RIGOL

Programming Guide

DP800 Series Programmable Linear DC Power Supply

Mar. 2013
RIGOL Technologies, Inc.

Guaranty and Declaration

Copyright

Trademark Information

Publication Number

PGH03102-1110

Notices

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

Product Certification

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

Contact Us

If you have any problem or requirement when using our products, please contact RIGOL Technologies, Inc. or your local distributors, or visit: www.rigol.com.

Safety Requirement

General Safety Summary

Please review the following safety precautions carefully before putting the instrument into operation so as to avoid any personal injuries or damages to the instrument and any product connected to it. To prevent potential hazards, please use the instrument only specified by this manual.

Use Proper Power Cord.
Only the power cord designed for the instrument and authorized by local country could be used.

Ground The Instrument.
The instrument is grounded through the Protective Earth lead of the power cord. To avoid electric shock, it is essential to connect the earth terminal of power cord to the Protective Earth terminal before any inputs or outputs.

Observe All Terminal Ratings.
To avoid fire or shock hazard, observe all ratings and markers on the instrument and check your manual for more information about ratings before connecting.

Use Proper Overvoltage Protection.
Make sure that no overvoltage (such as that caused by a thunderstorm) can reach the product, or else the operator might expose to danger of electrical shock.

Do Not Operate Without Covers.
Do not operate the instrument with covers or panels removed.

Use Proper Fuse.
Please use the specified fuses.

Avoid Circuit or Wire Exposure.
Do not touch exposed junctions and components when the unit is powered.

Do Not Operate With Suspected Failures.
If you suspect damage occurs to the instrument, have it inspected by qualified service personnel before further operations. Any maintenance, adjustment or replacement especially to circuits or accessories must be performed by **RIGOL** authorized personnel.

Keep Well Ventilation.
Inadequate ventilation may cause increasing of temperature or damages to the device. So please keep well ventilated and inspect the intake and fan regularly.

Do Not Operate in Wet Conditions.
In order to avoid short circuiting to the interior of the device or electric shock, please do not operate in a humid environment.

Do Not Operate in an Explosive Atmosphere.
In order to avoid damages to the device or personal injuries, it is important to operate the device away from an explosive atmosphere.

Keep Product Surfaces Clean and Dry.
To avoid the influence of dust and/or moisture in air, please keep the surface of device clean and dry.

Electrostatic Prevention.
Operate in an electrostatic discharge protective area environment to avoid damages induced by static

discharges. Always ground both the internal and external conductors of the cable to release static before connecting.

Handling Safety.

Please handle with care during transportation to avoid damages to buttons, knob interfaces and other parts on the panels.

Safety Terms and Symbols

Terms in this Manual. These terms may appear in this manual:

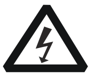
WARNING
Warning statements indicate the conditions or practices that could result in injury or loss of life.

CAUTION
Caution statements indicate the conditions or practices that could result in damage to this product or other property.

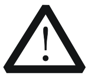
Terms on the Product. These terms may appear on the Product:

DANGER indicates an injury or hazard may immediately happen.
WARNING indicates an injury or hazard may be accessible potentially.
CAUTION indicates a potential damage to the instrument or other property might occur.

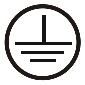
Symbols on the Product. These symbols may appear on the product:



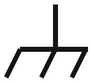
Hazardous Voltage



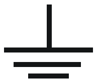
Safety Warning



Protective Earth Terminal



Chassis Ground



Test Ground

Document Overview

This manual introduces how to program the power supply over remote interfaces in details.

Main Topics in this Manual:

Chapter 1 Programming Overview
This chapter introduces how to build the remote communication between the power supply and PC and how to control the power supply remotely. Besides, it also provides a brief introduction of the SCPI commands.

Chapter 2 Command System
This chapter introduces the syntax, function, parameter and using instruction of each DP800 command in A-Z order.

Chapter 3 Application Examples
This chapter provides the application examples of the main functions of the power supply. In the application examples, a series of commands are combined to realize the basic functions of the power supply.

Chapter 4 Programming Demos
This chapter introduces how to program and control DP800 using various development tools, such as Visual C++, Visual Basic and LabVIEW.

Chapter 5 Appendix
This chapter provides various information, such as the command list and factory setting list.

> **Tip**
> For the newest version of this manual, please download it from www.rigol.com.

Format Conventions in this Manual:

1 **Button**
The function key at the front panel is denoted by the format of "Button Name (Bold) + Text Box" in the manual, for example, **Utility** denotes the "System Auxiliary Function Setting" key.

2 **Menu**
The menu item is denoted by the format of "Menu Word (Bold) + Character Shading" in the manual, for example, **SysInfo** denotes the "System Information" item under **Utility**.

3 **Operation Step**
The next step of the operation is denoted by an arrow "→" in the manual. For example, **Utility** → **System** denotes pressing **Utility** at the front panel and then pressing **System**.

Content Conventions in this Manual:

DP800 series programmable linear DC power supply includes the following models. The programmable ranges of the voltage and current for each model are as listed below. Unless otherwise noted, in this manual, DP831A is taken as an example to illustrate the using method of each DP800 series command.

DP831A				
Channel	Voltage Range	Voltage Default	Current Range	Current Default
CH1	0V to 8.4V	0V	0A to 5.3A	5A
CH2	0V to 32V	0V	0A to 2.1A	2A
CH3	-32V to 0V	0V	0A to 2.1A	2A
DP832A/DP832				
Channel	Voltage Range	Voltage Default	Current Range	Current Default
CH1	0V to 32V	0V	0A to 3.2A	3A
CH2	0V to 32V	0V	0A to 3.2A	3A
CH3	0V to 5.3V	0V	0A to 3.2A	3A

Contents

Chapter 1 Programming Overview

This chapter introduces how to build the remote communication between the PC and instrument and provides an overview of the syntax, symbol, parameter type and abbreviation rules of the SCPI commands.

Main topics of this chapter:

- To Build Remote Communication
- Remote Control Methods
- SCPI Command Overview

To Build Remote Communication

You can build the remote communication between DP800 and PC over USB, LAN, RS232 or GPIB (option, can be extended via the USB-GPIB interface converter) interface.

Note: the end mark of the command sent through RS232 interface is "\r\n".

Operation Steps:

1 **Install the Ultra Sigma common PC software**

Download the Ultra Sigma common PC software from www.rigol.com and install it according to the instructions.

2 **Connect the instrument and PC and configure the interface parameters of the instrument**

DP800 supports USB, LAN, RS232 and GPIB (option, can be extended via the USB-GPIB interface converter) communication interfaces, as shown in the figure below.

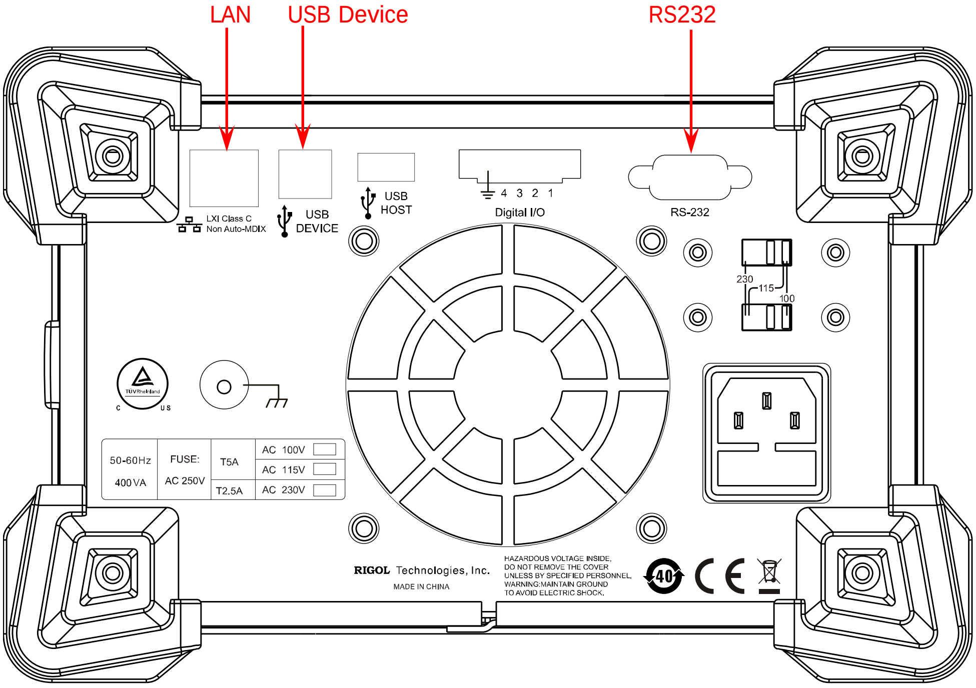


Figure 1-1 DP800 Communication Interfaces

(1) Use the USB interface: connect the USB Device interface at the rear panel of DP800 and the USB Host interface of the PC using USB cable.

(2) Use the LAN interface:

- Make sure that your PC is connected to the local network.
- Make sure whether your local network supports DHCP or auto IP mode. If not, you need to acquire the network interface parameters available, including the IP address, subnet mask, gateway and DNS.
- Connect DP800 to the local network using network cable.
- Press **Utility** → **I/O** → **LAN** to configure the IP address, subnet mask, gateway and DNS of the instrument.

