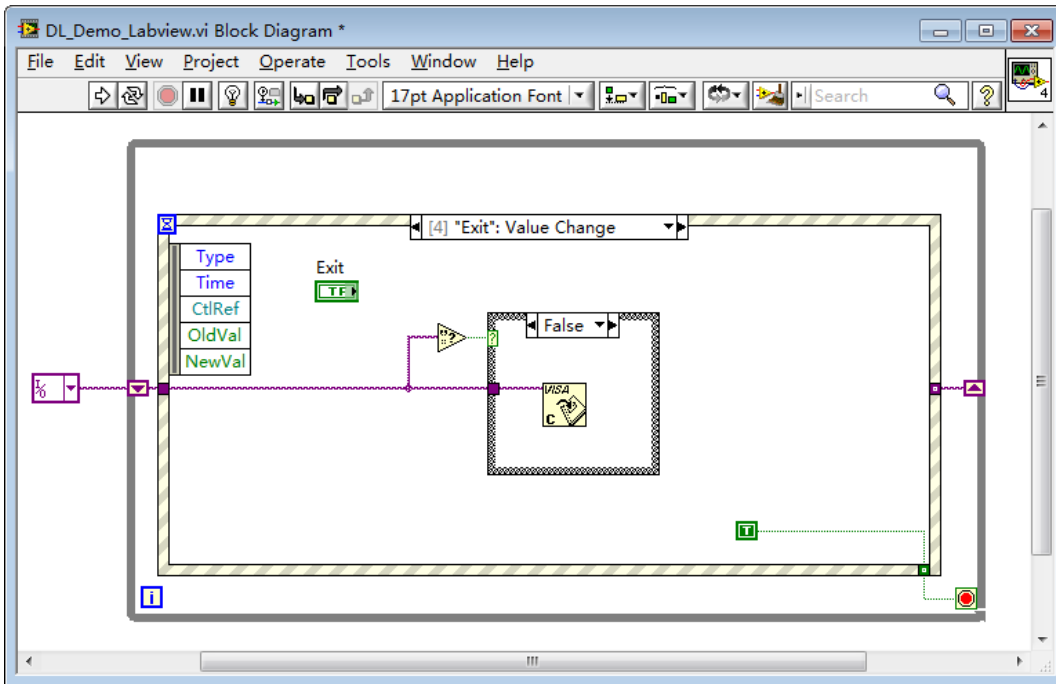


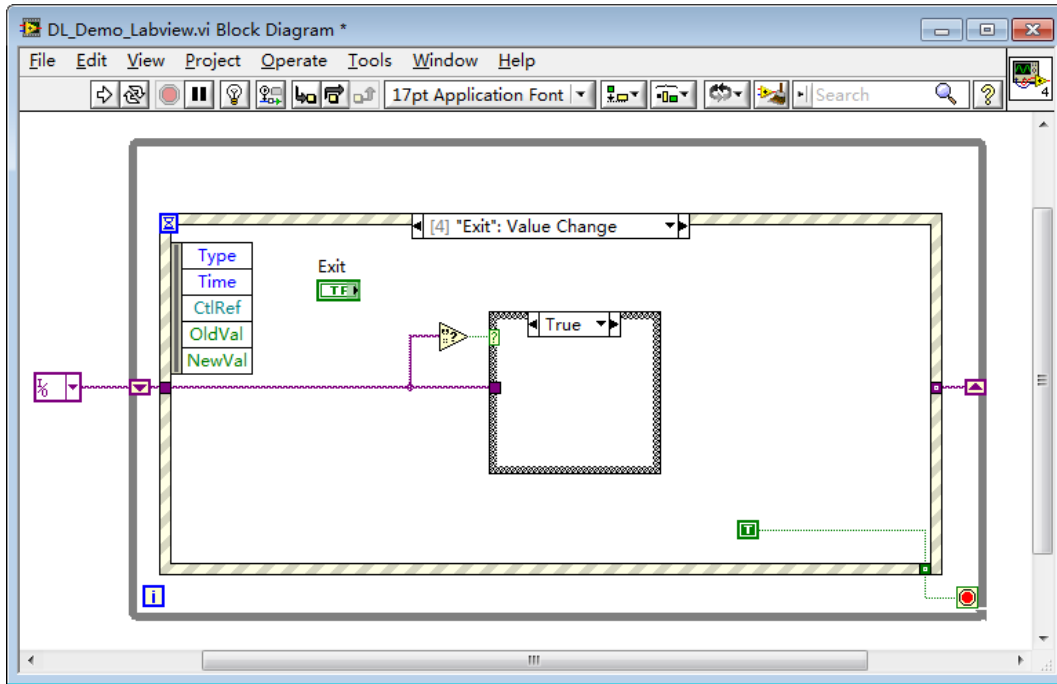
(3) Read (including error correction advice):