(3) Use the RS232 interface:

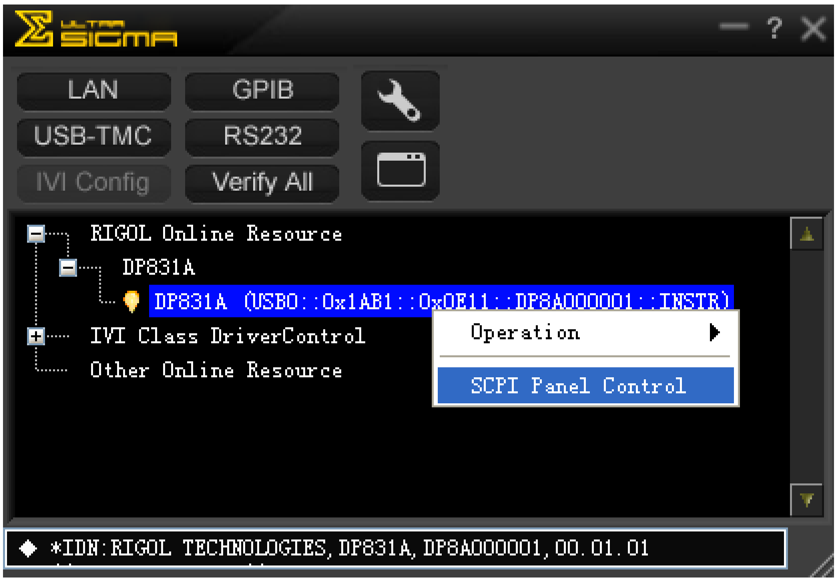
- Connect the RS232 interface with the PC or data terminal equipment (DTE) using RS232 cable.
- Press **Utility** → **I/O** → **RS232** to set interface parameters (baud rate, parity and etc) that match the PC or terminal equipment.

(4) Use the GPIB interface:

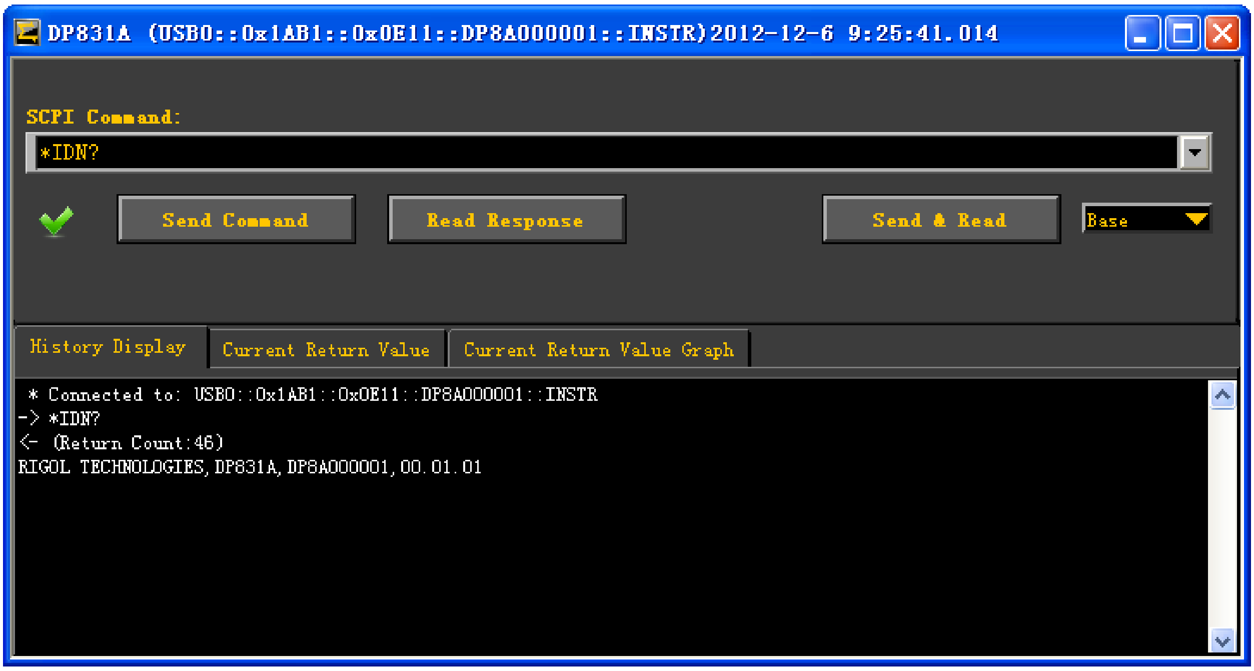
- Connect the USB-GPIB interface converter to the USB Device interface at the rear panel of DP800 to extend a GPIB interface.
- Connect the instrument with your PC using GPIB cable.
- Press **Utility** → **I/O** → **GPIB** to set the GPIB address of the instrument.

3 Check whether the connection is successful

Run the Ultra Sigma, search for resource, right-click the resource name and select “SCPI Panel Control” in the pop-up menu. Enter the correct command in the pop-up SCPI control panel and click **Send Command**, **Read Response** or **Send&Read** to check whether the connection is successful, as shown in the figure below (take the USB interface as an example).



```
*IDN:RIGOL TECHNOLOGIES,DP831A,DP8A000001,00.01.01
```


DP831A (USB0::0x1AB1::0x0E11::DP8A000001::INSTR)2012-12-6 9:25:41.014
SCPI Command:
Send Command
Read Response
Send & Read
Base
History Display
Current Return Value
Current Return Value Graph

SCPI Command: *IDN?

```
* Connected to: USB0::0x1AB1::0x0E11::DP8A000001::INSTR
-> *IDN?
<- (Return Count:46)
RIGOL TECHNOLOGIES,DP831A,DP8A000001,00.01.01
```

Remote Control Methods

1 **User-defined Programming**
You can program and control the instrument using the SCPI (Standard Commands for Programmable Instruments) commands listed in chapter 2 Command System in various development environments (such as Visual C++, Visual Basic and LabVIEW). For details, refer to the introductions in chapter 4 Programming Demos.

2 **Send SCPI Commands via the PC Software**
You can control the power supply remotely by sending SCPI commands via the PC software (Ultra Sigma) provided by **RIGOL**. Besides, you can also control the instrument using the “Measurement & Automation Explorer” of NI (National Instruments Corporation) or the “Agilent IO Libraries Suite” of Agilent (Agilent Technologies, Inc.).

SCPI Command Overview

SCPI (Standard Commands for Programmable Instruments) is a standardized instrument programming language that is built upon the standard IEEE488.1 and IEEE 488.2 and conforms to various standards (such as the floating point operation rule in IEEE754 standard, ISO646 7-bit coded character for information interchange (equivalent to ASCII programming)). This section introduces the syntax, symbols, parameters and abbreviation rules of the SCPI commands.

Syntax

SCPI commands present a hierarchical tree structure and contain multiple sub-systems, each of consists of a root keyword and one or more sub-keywords. The command string usually starts with ":"; the keywords are separated by ":" and are followed by the parameter settings available; "?" is added at the end of the command string to indicate query; the command and parameter are separated by "space".

For example,

```
:SYSTem:COMMunicate:LAN:IPADdress <ip>
:SYSTem:COMMunicate:LAN:IPADdress?
```

SYSTem is the root keyword of the command. COMMunicate, LAN and IPADdress are the second-level, third-level and forth-level keywords respectively. The command string starts with ":" which separates the multiple-level keywords. <ip> represents the parameters available for setting. "?" represents query and the power supply returns the response information (the output value or internal setting value of the instrument) when recieving a query command. The command :SYSTem:COMMunicate:LAN:IPADdress and parameter <ip> are separated by space.

"," is generally used for separating multiple parameters contained in the same command, for example,

```
:DELAY:PARAmeter <secnum>,{ON|OFF},<time>
```

Symbol Description

The following four symbols are not the content of SCPI commands and will not be sent with the commands. They are usually used to describe the parameters in the commands.

1 Braces { }

Usually, multiple optional parameters are enclosed in the braces and one of the parameters must be selected when sending the command. For example, :DISPlay:MODE {NORMal|WAVE|DIAL}.

2 Vertical Bar |

The vertical bar is used to separate multiple parameters and one of the parameters must be selected when sending the command. For example, :DISPlay:MODE {NORMal|WAVE|DIAL}.

3 Square Brackets [ ]

The content (command keyword) enclosed in the square brackets can be omitted. When the parameter is omitted, the instrument will set the parameter to its default. For example, for the :MEASure[:VOLTage][:DC]? command, sending any of the four commands below can achieve the same effect.

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

4 Triangle Brackets < >

The parameter enclosed in the triangle brackets must be replaced by an effective value. For example, send the :ANALyzer:CURRTime <value> command in :ANALyzer:CURRTime 5 form.

Parameter Type

The parameters of the commands introduced in this manual contains 5 types: bool, integer, real number, discrete and ASCII character string.

1 Bool

The parameter could be OFF or ON. For example, :RECorder[:STATe] {ON|OFF}.

2 Integer

Unless otherwise noted, the parameter can be any integer within the effective value range. Note that do not set the parameter to a decimal; otherwise, errors will occur. For example, in the :SYSTem:BRIGhtness <brightness> command, <brightness> can be any integer from 0 to 100.

3 Real Number

Unless otherwise noted, the parameter can be any real number within the effective value range. For example, for CH1 of DP831A, the ranges of <volt> and <curr> in the :APPLy {CH1|CH2|CH3},<volt>,<curr> command are 0 to 8.4V and 0 to 5.3A respectively.

4 Discrete

The parameter could only be one of specified values or characters. For example, in the :ANALyzer:OBJect {V|C|P} command, the parameter can be V, C or P.

5 ASCII Character String

The parameter should be the combinations of ASCII characters. For example, in the :MMEMory:STORe <file_name> command, <file_name> is the filename of the file to be saved and can include Chinese characters, English characters and numbers.

Besides, many commands contain the MINimum and MAXimum parameters which are used to set the parameter to its minimum or maximum value. For example, MINimum and MAXimum in the :SYSTem:BRIGhtness {<brightness>|MINimum|MAXimum} command are used to set the brightness to the minimum or maximum.

Command Abbreviation

All the commands are case-insensitive and you can use any of them. If abbreviation is used, all the capital letters in the command must be written completely. For example, the :ANALyzer:ANALyze command can be abbreviated to :ANAL:ANAL.

Chapter 2 Command System

This chapter introduces the syntax, function, parameter and using instruction of each DP800 command in A-Z order.

Main topics of this chapter:

- :ANALyzer Commands
- :APPLy Command
- :DELAY Commands
- :DISPlay Command
- IEEE488.2 Common Commands
- :INSTrument Commands
- :MEASure Commands
- :MEMory Commands
- :MMEMory Commands
- :MONItor Commands
- :OUTPut Commands
- :PRESet Commands
- :RECorder Commands
- :SOURce Commands
- :SYSTem Commands
- :TIMEr Commands
- :TRIGger Commands

Explanation: In this command system, setting commands relating to the time, voltage, current and power parameters can be sent with units. Unless otherwise noted, the units available and the default unit of each parameter are as shown in the table below.

Parameter Type	Units Available	Default Unit
Time	s[1]	s
Voltage	V, mV	V
Current	A, mA	A
Power	W, mW	W

Note[1]: For the :TRIGger:OUT:PERIod [D0|D1|D2|D3,]<value> command (setting the period of the square waveform of trigger output), <value> is a time parameter and the units available are s, ms and us. The default unit is s.

:ANALyzer Commands

The :ANALyzer commands are used to set the analyzer parameters, execute analysis and query the analysis results.

Command List[1]:

- :ANALyzer:ANALyze
- :ANALyzer:ENDTime
- :ANALyzer:FILE?
- :ANALyzer:MEMory
- :ANALyzer:MMEMory
- :ANALyzer:OBJect
- :ANALyzer:RESult?
- :ANALyzer:STARTTime
- :ANALyzer:VALue?

:ANALyzer:ANALyze

Syntax :ANALyzer:ANALyze

Description When receiving this command, the instrument executes the analysis operation according to the current setting.

Explanation You can send the :ANALyzer:RESult? command to view the analysis results.

Related Command :ANALyzer:RESult?

[1] **Note[1]:** In the "Command List" in this manual, the parameters in the setting commands and the query commands are not included and you can refer to the complete introductions of the commands in the text according to the keyword.

:ANALyzer:ENDTime

Syntax :ANALyzer:ENDTime {<value>|MINimum|MAXimum}

:ANALyzer:ENDTime? [MINimum|MAXimum]

Description Set the end time of the analyzer.

Query the end time of the analyzer.

Parameter

Name	Type	Range	Default
<value>	Integer	Start time to the current maximum record time	2

Explanation

- You can only set the end time when valid record file is opened (refer to the :ANALyzer:FILE? command).
- When receiving the :ANALyzer:ANALyze command, the analyzer will analyze the recorded data between the start time and end time.

Return Format The query returns an integer, for example, 125.

Example

```
:ANAL:ENDT 125      /*Set the end time to 125s*/
:ANAL:ENDT?         /*Query the current end time and the query returns 125*/
```

Related Commands :ANALyzer:ANALyze

:ANALyzer:FILE?

:ANALyzer:STARTTime

:ANALyzer:FILE?

Syntax :ANALyzer:FILE?

Description Query the record file currently opened.

Return Format The query returns the directory of the file currently opened, for example, C:\REC 10:test.ROF.

:ANALyzer:MEMory

Syntax :ANALyzer:MEMory {1|2|3|4|5|6|7|8|9|10}

Description Open the specified record file in the internal memory.

Parameter

Name	Type	Range	Default
{1|2|3|4|5|6|7|8|9|10}	Discrete	1|2|3|4|5|6|7|8|9|10	None

Explanation This command is only available when valid record file is stored in the specified location.

Example

```
:ANAL:MEMory 10      /*Open the record file currently stored in record file storage
                       location 10 in C disk*/
```

:ANALyzer:MMEMory

Syntax :ANALyzer:MMEMory <dest>

Description Open the specified record file.

Parameter

Name	Type	Range	Default
<dest>	ASCII character string	Valid directory under D disk	None

Explanation This command is only available when valid record file is stored in the specified location.

Example :ANAL:MMEMory D:\record.ROF /*Open the record.ROF file under D disk*/

:ANALyzer:OBJect

Syntax :ANALyzer:OBJect {V|C|P}

:ANALyzer:OBJect?

Description Set the analysis object of the analyzer to voltage, current or power.

Query the analysis object of the analyzer.

Parameter

Name	Type	Range	Default
{V|C|P}	Discrete	V|C|P	V

Explanation You can only set the analysis object when valid record file is opened (refer to the :ANALyzer:FILE? command).

Return Format The query returns V, C or P.

Example :ANAL:OBJ V /*Set the analysis object of the analyzer to voltage*/

:ANAL:OBJ? /*Query the analysis object of the analyzer and the query returns V*/

Related Command :ANALyzer:FILE?

:ANALyzer:RESult?

Syntax :ANALyzer:RESult?

Description Query the analysis results, including the number of groups, median, mode, average, variance, range, min, max and mean deviation.

Return Format The query returns the analysis results with the data separated by commas, for example,
Group:1029,Median:0.0155V,Mode:0.0155V,Average:0.0154V,Variance:0.0000V,
Range:0.0005V,Min:0.0152V,Max:0.0157V,Mean:0.0000V.

Example :ANAL:RES? /*Query the analysis results*/

:ANALyzer:STARTTime

Syntax :ANALyzer:STARTTime {<value>|MINimum|MAXimum}

:ANALyzer:STARTTime? [MINimum|MAXimum]

Description Set the start time of the analyzer.

Query the start time of the analyzer.

Parameter

Name	Type	Range	Default
<value>	Integer	1s to end time	1

Explanation

- You can only set the start time when valid record file is opened (refer to the :ANALyzer:FILE? command).
- Send the :ANALyzer:ANALyze command and the analyzer analyzes the recorded data between the start time and end time.

Return Format The query returns an integer, for example, 1.

Example

```
:ANAL:STARTT 1      /*Set the start time to 1s*/
:ANAL:STARTT?      /*Query the current start time and the query returns 1*/
```

Related Commands :ANALyzer:ANALyze

:ANALyzer:FILE?

:ANALyzer:ENDTime

:ANALyzer:VALue?

Syntax :ANALyzer:VALue? <time>

Description Query the voltage, current and power at the specified time in the record file opened.

Parameter

Name	Type	Range	Default
<time>	Integer	Start time to end time	None

Return Format The query returns the voltage, current and power separated by commas, for example, Volt:1.2817V,Curr:0.0485A,Power:0.0622W.

Example

```
:ANAL:VAL? 5   /*Query the voltage, current and power at 5s and the query returns
                 Volt:1.2817V,Curr:0.0485A,Power:0.0622W*/
```

Related Commands :ANALyzer:ENDTime

:ANALyzer:FILE?

:ANALyzer:STARTTime

:APPLy Command

The :APPLy command provides the most straightforward method to program the power supply over the remote interface.

Syntax

```
:APPLy {CH1|CH2|CH3}
[,<volt>|MINimum|MAXimum][,<curr>|MINimum|MAXimum]
```

```
:APPLy? {CH1|CH2|CH3}[,CURRent|VOLTage]
```

Description Set the voltage/current of the specified channel.

Query the voltage/current of the specified channel.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	None
<volt>	Real	Refer to the "**Explanation**"	
<curr>	Real	Refer to the "**Explanation**"	

Explanation

- <volt> and <curr> can be omitted. If both of them are omitted, the command will select the desired channel; if only one of them is omitted, the command will set the voltage of the specified channel.
- The parameter ranges and defaults of each channel for different models are as listed in the table below.

Channel	<volt> Range	<volt> Default	<curr> Range	<curr> Default
CH1	0V to 8.4V	0V	0A to 5.3A	5A
CH2	0V to 32V	0V	0A to 2.1A	2A
CH3	-32V to 0V	0V	0A to 2.1A	2A
Channel	<volt> Range	<volt> Default	<curr> Range	<curr> Default
CH1	0V to 32V	0V	0A to 3.2A	3A
CH2	0V to 32V	0V	0A to 3.2A	3A
CH3	0V to 5.3V	0V	0A to 3.2A	3A

- If [,CURRent|VOLTage] is omitted in the query command, the command queries the voltage and current of the specified channel.

Return Format If [,CURRent|VOLTage] is not omitted, the query returns the voltage or current, for example, 5.000; if the parameter is omitted, the query returns the label, voltage and current of the specified channel (separated by commas), for example, CH1,5.000,1.0000.

Example

```
:APPL CH1,5,1      /*Set the voltage and current of CH1*/
:APPL? CH1         /*Query the voltage and current of CH1 and the query returns
                     CH1,5.000,1.0000*/
```

:DELAY Commands

The :DELAY commands are used to set the delayer parameters (the number of groups, number of cycles, end state and etc) as well as enable or disable the delayer.

Command List:

- :DELAY:CYCLEs
- :DELAY:ENDState
- :DELAY:GROUPs
- :DELAY:PARAmeter
- :DELAY[:STATe]
- :DELAY:STATe:GEN
- :DELAY:STOP
- :DELAY:TIME:GEN

:DELAY:CYCLEs

Syntax :DELAY:CYCLEs {N|I}[,<value>]

:DELAY:CYCLEs?

Description Set the number of cycles of the delayer.

Query the number of cycles of the delayer.

Parameter

Name	Type	Range	Default
{N|I}	Discrete	N|I	N
<value>	Integer	1 to 99999	1

Explanation

- The number of cycles refers to the number of times that the instrument performs delay output according to the preset state. Wherein, I represents infinite number of cycles; N represents finite number of cycles, the number of cycles is specified by <value> and when this parameter is omitted, the number of cycles is set to 1 by default.
- The total number of groups in each delay output = the number of groups × the number of cycles; wherein, the number of groups is set by the :DELAY:GROUPs command.
- The power supply will terminate the delayer function when the total number of groups of delays is finished. At this point, the state of the power supply depends on the setting of the :DELAY:ENDState command.

Return Format The query returns I or N,<value>, for example, N,100.

Example

```
:DELAY:CYCLE I          /*Set the number of cycles to "Infinite" */
:DELAY:CYCLE N          /*Set the number of cycles to 1*/
:DELAY:CYCLE N,100      /*Set the number of cycles to 100*/
:DELAY:CYCLE?   /*Query the current number of cycles and the query returns
                  N,100*/
```

Related Commands :DELAY:ENDState

:DELAY:GROUPs

:DELAY:ENDState

Syntax :DELAY:ENDState {ON|OFF|LAST}

:DELAY:ENDState?