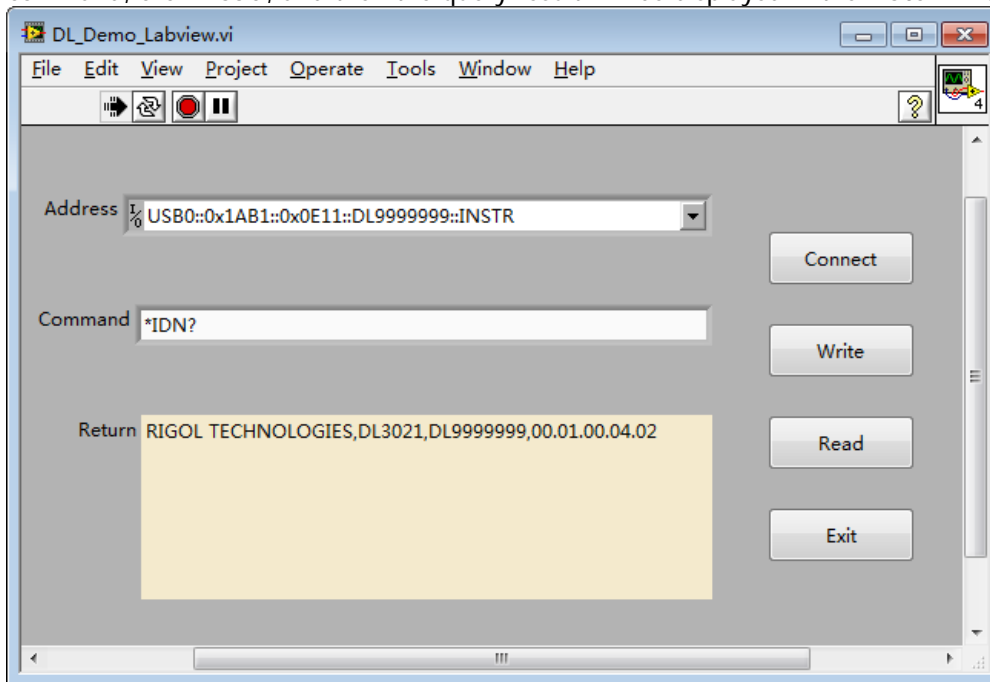


(4) Exit:





- Run the program, and then the following interface is displayed below. Click the VISA resource name from the drop-down list under **Address**, and click **Connect** to connect the instrument. Then, input *IDN? in the **Command** field. Click **Write** to write the command to the instrument. If it is a query command, click **Read**, and then the query result will be displayed in the **Return** field.