Description Set the end state of the delayer to On, Off or Last.

Query the end state of the delayer.

Parameter

Name	Type	Range	Default
{ON|OFF|LAST}	Discrete	ON|OFF|LAST	OFF

Explanation

- The end state refers to the state of the instrument when the delayer stops.
- ON: output on, the instrument turns on the output automatically; OFF: output off, the instrument turns off the output automatically; LAST: last state, the instrument stops at the output state of the last group.
- The total number of groups in each delay output = the number of groups × the number of cycles. Wherein, the number of groups is set by the :DELAY:GROUPs command and the number of cycles is set by the :DELAY:CYCLEs command.

Return Format The query returns ON, OFF or LAST.

Example

```
:DELAY:ENDS LAST      /*Set the end state to Last*/
:DELAY:ENDS?          /*Query the current end state and the query returns LAST */
```

Related Commands :DELAY:GROUPs

:DELAY:CYCLEs

:DELAY:GROUPs

Syntax :DELAY:GROUPs <value>

:DELAY:GROUPs?

Description Set the number of output groups of the delayer.

Query the number of output groups of the delayer.

Parameter

Name	Type	Range	Default
<value>	Integer	1 to 2048	1

Explanation

- The number of output groups refers to the number of times that the instrument turns on or off the output according to the preset state.
- The total number of groups in each delay output = the number of groups × the number of cycles. Wherein, the number of cycles is set by the :DELAY:CYCLEs command.
- The power supply will terminate the delayer function when the total number of groups of delays is finished. At this point, the state of the power supply depends on the setting of the :DELAY:ENDState command.

Return Format The query returns an integer from 1 to 2048.

Example

```
:DELAY:GROUP 125      /*Set the number of groups to 125*/
:DELAY:GROUP?         /*Query the current number of groups and the query
                        returns 125*/
```

Related Commands :DELAY:CYCLEs

:DELAY:ENDState

:DELAY:PARAmeter

Syntax :DELAY:PARAmeter <secnum>,{ON|OFF},<time>

:DELAY:PARAmeter? <firnum>,<timercount>

Description Set the delayer parameters of the specified groups.

Query the delayer parameters of the specified groups.

Parameter

Name	Type	Range	Default
<secnum>	Integer	0 to 2047	None
{ON|OFF}	Bool	ON|OFF	OFF (even group); ON (odd group)
<time>	Integer	1 to 99999	1s
<firnum>	Integer	0 to 2047	None
<timercount>	Integer	1 to 2048	None

Explanation

- <secnum> is the group number of the delayer parameters; {ON|OFF} is the output state; <time> is the delay time.
- <firnum> is the group number of the first group of delayer parameters to be queried; <timercount> is the total number of groups of delayer parameters to be queried.

Return Format For example, **#90000**00015**1,ON,1;2,OFF,1**; wherein, **#90000** is the data block header; 00015 is the number of bytes followed; **1,ON,1;2,OFF,1;** are the delayer parameters returned. The format of each group of parameters is “number,output state,delay time”, multiple groups of return values are separated by semicolons and parameters of the same group are separated by commas.

Example

```
:DELAY:PARA 1,ON,2     /*Set the state of the first group to ON and the time to 2s*/
:DELAY:PARA? 3,2       /*Query two groups of delayer parameters starting from the
                         third group*/
                      /*The query returns #90000000153,ON,1;4,OFF,1;*/
```

:DELAY[:STATe]

Syntax :DELAY[:STATe] {ON|OFF}

:DELAY[:STATe]?

Description Enable or disable the delay output function.

Query the state of the delay output function.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	OFF

Explanation

- Enabling the delayer will change the output state of the channel. Make sure that the devices connected to the power supply will not be affected before enabling the delayer.
- The delayer parameters cannot be modified when the delayer is enabled.

Return Format The query returns ON or OFF.

Example

```
:DELAY ON     /*Enable the delay output*/
:DELAY?       /*Query the status of the delay output and the query returns ON*/
```

:DELAY:STATe:GEN

Syntax :DELAY:STATe:GEN {01P|10P}

:DELAY:STATe:GEN?

Description Select the pattern used when generating state automatically.

Query the pattern used when generating state automatically.

Parameter

Name	Type	Range	Default
{01P|10P}	Discrete	01P|10P	01P

Explanation

- 01P: 0 1 pattern. The state is set to “Off” and “On” alternately.
- 10P: 1 0 pattern. The state is set to “On” and “Off” alternately.

Return Format The query returns 01P or 10P.

Example

```
:DELAY:STAT:GEN 10P   /*Select 1 0 pattern*/
:DELAY:STAT:GEN?      /*Query the pattern used when generating state
                        automatically and the query returns 10P */
```

:DELAY:STOP

Syntax :DELAY:STOP {NONE|<V|>V|<C|>C|<P|>P}[,<value>|MINimum|MAXimum]

:DELAY:STOP? [MINimum|MAXimum]

Description Set the stop condition of the delayer.

Query the stop condition of the delayer.

Parameter

Name	Type	Range	Default
{NONE|<V|>V|<C|>C|<P|>P}	Discrete	NONE|<V|>V|<C|>C|<P|>P	NONE
<value>	Real	0 to the maximum voltage/current/power of the current channel	

Explanation

- This command sets a stop condition. The power supply monitors the output voltage, current and power during delay output and stops the delay output when state that fulfills this condition is detected.
- {NONE|<V|>V|<C|>C|<P|>P} can set the stop condition to “None”, “<Volt”, “>Volt”, “<Curr”, “>Curr”, “<Power” or “>Power”. <value> is used to set the voltage, current or power of the stop condition and when it is omitted, the corresponding value will be set to 0.

Return Format The query returns NONE or “stop condition,value”, for example, >V,8.000.

Example

```
:DELAY:STOP >V,8     /*Set the stop condition to “>Volt” and the voltage to 8V*/
:DELAY:STOP?         /*Query the current stop condition and the query
                       returns >V,8.000*/
```

:DELAY:TIME:GEN

Syntax

```
:DELAY:TIME:GEN {FIX|INC|DEC}[,<value0>[,<value1>]]
:DELAY:TIME:GEN?
```

Description

Set the method used to generate time automatically as well as the corresponding on/off delay time or the time base value and step value.

Query the method used to generate time automatically as well as the corresponding parameters.

Parameter

Name	Type	Range	Default
{FIX|INC|DEC}	Discrete	FIX|INC|DEC	FIX
<value0>	Integer	Refer to the "**Explanation**"	1s
<value1>	Integer	Refer to the "**Explanation**"	1s

Explanation

- When FIX (fixed time) is selected, <value0> and <value1> are used to set the on/off delay time and the range is from 1s to 99999s. When both of the parameters are omitted, the on/off delay time will be set to 1s; when only one of the parameters is omitted, the on delay time will be set.
- When INC (monotonic increase) or DEC (monotonic decline) is selected, <value0> and <value1> are used to set the time base value and step value. The time increases or declines gradually from the time base value at the specified step to generate time. The two fulfills the relation: time base value + number of output groups*step value≤99999s. When both of the two parameters are omitted, the time base value and step value will both be set to 1s; when only one of the parameters is omitted, the time base value will be set.

Return Format

Fix: the query returns FIX,<value0>,<value1>, for example, FIX,1,2;

INC: the query returns INC,<value0>,<value1>, for example, INC,2,5;

DEC: the query returns DEC,<value0>,<value1>, for example, DEC,200,5.

Example

```
:DELAY:TIME:GEN INC,2,5    /*Monotonic increase, the time base value is 2s and
                             the step is 5s*/
:DELAY:TIME:GEN?           /*Query the method used to generate time automatically
                             and the parameters; the query returns INC,2,5*/
```

:DISPlay Command

Syntax

```
:DISPlay:MODE {NORMal|WAVE|DIAL}
:DISPlay:MODE?
```

Description Set the display mode to normal, wave or dial.

Query the current display mode.

Parameter

Name	Type	Range	Default
{NORMal|WAVE|DIAL}	Discrete	NORMal|WAVE|DIAL	NORMal

Explanation

- NORMal: normal mode. The parameters (such as the voltage and current) of all the channels are displayed in number format.
- WAVE: waveform mode. The parameters (such as the voltage and current) of the channel currently selected are displayed in both waveform and number formats.
- DIAL: dial mode. The parameters (such as the voltage and current) of the channel currently selected are displayed in both dial and number formats.

Return Format The query returns NORMAL, WAVE or DIAL.

Example

```
:DISP:MODE WAVE      /*Select the waveform display mode*/
:DISP:MODE?      /*Query the current display mode and the query returns WAVE */
```

IEEE488.2 Common Commands

Command List:

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

*IDN?

Syntax *IDN?

Description Query the ID character string of the instrument.

Return Format The query returns the ID character string of the instrument, for example, RIGOL TECHNOLOGIES,DP831A,DP8A000001,00.01.01.

*RCL

Syntax *RCL {1|2|3|4|5|6|7|8|9|10}

Description Read the instrument state stored.

Parameter

Name	Type	Range	Default
{1|2|3|4|5|6|7|8|9|10}	Discrete	1|2|3|4|5|6|7|8|9|10	None

Explanation

- The power supply provides 10 storage locations (numbered 1 to 10) for instrument states. This command reads the instrument state stored in the specified location.
- This command is only available when the specified storage location contains a state file.

Related Command :MEMory[:STATe]:LOAD

*RST

Syntax *RST

Description Restore the power supply to factory state (refer to Appendix B: Factory Setting) and clear the error queue.

Related Commands :PRESet:KEY

:PRESet[:APPLy]

*SAV

Syntax *SAV {1|2|3|4|5|6|7|8|9|10}

Description Save the current system state to the specified storage location using the default name. The default name is RIGOLn.RSF; n corresponds to the number of the storage location.

Parameter

Name	Type	Range	Default
{1|2|3|4|5|6|7|8|9|10}	Discrete	1|2|3|4|5|6|7|8|9|10	None

Explanation

- The power supply provides 10 storage locations (numbered 1 to 10) for instrument states.
- If the specified storage location already contains a state file, this command will directly overwrite the original file. If the state file stored in the specified storage location is locked (refer to the :MEMory[:STATe]:LOCK command), this command will not overwrite the original file and the storage operation is invalid.