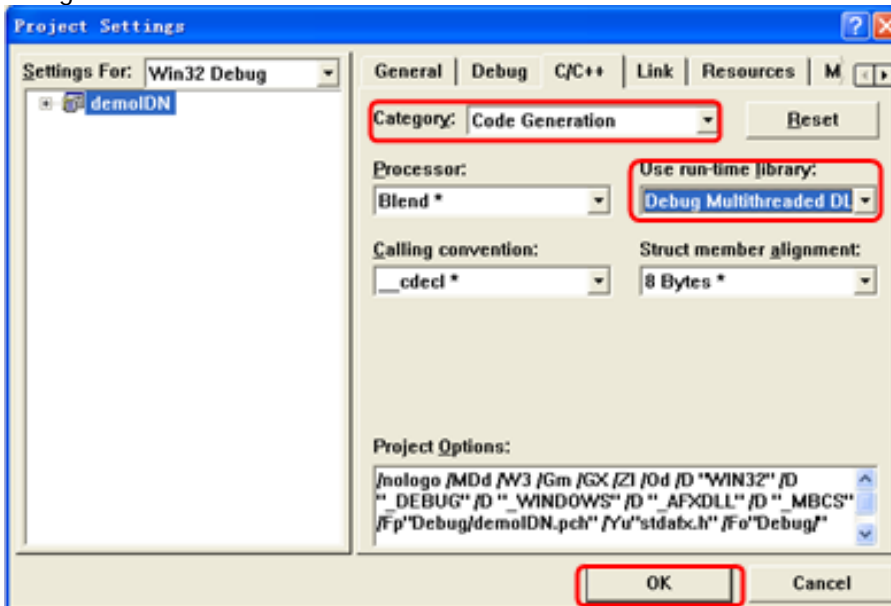


Visual C++ Programming Instance

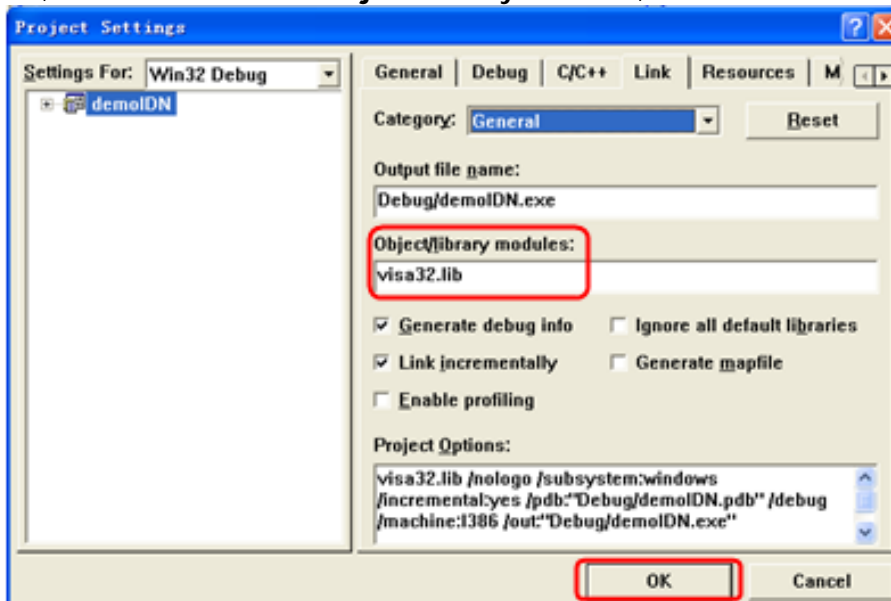
Program used in this example: Microsoft Visual C++6.0

Function realized in this example: sending the *IDN? command and reading the instrument information.

1. Run Microsoft Visual C++6.0. Create a MFC project based on a dialog box.
2. Click **Project→Settings** to open the "Project Settings" dialog box. In the dialog box, click the "C/C++" tab, select **Code Generation** from the drop-down list under **Category**. Choose **Debug Multithreaded DLL** from the drop-down list under **Use run-time library**. Click **OK** to close the dialog box.



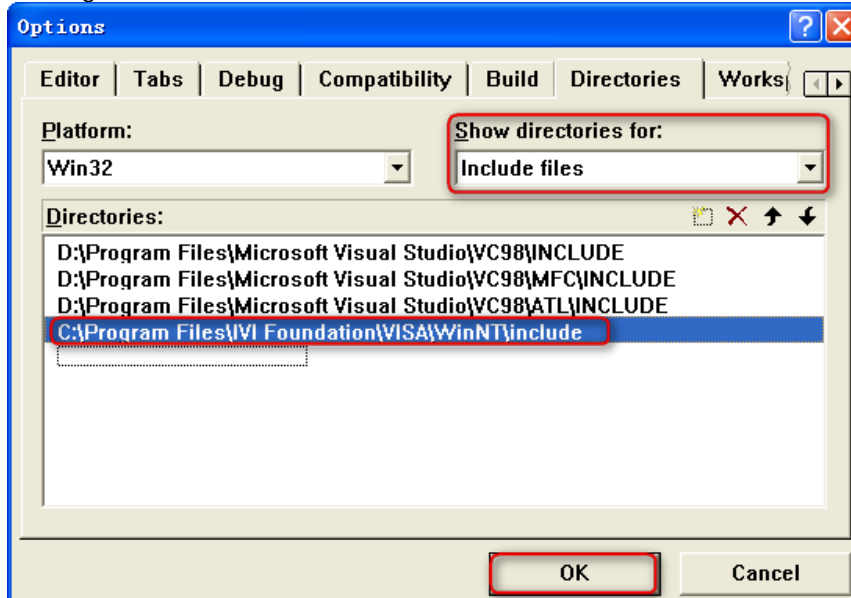
3. Click **Project→Settings** to open the "Project Settings" dialog box. In the dialog box, click the "Link" tab, add "visa32.lib" under **Object/library modules**, then click **OK** to close the dialog box.