Example

```
*SAV 5   /*Save the current instrument state to storage location 5 with the
          filename RIGOL5.RSF*/
```

Related Command :MEMory[:STATe]:STORe

*TST?

Syntax *TST?

Description Query the self-test results of the instrument.

Explanation The power supply executes self-test at start-up. This command queries the self-test results (including TopBoard, BottomBoard and fan).

Return Format The query returns the self-test results of TopBoard, BottomBoard and fan respectively, for example, TopBoard:PASS,BottomBoardPASS,Fan:PASS.

Related Commands :SYSTem:SELF:TEST:BOARD?

:SYSTem:SELF:TEST:FAN?

:INSTrument Commands

Command List:

- :INSTrument:NSELect
- :INSTrument[:SELEct]

:INSTrument:NSELect

Syntax :INSTrument:NSELect {1|2|3}

:INSTrument:NSELect?

Description Select the desired channel.

Query the channel currently selected.

Parameter

Name	Type	Range	Default
{1|2|3}	Discrete	1|2|3	1

Explanation In this command, numbers are used in place of the channel labels in the :INSTrument[:SELEct] command.

Return Format The query returns 1, 2 or 3.

Example :INST:NSEL 3 /*Select CH3*/

:INST:NSEL? /*Query the channel currently selected and the query returns 3*/

Related Command :INSTrument[:SELEct]

:INSTrument[:SELEct]

Syntax :INSTrument[:SELEct] {CH1|CH2|CH3}

:INSTrument[:SELEct]?

Description Select the desired channel.

Query the channel currently selected.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	CH1

Return Format The query returns CH1:8V/5A, CH2:30V/2A, CH3:-30V/2A (DP831A) or CH1:30V/3A, CH2:30V/3A, CH3:5V/3A (DP832A and DP832).

Example :INST CH3 /*Select CH3*/

:INST? /*Query the channel currently selected and the query returns CH3:-30V/2A */

Related Command :INSTrument:NSELect

:MEASure Commands

Command List:

- :MEASure:ALL[:DC]?
- :MEASure:CURRent[:DC]?
- :MEASure:POWEr[:DC]?
- :MEASure[:VOLTage][:DC]?

:MEASure:ALL[:DC]?

Syntax :MEASure:ALL[:DC]? [{CH1|CH2|CH3}]

Description Query the voltage, current and power measured internally on the specified channel.

Parameter

Name	Type	Range	Default
[{CH1|CH2|CH3}]	Discrete	CH1|CH2|CH3	None

Explanation

- If the parameter is omitted, the command queries the voltage, current and power of the channel currently selected.
- You can send the :MEASure:CURRent[:DC]?, :MEASure:POWEr[:DC]? and :MEASure[:VOLTage][:DC]? commands to query the current, power and voltage respectively.

Return Format The query returns the voltage, current and power separated by commas, for example, 5.0000,1.0000,5.000.

Example :MEAS:ALL? CH1 /*Query the voltage, current and power of CH1 and the query returns 5.0000,1.0000,5.000*/

Related Commands :MEASure:CURRent[:DC]?
:MEASure:POWEr[:DC]?
:MEASure[:VOLTage][:DC]?

:MEASure:CURRent[:DC]?

Syntax :MEASure:CURRent[:DC]? [{CH1|CH2|CH3}]

Description Query the output current measured internally on the specified channel.

Parameter

Name	Type	Range	Default
[{CH1|CH2|CH3}]	Discrete	CH1|CH2|CH3	None

Explanation

- If the parameter is omitted, the command queries the current of the channel currently selected.
- You can send the :MEASure:POWEr[:DC]? and :MEASure[:VOLTage][:DC]? commands to query the power and voltage respectively, or send the :MEASure:ALL[:DC]? command to query the voltage, current and power at the same time.

Return Format The query returns the current, for example, 1.0000.

Example :MEAS:CURR? CH1 /*Query the current of CH1 and the query returns 1.0000*/

Related Commands :MEASure:ALL[:DC]?
:MEASure:POWEr[:DC]?
:MEASure[:VOLTage][:DC]?

:MEASure:POWEr[:DC]?

Syntax :MEASure:POWEr[:DC]? [{CH1|CH2|CH3}]

Description Query the output power measured internally on the specified channel.

Parameter

Name	Type	Range	Default
[{CH1|CH2|CH3}]	Discrete	CH1|CH2|CH3	None

Explanation

- If the parameter is omitted, the command queries the power of the channel currently selected.
- You can send the :MEASure:CURRent[:DC]? and :MEASure[:VOLTage][:DC]? commands to query the current and voltage respectively, or send the :MEASure:ALL[:DC]? command to query the voltage, current and power at the same time.

Return Format The query returns the power, for example, 5.000.

Example :MEAS:POWE? CH1 /*Query the power of CH1 and the query returns 5.000*/

Related Commands :MEASure:ALL[:DC]?

:MEASure:CURRent[:DC]?

:MEASure[:VOLTage][:DC]?

:MEASure[:VOLTage][:DC]?

Syntax :MEASure[:VOLTage][:DC]? [{CH1|CH2|CH3}]

Description Query the output voltage measured internally on the specified channel.

Parameter

Name	Type	Range	Default
[{CH1|CH2|CH3}]	Discrete	CH1|CH2|CH3	None

Explanation

- If the parameter is omitted, the command queries the voltage of the channel currently selected.
- You can send the :MEASure:CURRent[:DC]? and :MEASure:POWEr[:DC]? commands to query the current and power respectively, or send the :MEASure:ALL[:DC]? command to query the voltage, current and power at the same time.

Return Format The query returns the voltage, for example, 5.0000.

Example :MEAS? CH1 /*Query the voltage of CH1 and the query returns 5.0000*/

Related Commands :MEASure:ALL[:DC]?

:MEASure:CURRent[:DC]?

:MEASure:POWEr[:DC]?

:MEMory Commands

The :MEMory commands are used to save, delete, read or lock the file stored in the specified storage location in the internal memory. DP800 allows four kinds of files to be saved in the internal memory.

1. State File (RSF): store the current system state, including the voltage, current, OVP, OCP and track function status of each channel as well as the system parameters.
2. Record File (ROF): store the output voltage, current and power of each channel when the recorder is enabled (for the channel of which the output is disabled, the corresponding recorded data will be 0).
3. Timer File (RTF): store the timer parameters edited (the voltage, current and time of each group of parameters).
4. Delay File (RDF): store the delayer parameters edited (the state and time of each group of parameters).

Command List:

- :MEMory[:STATe]:DELete
- :MEMory[:STATe]:LOAD
- :MEMory[:STATe]:LOCK
- :MEMory[:STATe]:STORe
- :MEMory[:STATe]:VALid?

:MEMory[:STATe]:DELete

Syntax :MEMory[:STATe]:DELete {RSF|ROF|RTF|RDF},{1|2|3|4|5|6|7|8|9|10}

Description Delete the specified file stored, including state file (RSF), record file (ROF), timer file (RTF) and delay file (RDF).

Parameter

Name	Type	Range	Default
{RSF|ROF|RTF|RDF}	Discrete	RSF|ROF|RTF|RDF	None
{1|2|3|4|5|6|7|8|9|10}	Discrete	1|2|3|4|5|6|7|8|9|10	None

Explanation

- This command is only available when the specified storage location contains file.
- This command is invalid when the file stored in the specified storage location is locked (refer to the :MEMory[:STATe]:LOCK command).

Example :MEM:DEL RSF,5 /*Delete the state file currently stored in storage location 5*/

:MEMory[:STATe]:LOAD

Syntax :MEMory[:STATe]:LOAD {RSF|RTF|RDF},{1|2|3|4|5|6|7|8|9|10}

Description Read the specified file stored, including state file (RSF), timer file (RTF) and delay file (RDF).

Parameter

Name	Type	Range	Default
{RSF|RTF|RDF}	Discrete	RSF|RTF|RDF	None
{1|2|3|4|5|6|7|8|9|10}	Discrete	1|2|3|4|5|6|7|8|9|10	None

Explanation This command is only available when the specified storage location contains file.

Example :MEM:LOAD RSF,5 /*Read the state file currently stored in storage location 5*/

Related Command *RCL

:MEMory[:STATe]:LOCK

Syntax :MEMory[:STATe]:LOCK {RSF|ROF|RTF|RDF},{1|2|3|4|5|6|7|8|9|10},{ON|OFF}

:MEMory[:STATe]:LOCK? {RSF|ROF|RTF|RDF},{1|2|3|4|5|6|7|8|9|10}

Description Lock or unlock the file in the specified storage location.

Query whether the file in the specified storage location is locked.

Parameter

Name	Type	Range	Default
{RSF|ROF|RTF|RDF}	Discrete	RSF|ROF|RTF|RDF	None
{1|2|3|4|5|6|7|8|9|10}	Discrete	1|2|3|4|5|6|7|8|9|10	None
{ON|OFF}	Discrete	ON|OFF	OFF

Explanation The locked file cannot be saved and deleted, but can be read.

Return Format The query returns YES or NO.

Example

```
:MEM:LOCK RSF,5,ON      /*Lock the file in state file storage location 5*/
:MEM:LOCK? RSF,5        /*Query the locking state of the file in state file storage
                          location 5 and the query returns YES*/
```

:MEMory[:STATe]:STORe

Syntax :MEMory[:STATe]:STORe {RSF|RTF|RDF},{1|2|3|4|5|6|7|8|9|10}

Description Save the specified type of file to the specified storage location.

Parameter

Name	Type	Range	Default
{RSF|RTF|RDF}	Discrete	RSF|RTF|RDF	None
{1|2|3|4|5|6|7|8|9|10}	Discrete	1|2|3|4|5|6|7|8|9|10	None

Explanation

- This command is used to store state file, timer file and delay file. The storage directory of the record file is specified by the :RECorder:MEMory command and the record file is stored automatically when the recorder is turned off.
- If the specified storage location already contains file, this command will overwrite the original file directly. If the file stored in the specified storage location is locked (refer to the :MEMory[:STATe]:LOCK command), this command will not overwrite the original file and the storage operation is invalid.

Example

```
:MEM:STOR RSF,5      /*Store the current instrument state to location 5*/
```

Related Commands *SAV

:RECorder:MEMory

:MEMory[:STATe]:VALid?

Syntax :MEMory[:STATe]:VALid? {RSF|ROF|RTF|RDF},{1|2|3|4|5|6|7|8|9|10}

Description Query whether the specified storage location contains a valid file.

Parameter

Name	Type	Range	Default
{RSF|ROF|RTF|RDF}	Discrete	RSF|ROF|RTF|RDF	None
{1|2|3|4|5|6|7|8|9|10}	Discrete	1|2|3|4|5|6|7|8|9|10	None

Explanation The read and delete operations are only available when the specified storage location contains a valid file.

Return Format The query returns YES or NO.

Example

```
:MEM:VAL? RSF,5   /*Query whether the state file storage location 5 contains a
                    valid file and the query returns YES*/
```

:MMEMory Commands

The :MMEMory commands are used to store the file to the specified external storage directory, read or delete the specified file in the external memory as well as query the disk information of the external memory.

Command List:

- :MMEMory:CATalog?
- :MMEMory:CDIRectory
- :MMEMory:DELete
- :MMEMory:DISK?
- :MMEMory:LOAD
- :MMEMory:MDIRectory
- :MMEMory:STORe

:MMEMory:CATalog?

Syntax :MMEMory:CATalog?

Description Query all the files and folders in the current operation directory.

Explanation This command is only applicable to external memory.

Return Format The query returns all the files and folders under the current operation directory, for example, RIGOL0.BMP,RIGOL1.BMP,RIGOL2.BMP,cc.RSF.

:MMEMory:CDIRectory

Syntax :MMEMory:CDIRectory <directory_name>

:MMEMory:CDIRectory?

Description Set the current operation directory.

Query the current operation directory.

Parameter

Name	Type	Range	Default
<directory_name>	ASCII character string	Valid directory	NULL

Explanation

- This command is only applicable to external memory. If the directory set does not exist, "Invalid directory." will be displayed.
- After setting the external storage directory to the current operation directory, send the :MMEMory:DELete, :MMEMory:LOAD or :MMEMory:STORe command to delete or read the file under the current directory or to save the file to the current directory. Sending the :MMEMory:MDIRectory command will create a folder under the current directory.

Return Format The query returns the current operation directory, for example, D:\RIGOL.

Example

```
:MMEM:CDIR D:\RIGOL   /*Set the current operation directory to D:\RIGOL*/
:MMEM:CDIR?           /*Query the current operation directory and the query
                        returns D:\RIGOL*/
```

Related Commands :MMEMory:DELete

:MMEMory:LOAD

:MMEMory:MDIRectory

:MMEMory:STORe

:MMEMory:DELete

Syntax :MMEMory:DELete <file_name>

Description Delete the specified file under the current operation directory.

Parameter

Name	Type	Range	Default
<file_name>	ASCII character string	Filename of the file to be deleted	None

Explanation

- This command is only available when the current operation directory contains the specified file.
- If the file to be deleted is not under the current operation directory, please use the :MMEMory:CDIRectory command to modify the current operation directory.

Example

```
:MMEM:DEL STA.RSF   /*Delete the STA.RSF file under the current operation
                      directory*/
```

Related Command :MMEMory:CDIRectory

:MMEMory:DISK?

Syntax :MMEMory:DISK?

Description Query the drive letter available of the power supply.

Explanation This command only queries the drive letter of the external memory of the power supply.

Return Format The query returns the drive letter available, for example, D:\.

Example :MMEM:DISK? /*Query the drive letter available and the query returns D:\*/

:MMEMory:LOAD

Syntax :MMEMory:LOAD <file_name>

Description Read the specified file under the current operation directory.

Parameter

Name	Type	Range	Default
<file_name>	ASCII character string	Filename of the file to be read	None

Explanation

- With this command, you can read the state file (RSF), timer file (RTF) and delay file (RDF).
- This command is only available when the current operation directory contains the specified file.
- If the file to be read is not under the current operation directory, please use the :MMEMory:CDIRectory command to modify the current operation directory.

Example :MMEM:LOAD STT.RSF /*Read the STT.RSF file under the current operation directory*/

Related Command :MMEMory:CDIRectory

:MMEMory:MDIRectory

Syntax :MMEMory:MDIRectory <dir_name>

Description Create a new folder under the current operation directory.

Parameter

Name	Type	Range	Default
<dir_name>	ASCII character string	Filename of the folder to be created, up to 9 characters, including Chinese characters, English characters or numbers	None

Example :MMEM:MDIR NEW /*Create a folder with the name NEW under the current operation directory*/

:MMEMory:STORe

Syntax :MMEMory:STORe <file_name>

Description Save the file with the specified filename under the current operation directory.

Parameter

Name	Type	Range	Default
<file_name>	ASCII character string	Filename of the file to be saved, including Chinese characters, English characters or numbers	None

Explanation

- With this command, you can save the state file (RSF), timer file (RTF) and delay file (RDF).
- If the current operation directory already contains a file with the same filename, this command will overwrite the original file directly.

Example

```
:MMEM:STORe STB.RSF   /*Save the current instrument state under the current
                        operation directory with the name STB.RSF*/
```

:MONItor Commands

The :MONItor commands are used to set the monitor condition and stop mode of the monitor as well as enable or disable the monitor.

Command List:

- :MONItor:CURRent:CONDition
- :MONItor:CURRent[:VALue]
- :MONItor:POWER:CONDition
- :MONItor:POWER[:VALue]
- :MONItor[:STATe]
- :MONItor:STOPway
- :MONItor:VOLTage:CONDition
- :MONItor:VOLTage[:VALue]

:MONItor:CURRent:CONDition

Syntax :MONItor:CURRent:CONDition {<C|>C|NONE},{AND|OR|NONE}

:MONItor:CURRent:CONDition?

Description Set the current monitor condition of the monitor.

Query the current monitor condition of the monitor.

Parameter

Name	Type	Range	Default
{<C|>C|NONE}	Discrete	<C|>C|NONE	NONE
{AND|OR|NONE}	Discrete	AND|OR|NONE	NONE

Explanation

- After setting the current monitor condition, sending the :MONItor:CURRent[:VALue] command can set the current value.
- You can set the current monitor condition to "<Curr" ot ">Curr".
- The actual monitor condition is the logic combination of the voltage, current and power. You need to set the logic relation between the current monitor condition and the voltage/power monitor conditions; you can set the logic relation to "AND", "OR" or "None"; wherein, "None" indicates that no logic relation will be set.

Return Format The query returns "condition,logic relation", for example, <C,AND.

Example

```
:MONI:CURR:COND <C,AND      /*Set the current monitor condition to "<Curr"
                              and the logic relation to "AND" */
:MONI:CURR:COND?           /*Query the current monitor condition and the
                             query returns <C,AND*/
```

Related Commands

:MONItor:CURRent[:VALue]

:MONItor:POWER:CONDition

:MONItor:POWER[:VALue]

:MONItor:VOLTage:CONDition

:MONItor:VOLTage[:VALue]

:MONItor:CURRent[:VALue]

Syntax :MONItor:CURRent[:VALue] {<value>|MINimum|MAXimum}

:MONItor:CURRent[:VALue]? [MINimum|MAXimum]

Description Set the current of the monitor condition.

Query the current of the monitor condition.

Parameter

Name	Type	Range	Default
<value>	Real	0 to the maximum current of the current channel	0.5*rated current of the current channel

Return Format The query returns the current of the monitor condition, for example, 4.0000.

Example :MONI:CURR 4 /*Set the current of the monitor condition to 4A*/

:MONI:CURR? /*Query the current of the monitor condition and the query returns 4.0000*/

Related Commands :MONItor:CURRent:CONDition

:MONItor:POWER:CONDition

:MONItor:POWER[:VALue]

:MONItor:VOLTage:CONDition

:MONItor:VOLTage[:VALue]

:MONItor:POWER:CONDition

Syntax :MONItor:POWER:CONDition {<P|>P|NONE}

:MONItor:POWER:CONDition?

Description Set the power monitor condition of the monitor to "<Power" or ">Power". NONE indicates that no power monitor condition will be set.

Query the power monitor condition of the monitor.

Parameter

Name	Type	Range	Default
{<P|>P|NONE}	Discrete	<P|>P|NONE	NONE

Explanation

➢ After setting the power monitor condition, sending the :MONItor:POWER[:VALue] command can set the power value.

➢ The actual monitor condition is the logic combination of the voltage, current and power.

Return Format The query returns "condition", for example, <P.

Example :MONI:POWER:COND <P /*Set the power monitor condition to "<Power"*/

:MONI:POWER:COND? /*Query the current power monitor condition and the query returns <P*/

Related Commands :MONItor:POWER[:VALue]

:MONItor:CURRent:CONDition

:MONItor:CURRent[:VALue]

:MONItor:VOLTage:CONDition

:MONItor:VOLTage[:VALue]

:MONItor:POWER[:VALue]

Syntax :MONItor:POWER[:VALue] {<value>|MINimum|MAXimum}

:MONItor:POWER[:VALue]? [MINimum|MAXimum]

Description Set the power of the monitor condition.

Query the power of the monitor condition.

Parameter

Name	Type	Range	Default
<value>	Real	0 to the maximum power of the current channel	0.25*rated power of the current channel

Return Format The query returns the power of the current monitor condition, for example, 20.000.

Example

```
:MONI:POWER 20      /*Set the power of the monitor condition to 20W*/
:MONI:POWER?        /*Query the power of the current monitor condition and the
                      query returns 20.000*/
```

Related Commands

:MONItor:CURRent:CONDition

:MONItor:CURRent[:VALue]

:MONItor:POWER:CONDition

:MONItor:VOLTage:CONDition

:MONItor:VOLTage[:VALue]

:MONItor[:STATe]

Syntax :MONItor[:STATe] {ON|OFF}

:MONItor[:STATe]?