4. Click **Tools→Options** to open the "Options" dialog box. Then, click the "Directories" tab.

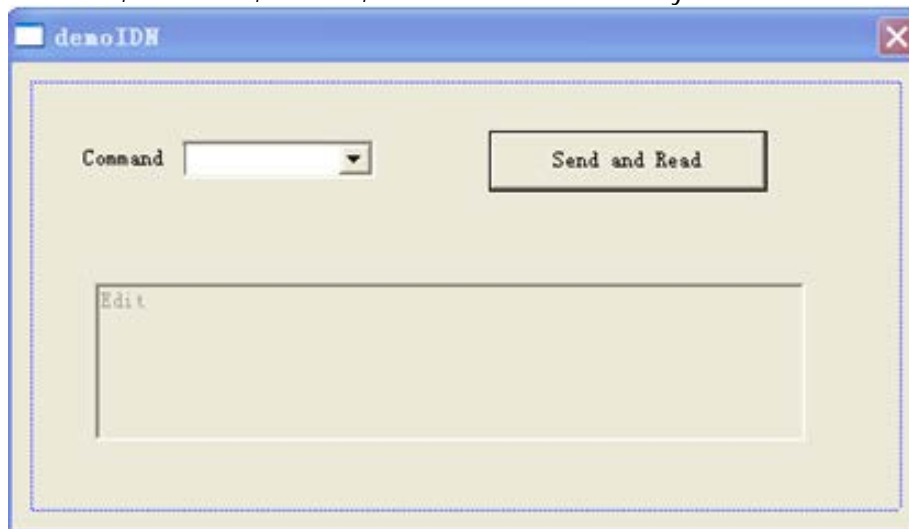
Select **Include files** from the drop-down list under **Show directories for**. Double click the empty space under **Directories** to enter the specified path of Include files: C:\Program Files\IVI Foundation\VISA\WinNT\include. Click **OK** to close the dialog box.

In the "Directories" tab, select **Library files** from the drop-down list under **Show directories for**. Double click the empty space under **Directories** to enter the specified path of Library files: C:\Program Files\IVI Foundation\VISA\WinNT\lib\msc. Click **OK** to close the dialog box.

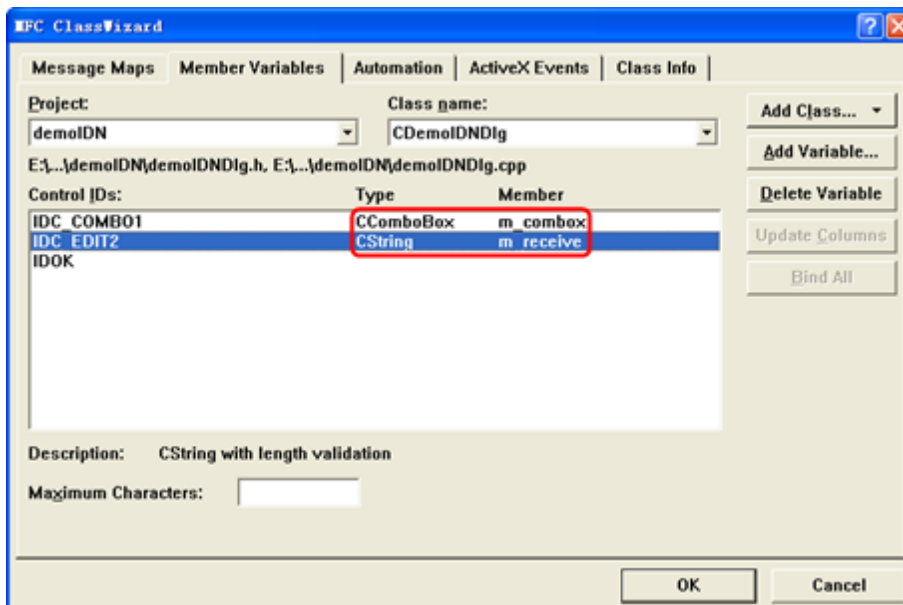
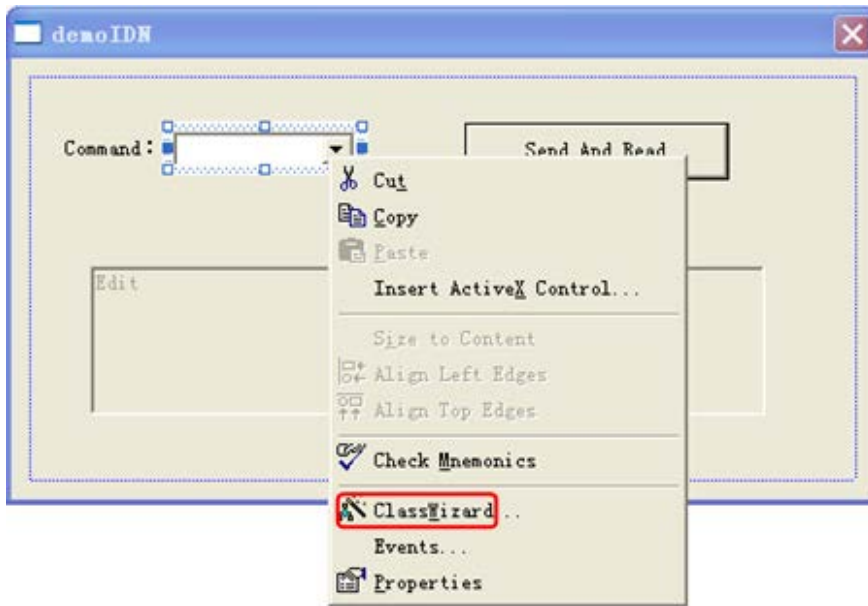


Note: By now, VISA library has been added.

5. Add **Text**, **Com box**, **Button**, and **Edit** controls. The layout interface is as follows:



6. Modify the control attributes.
 - 1) Name "Text" as "Command".
 - 2) Open the Data item in the Com box attribute and input the following command manually.
*IDN?
 - 3) Open the General item in the Edit attribute and select Disable.
 - 4) Name "Button" as "Send and Read".
7. Add the variables "m_combox" and "m_receive" for the Com box and Edit controls respectively.



8. Add codes.
 Double-click **Send and Read** to enter the programming environment. Declare the VISA library "#include <visa.h>" in the header file and then add the following codes:
- ```

ViSession defaultRM, vi;
charbuf [256] = {0};
CStrings, strTemp;
char* stringTemp;
ViChar buffer [VI_FIND_BUFLLEN];
ViRsrc matches=buffer;
ViUInt32 nmatches;
ViFindList list;
viOpenDefaultRM (&defaultRM);
//obtain the USB resource of VISA
viFindRsrc(defaultRM, "USB?*", &list,&nmatches, matches);
viOpen (defaultRM,matches,VI_NULL,VI_NULL,&vi);
viPrintf (vi, "*RST\n");
//send the received command
m_combox.GetLBText(m_combox.GetCurSel(),strTemp);

```

```
strTemp = strTemp + "\n";
stringTemp = (char *) (LPCTSTR)strTemp;
viPrintf (vi,stringTemp);
//read results
viScanf (vi, "%t\n", &buf);
//display the results
UpdateData (TRUE);
m_receive = buf;
UpdateData (FALSE);
viClose (vi);
viClose (defaultRM);
```

9. Save, compile, and run the project to obtain a single exe file. When the electronic load is successfully connected to the PC, select "\*IDN?" from the **Com box**. Press **Send and Read**, and the results of the load will be returned.