Description Enable or disable the monitor.

Query the state of the monitor.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	OFF

Explanation When the monitor is enabled and the channel output state meets the monitor condition set, the instrument will turn off the output, display the corresponding prompt message or sound the beeper according to the stop mode selected.

Return Format The query returns ON or OFF.

Example

```
:MONI ON      /*Enable the monitor*/
:MONI?        /*Query the state of the monitor and the query returns ON*/
```

:MONItor:STOPway

Syntax

:MONItor:STOPway {OUTOFF|WARN|BEEPER|NONE},{ON|OFF}

:MONItor:STOPway?

Description

Set the stop mode of the monitor. Multiple stop modes can be enabled at the same time.

Query the stop mode of the monitor.

Parameter

Name	Type	Range	Default
{OUTOFF|WARN|BEEPER|NONE}	Discrete	OUTOFF|WARN|BEEPER	None
{ON|OFF}	Bool	ON|OFF	ON

Explanation

When the channel output state meets the monitor condition set, the instrument will turn off the output, display the corresponding prompt message or sound the beeper according to the stop mode selected.

Return Format

The query returns the on/off status of OUTOFF, WARN and BEEPER, for example, OutputOff:ON,Warn:ON,Beep:ON.

Example

```
:MONI:STOP BEEPER,OFF     /*Disable the "Beeper" stop mode*/
:MONI:STOP?         /*Query the current states of the stop modes of the monitor*/
                    /*The query returns OutputOff:ON,Warn:ON,Beep:OFF*/
```

:MONItor:VOLTage:CONDition

Syntax :MONItor:VOLTage:CONDition {<V|>V|NONE},{AND|OR|NONE}

:MONItor:VOLTage:CONDition?

Description Set the voltage monitor condition of the monitor.

Query the voltage monitor condition of the monitor.

Parameter

Name	Type	Range	Default
{<V|>V|NONE}	Discrete	<V|>V|NONE	NONE
{AND|OR|NONE}	Discrete	AND|OR|NONE	NONE

Explanation

- After setting the voltage monitor condition, sending the :MONItor:VOLTage[:VALue] <value> command can set the voltage value.
- You can set the voltage monitor condition to "<Volt" ot ">Volt".
- The actual monitor condition is the logic combination of the voltage, current and power. You need to set the logic relation between the voltage monitor condition and the current/power monitor conditions; you can set the logic relation to "AND", "OR" or "None"; wherein, "None" indicates that no logic relation will be set.

Return Format The query returns "condition,logic relation", for example, <V,AND.

Example

```
:MONI:VOLT:COND <V,AND      /*Set the voltage monitor condition to "<Volt" and
                              logic relation to "AND"*/
:MONI:VOLT:COND?            /*Query the current voltage monitor condition and
                              the query returns <V,AND*/
```

Related Commands :MONItor:CURRent:CONDition

:MONItor:CURRent[:VALue]

:MONItor:POWER:CONDition

:MONItor:POWER[:VALue]

:MONItor:VOLTage[:VALue]

:MONItor:VOLTage[:VALue]

Syntax :MONItor:VOLTage[:VALue] {<value>|MINimum|MAXimum}

:MONItor:VOLTage[:VALue]? [MINimum|MAXimum]

Description Set the voltage of the monitor condition.

Query the voltage of the monitor condition.

Parameter

Name	Type	Range	Default
<value>	Real	0 to the maximum voltage of the current channel	0.5*rated voltage of the current channel

Return Format The query returns the voltage of the current monitor condition, for example, 5.000.

Example

```
:MONI:VOLT 5      /*Set the voltage of the monitor condition to 5V*/
:MONI:VOLT?       /*Query the voltage of the current monitor condition and the
                    query returns 5.000*/
```

Related Commands

:MONItor:CURRent:CONDition

:MONItor:CURRent[:VALue]

:MONItor:POWER:CONDition

:MONItor:POWER[:VALue]

:MONItor:VOLTage:CONDition

:OUTPut Commands

Command List:

- :OUTPut:MODE?
- :OUTPut:OCP:CLEAR
- :OUTPut:OCP:QUES?
- :OUTPut:OCP[:STATe]
- :OUTPut:OCP:VALue
- :OUTPut:OVP:CLEAR
- :OUTPut:OVP:QUES?
- :OUTPut:OVP[:STATe]
- :OUTPut:OVP:VALue
- :OUTPut[:STATe]
- :OUTPut:TRACk

:OUTPut:MODE?

Syntax :OUTPut:MODE? {CH1|CH2|CH3}

Description Query the current output mode of the specified channel.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	None

Explanation The output modes of the power supply include CV (Constant Voltage), CC (Constant Current) and UR. In CV mode, the output voltage equals the voltage setting value and the output current is determined by the load; in CC mode, the output current equals the current setting value and the output voltage is determined by the load; UR is the critical mode between CV and CC.

Return Format The query returns CV, CC or UR.

Example :OUTP:MODE? CH1 /*Query the output mode of CH1 and the query returns CV*/

:OUTPut:OCP:CLEAR

Syntax :OUTPut:OCP:CLEAR {CH1|CH2|CH3}

Description Clear the overcurrent protection occurred on the specified channel.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	None

Example

```
:OUTP:OCP:QUES? CH1   /*Query whether overcurrent protection currently
                        occurred on CH1 and the query returns YES*/
:OUTP:OCP:CLEAR CH1   /*Clear the overcurrent protection occurred on CH1*/
:OUTP:OCP:QUES? CH1   /*The query returns NO*/
```

Related Commands :OUTPut:OCP:VALue

:OUTPut:OCP:QUES?

:OUTPut:OCP:QUES?

Syntax :OUTPut:OCP:QUES? {CH1|CH2|CH3}

Description Query whether overcurrent protection occurs to the specified channel currently, namely the output current exceeds the overcurrent protection limit and the output turns off automatically.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	None

Explanation You can clear the overcurrent protection occurred on the specified channel using the :OUTPut:OCP:CLEAR command.

Return Format The query returns YES or NO.

Example :OUTP:OCP:QUES? CH1 /*Query whether overcurrent protection occurs to CH1 currently and the query returns YES*/

Related Commands :OUTPut:OCP:CLEAR

:OUTPut:OCP[:STATe]

:OUTPut:OCP:VALue

:OUTPut:OCP[:STATe]

Syntax :OUTPut:OCP[:STATe] {CH1|CH2|CH3},{ON|OFF}

:OUTPut:OCP[:STATe]? {CH1|CH2|CH3}

Description Enable or disable the overcurrent protection function of the specified channel.

Query the status of the overcurrent protection function of the specified channel.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	None
{ON|OFF}	Bool	ON|OFF	OFF

Explanation When the overcurrent protection function is enabled, the output will turn off automatically when the output current exceeds the overcurrent protection limit currently set (:OUTPut:OCP:VALue). You can send the :OUTPut:OCP:QUES? command to query whether overcurrent protection occurs to the specified channel currently.

Return Format The query returns ON or OFF.

Example :OUTP:OCP CH1,ON /*Enable the overcurrent protection function of CH1*/

:OUTP:OCP? CH1 /*Query the status of the overcurrent protection function of CH1 and the query returns ON*/

Related Commands :OUTPut:OCP:QUES?

:OUTPut:OCP:VALue

:OUTPut:OCP:VALue

Syntax :OUTPut:OCP:VALue {CH1|CH2|CH3},{<value>|MINimum|MAXimum}

:OUTPut:OCP:VALue? {CH1|CH2|CH3}[,MINimum|MAXimum]

Description Set the overcurrent protection limit of the specified channel.

Query the overcurrent protection limit of the specified channel.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	None
<value>	Real	Current range of the current channel	Maximum current of the current channel

Explanation When the overcurrent protection function is enabled, the output will turn off automatically when the output current exceeds the overcurrent protection limit currently set. You can send the :OUTPut:OCP:QUES? command to query whether overcurrent protection occurs to the specified channel currently.

Return Format The query returns the overcurrent protection limit, for example, 5.0000.

Example :OUTP:OCP:VAL CH1,5

:OUTP:OCP:VAL? CH1

Related Commands :OUTPut:OCP:QUES?

:OUTPut:OCP[:STATe]

:OUTPut:OVP:CLEAR

Syntax :OUTPut:OVP:CLEAR {CH1|CH2|CH3}

Description Clear the overvoltage protection occurred on the specified channel.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	None

Example

```
:OUTP:OVP:QUES? CH1   /*Query whether overvoltage protection currently
                        occurred on CH1 and the query returns YES*/
:OUTP:OVP:CLEAR CH1   /*Clear the overvoltage protection occurred on CH1*/
:OUTP:OVP:QUES? CH1   /*The query returns NO*/
```

Related Commands :OUTPut:OVP:QUES?

:OUTPut:OVP:VALue

:OUTPut:OVP:QUES?

Syntax :OUTPut:OVP:QUES? {CH1|CH2|CH3}

Description Query whether overvoltage protection occurs to the specified channel, namely the output voltage exceeds the overvoltage protection limit and the output turns off automatically.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	None

Explanation You can clear the overvoltage protection occurred on the specified channel using the :OUTPut:OVP:CLEAR command.

Return Format The query returns YES or NO.

Example :OUTP:OVP:QUES? CH1

Related Command :OUTPut:OVP:CLEAR

:OUTPut:OVP[:STATe]

Syntax :OUTPut:OVP[:STATe] {CH1|CH2|CH3},{ON|OFF}

:OUTPut:OVP[:STATe]? {CH1|CH2|CH3}

Description Enable or disable the overvoltage protection function of the specified channel.

Query the status of the overvoltage protection function of the specified channel.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	None
{ON|OFF}	Bool	ON|OFF	OFF

Explanation When the overvoltage protection function is enabled, the output will turn off automatically when the output voltage exceeds the overvoltage protection limit currently set (:OUTPut:OVP:VALue). You can send the :OUTPut:OVP:QUES? command to query whether overvoltage protection occurs to the specified channel currently.