## Visual C# Programming Instance

**Program used in this instance:** Microsoft Visual Studio 2010

**Function realized in this example:** sending the \*IDN? command and reading the instrument information.

1. Run Microsoft Visual Studio 2010, and create a project. Search for visa32.cs from the VISA installation directory and add it to the project. Then, visa32.cs appears in the "solution manager", and other relevant files concerning visa32 are imported.
2. First open the resource manager, and then turn on the device (you need to set the descriptor when turning on the device).

```
Const string Cdevice_INSTR=@"USB0::0x1AB1::0x0E11::DL3000A000001::INSTR"
```

```
viError=visa32.viOpenDefaultRM(out viDefRm);
viError=visa32.viOpen(viDefRm,cDEVICE_INSTR,0,5000,out viSession);
```

3. Send requests and read data.

```
visa32.viPrintf(viSession,"*IDN?\n");
byteReadBuf = new byte[128];
visa32.viRead(viSession,byteReadBuf,128,out retCnt);
```

The data to be read is in byte form, you need to convert it into strings.

```
StrIdn="";
For(i=0;i<retCnt;i++)
{
 strIdn=strIdn+Convert.ToChar(byteReadBuf[i]);
}
```

4. Turn off the device and the resource manager after finishing the communication.

```
visa32.viClose(viSession);
visa32.viClose(viDefRm);
```

5. C# program

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace IDN
{
```

```
 class Program
 {
```

```
 const string cDEVICE_INSTR = @"USB0::0x1AB1::0x0E11::DL3000A000001::INSTR";
```

```
static void Main(string[] args)
```

```
{
```

```
 Int32 viError;
 Int32 viDefRm;
 Int32 viSession;
 Int32 retCnt;
 Int32 i;
 String strIdn;
 byte [] byteReadBuf;
```



```
viError = visa32.viOpenDefaultRM(out viDefRm);
 viError = visa32.viOpen(viDefRm, cDEVICE_INSTR, 0, 5000, out viSession);
visa32.viPrintf(viSession, "*IDN?\n");
byteReadBuf = new byte[128];
visa32.viRead(viSession, byteReadBuf, 128, out retCnt);
strIdn = "";
for (i = 0; i < retCnt; i++)
{
 strIdn = strIdn + Convert.ToChar(byteReadBuf[i]);
}
Console.WriteLine(strIdn);
Console.ReadKey();
visa32.viClose(viSession);
visa32.viClose(viDefRm);
}
}
}
```



## Chapter 5 Appendix

### Appendix A: Default Settings

**Note\*:** These parameters will not be affected by the default setting operations.

You can restore the instruments to factory defaults by doing either of the following operations:

- Send the **\*RST** command.
- Press **Utility** → **System** → **Power-on** to select "Default", and then when you power on the instrument the next time, it will restore to its factory default settings.

| Channel Parameter      |          |
|------------------------|----------|
| <b>CC</b>              |          |
| Current                | 0 A      |
| Range                  | 60 A     |
| Slew Rate              | 0.1 A/μs |
| Starting Voltage (Von) | 0 V      |
| V_Limit                | 180 V    |
| C_Limit                | 70 A     |
| <b>CV</b>              |          |
| Voltage                | 0 V      |
| Range                  | 150 V    |
| V_Limit                | 180 V    |
| C_Limit                | 70 A     |
| <b>CR</b>              |          |
| Resistance             | 2 Ω      |
| Range                  | 15 kΩ    |
| V_Limit                | 180 V    |
| C_Limit                | 70 A     |
| <b>CP</b>              |          |
| Power                  | 0 W      |
| V_Limit                | 180 V    |
| C_Limit                | 70 A     |
| <b>Con</b>             |          |
| Range                  | 60 A     |
| A_Level                | 4 A      |
| B_Level                | 1 A      |
| Rising slew rate       | 0.1 A/μs |
| Falling slew rate      | 0.1 A/μs |
| Period                 | 1 kHz    |
| Duty                   | 50%      |
| Trigger Source         | TRAN     |
| <b>Pul</b>             |          |
| Range                  | 60 A     |
| A_Level                | 4 A      |
| B_Level                | 1 A      |
| Rising slew rate       | 0.1 A/μs |
| Falling slew rate      | 0.1 A/μs |
| Width                  | 2 s      |

|                        |                |
|------------------------|----------------|
| Trigger Source         | TRAN           |
| <b>Tog</b>             |                |
| Range                  | 60 A           |
| A_Level                | 4 A            |
| B_Level                | 1 A            |
| Rising slew rate       | 0.1 A/ $\mu$ s |
| Falling slew rate      | 0.1 A/ $\mu$ s |
| Trigger Source         | TRAN           |
| <b>List</b>            |                |
| Mode                   | CC             |
| Range                  | 60 A           |
| Circle                 | 1              |
| Steps                  | 2              |
| Last State             | Off            |
| Trigger Source         | TRAN           |
| Step Value             | 2 A            |
| Step Duration          | 1 s            |
| Step Slew              | 0.1 A/ $\mu$ s |
| <b>Battery</b>         |                |
| Current                | 0 A            |
| Range                  | 60 A           |
| Cut-off Voltage        | OFF, 0 V       |
| Battery Capacity       | OFF, 0 mAh     |
| Discharge Time         | OFF, 0 s       |
| Starting Voltage (Von) | 0.5 V          |
| <b>OCP</b>             |                |
| Range                  | 60 A           |
| Starting Voltage (Von) | 0.01 V         |
| Delay Time             | 500 ms         |
| Original Current       | 0 A            |
| Step Current           | 1 A            |
| Step Delay             | 500 ms         |
| Protection Voltage     | 0.5 V          |
| Maximum Current        | 10 A           |
| Minimum Current        | 9 A            |
| Protection Time        | 500 $\mu$ s    |
| <b>OPP</b>             |                |
| Starting Voltage (Von) | 0.01 V         |
| Delay Time             | 500 ms         |
| Starting Power         | 0 W            |
| Step Power             | 1 W            |
| Step Delay             | 500 ms         |
| Protection Voltage     | 0.5 V          |
| Maximum Power          | 100 W          |
| Minimum Power          | 90 W           |
| Protection Time        | 500 $\mu$ s    |
| <b>Factory</b>         |                |

|               |       |
|---------------|-------|
| Mode          | CC    |
| Current       | 0 A   |
| Range         | 60 A  |
| Maximum Value | 10 A  |
| Minimum Value | 9 A   |
| Pass_Tip      | ON    |
| Fail_Tip      | OFF   |
| Pass_Sound    | ON    |
| Fail_Sound    | OFF   |
| <b>CC+CV</b>  |       |
| Current       | 0 A   |
| C_Range       | 60 A  |
| Voltage       | 0 V   |
| V_Range       | 150 V |
| V_Limit       | 180 V |
| C_Limit       | 70 A  |

| <b>System</b>  |         |
|----------------|---------|
| Language*      | 简体中文    |
| Power-on       | Default |
| Beeper         | ON      |
| Digital Input  | Disable |
| Digital Output | Disable |
| Brightness*    | 50%     |
| Sense          | OFF     |
| Log            | OFF     |
| Vmon_EXT       | OFF     |
| Imon_EXT       | OFF     |
| SHORT          | HOLD    |
| Von Latch      | ON      |

| <b>Communication Interface Setting</b> |      |
|----------------------------------------|------|
| GPIB address                           | 2    |
| <b>RS232*</b>                          |      |
| Baud Rate                              | 9600 |
| Data Bits                              | 8    |
| Stop Bits                              | 1    |
| Parity                                 | None |
| Flow                                   | OFF  |
| <b>LAN*</b>                            |      |
| DHCP                                   | ON   |
| AutoIP                                 | ON   |
| ManualIP                               | OFF  |

| <b>Store</b> |            |
|--------------|------------|
| Cursor       | Contents   |
| Contents     | C:\        |
| File         | First File |

## Appendix B: Warranty

**RIGOL** TECHNOLOGIES, INC. (hereinafter referred to as **RIGOL**) warrants that the product will be free from defects in materials and workmanship within the warranty period. If a product proves defective within the warranty period, **RIGOL** guarantees free replacement or repair for the defective product.

To get repair service, please contact with your nearest **RIGOL** sales or service office.

There is no other warranty, expressed or implied, except such as is expressly set forth herein or other applicable warranty card. There is no implied warranty of merchantability or fitness for a particular purpose. Under no circumstances shall **RIGOL** be liable for any consequential, indirect, ensuing, or special damages for any breach of warranty in any case.