Return Format The query returns ON or OFF.

Example

```
:OUTP:OVP CH1,ON      /*Enable the overvoltage protection function of CH1*/
:OUTP:OVP? CH1        /*Query the status of the overvoltage protection function
                        of CH1 and the query returns ON*/
```

Related Commands :OUTPut:OVP:QUES?

:OUTPut:OVP:VALue

:OUTPut:OVP:VALue

Syntax :OUTPut:OVP:VALue {CH1|CH2|CH3},{<value>|MINimum|MAXimum}

:OUTPut:OVP:VALue? {CH1|CH2|CH3}[,MINimum|MAXimum]

Description Set the overvoltage protection limit of the specified channel.

Query the overvoltage protection limit of the specified channel.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	None
<value>	Real	Voltage range of the current channel	Maximum voltage of the current channel

Explanation When the overvoltage protection function is enabled, the output will turn off automatically when the output voltage exceeds the overvoltage protection limit currently set. You can send the :OUTPut:OVP:QUES? command to query whether overvoltage protection occurs to the specified channel currently.

Return Format The query returns the overvoltage protection limit, for example, 8.800.

Example

```
:OUTP:OVP:VAL CH1,8.8     /*Set the overvoltage protection limit of CH1 to 8.8V*/
:OUTP:OVP:VAL? CH1        /*Query the overvoltage protection limit of CH1 and the
                            query returns 8.800*/
```

Related Commands :OUTPut:OVP[:STATe]

:OUTPut:OVP:QUES?

:OUTPut[:STATe]

Syntax :OUTPut[:STATe] [{CH1|CH2|CH3},]{ON|OFF}

:OUTPut[:STATe]? [{CH1|CH2|CH3}]

Description Enable or disable the output of the specified channel.

Query the output status of the specified channel.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	None
{ON|OFF}	Bool	ON|OFF	OFF

Explanation Make sure that the current setting will not affect the device connected to the power supply before enabling the channel output.

Return Format The query returns ON or OFF.

Example

```
:OUTP CH1,ON     /*Enable the output of CH1*/
:OUTP? CH1       /*Query the current output status of CH1 and the query returns
                   ON*/
```

:OUTPut:TRACk

Syntax

:OUTPut:TRACk {CH1|CH2|CH3},{ON|OFF}

:OUTPut:TRACk? [{CH1|CH2|CH3}]

Description

Enable or disable the track function of the specified channel.

Query the status of the track function of the specified channel.

Parameter

Name	Type	Range	Default
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	None
{ON|OFF}	Bool	ON|OFF	OFF

Explanation

- This command is only applicable to channels supporting the track function: CH2 and CH3 of DP831A as well as CH1 and CH2 of DP832A/DP832. For channels that do not support the track function, the query returns NONE.
- For two channels supporting the track function, when the track function of one channel (the tracked channel) is enabled, the voltage setting value of the other channel (the tracking channel, the track status label (for example, ∞) is displayed in the display area corresponding to this channel and the voltage value can not be set) will change accordingly when the voltage setting value of this channel is changed.
- By default, the track function is disabled and it is usually used to provide symmetric voltage for the calculation amplifier or other circuits.
- The track function only tracks the voltage setting value and the actual output voltage will not be affected.

Return Format

The query returns ON or OFF. For channels that do not support the track function, the query returns NONE.

Example

```
:OUTP:TRAC CH1,ON
:OUTP:TRAC? CH1
```

:PRESet Commands

Command List:

- :PRESet[:APPLy]
- :PRESet:KEY
- :PRESet:USER[n]:SET:CURRent
- :PRESet:USER[n]:SET:DEFault
- :PRESet:USER[n]:SET:TRACk
- :PRESet:USER[n]:SET:OCP
- :PRESet:USER[n]:SET:OVP
- :PRESet:USER[n]:SET:OTP
- :PRESet:USER[n]:SET:SURE
- :PRESet:USER[n]:SET:VOLTage

Note: When [n] is omitted, the operation is performed on the user-preset state currently selected by default.

:PRESet[:APPLy]

Syntax :PRESet[:APPLy]

Description Restore the instrument to default value or user-preset value.

Explanation Sending this command is equivalent to pressing **Preset** at the front panel, namely recalling the default value or user-preset value related to the key (depend on the :PRESet:KEY command).

Related Command :PRESet:KEY

:PRESet:KEY

Syntax

```
:PRESet:KEY {DEFAULT|USER1|USER2|USER3|USER4}
:PRESet:KEY?
```

Description Define the setting recalled by **Preset** at the front panel.

Query the setting recalled by **Preset** at the front panel.

Parameter

Name	Type	Range	Default
{DEFAULT|USER1|USER2|USER3|USER4}	Discrete	DEFAULT|USER1|USER2|USER3|USER4	DEFAULT

Explanation

- DP800 supports to restore the instrument to default state or user-preset state (up to 4 kinds).
- After sending this command to set the instrument, sending the :PRESet[:APPLy] command or pressing **Preset** at the front panel can restore the instrument to the default value or user-preset value.
- When user-preset state is selected, you can use the :PRESet:USER:SET series commands to set the parameters of the user-preset state currently selected.

Return Format The query returns DEFAULT, USER1, USER2, USER3 or USER4.

Example

```
:PRES:KEY USER1
:PRES:KEY?
```

Related Command :PRESet Commands

:PRESet:USER[n]:SET:CURRent

Syntax

:PRESet:USER[n]:SET:CURRent {<current>|MINimum|MAXimum}

:PRESet:USER[n]:SET:CURRent? [MINimum|MAXimum]

Description

Set the current of the specified user-preset state.

Query the current of the specified user-preset state.

Parameter

Name	Type	Range	Default
[n]	Discrete	1|2|3|4	None
<current>	Real	Current range of the current channel	Refer to the "**Explanation**"

Explanation

- ➢ By default, this command sets the parameter of the channel currently selected. If you want to set the parameter of other channel, please use the :INSTrument[:SELEct] command to select the desired channel.
- ➢ The current default values of each user-preset state are as shown in the table below.

	DP831A			DP832A/DP832		
	CH1	CH2	CH3	CH1	CH2	CH3
USER1	0.5000A	1.0000A	1.0000A	1.0000A	1.0000A	1.0000A
USER2	1.0000A	1.0000A	1.0000A	1.0000A	1.0000A	1.0000A
USER3	1.0000A	1.5000A	1.5000A	2.0000A	2.0000A	1.0000A
USER4	1.0000A	2.0000A	2.0000A	3.0000A	3.0000A	2.0000A

Return Format

The query returns the current, for example, 1.5000.

Example

```
:PRES:USER1:SET:CURR 1.5
:PRES:USER1:SET:CURR?
```

:PRESet:USER[n]:SET:DEFault

Syntax

:PRESet:USER[n]:SET:DEFault

Description

Restore the parameters of the specified user-preset state to default values.

Parameter

Name	Type	Range	Default
[n]	Discrete	1|2|3|4	None

Explanation

For the default values of the user-preset states, refer to the "**Explanation**" of the :PRESet:USER[n]:SET:CURRent and :PRESet:USER[n]:SET:VOLTage commands.

Related Commands

:PRESet:USER[n]:SET:CURRent

:PRESet:USER[n]:SET:VOLTage

:PRESet:USER[n]:SET:TRACk

Syntax :PRESet:USER[n]:SET:TRACk {ON|OFF}

:PRESet:USER[n]:SET:TRACk?

Description Enable or disable the track function of the specified user-preset state.

Query the status of the track function of the specified user-preset state.

Parameter

Name	Type	Range	Default
[n]	Discrete	1|2|3|4	None
{ON|OFF}	Bool	ON|OFF	OFF

Explanation

- By default, this command sets the parameter of the channel currently selected. If you want to set the parameter of other channel, please use the :INSTrument[:SELEct] command to select the desired channel.
- This command is only applicable to channels supporting the track function: CH2 and CH3 of DP831A as well as CH1 and CH2 of DP832A/DP832. For channels that do not support the track function, the query returns NONE.

Return Format The query returns ON or OFF. For channels that do not support the track function, the query returns NONE.

Example :PRES:USER1:SET:TRAC ON

:PRES:USER1:SET:TRAC?

:PRESet:USER[n]:SET:OCP

Syntax :PRESet:USER[n]:SET:OCP {ON|OFF}[,<current>|MINimum|MAXimum]

:PRESet:USER[n]:SET:OCP? [MINimum|MAXimum]

Description Enable or disable the overcurrent protection function of the specified user-preset state and set the overcurrent protection limit.

Query the status and limit of the overcurrent protection function of the specified user-preset state.

Parameter

Name	Type	Range	Default
[n]	Discrete	1|2|3|4	None
{ON|OFF}	Bool	ON|OFF	OFF
<current>	Real	Current range of the current channel	Maximum current of the current channel

Explanation By default, this command sets the parameter of the channel currently selected. If you want to set the parameter of other channel, please use the :INSTrument[:SELEct] command to select the desired channel.

Return Format The query returns OFF or ON,limit of overcurrent protection, for example, ON,1.5000.

Example :PRES:USER1:SET:OCP ON,1.5

:PRES:USER1:SET:OCP?

:PRESet:USER[n]:SET:OVP

Syntax :PRESet:USER[n]:SET:OVP {ON|OFF}[,<voltage>|MINimum|MAXimum]

:PRESet:USER[n]:SET:OVP? [MINimum|MAXimum]

Description Enable or disable the overvoltage protection function of the specified user-preset state and set the overvoltage protection limit.

Query the status and limit of the overvoltage protection function of the specified user-preset state.

Parameter

Name	Type	Range	Default
[n]	Discrete	1|2|3|4	None
{ON|OFF}	Bool	ON|OFF	OFF
<voltage>	Real	Voltage range of the current channel	Maximum voltage of the current channel

Explanation By default, this command sets the parameter of the channel currently selected. If you want to set the parameter of other channel, please use the :INSTrument[:SELEct] command to select the desired channel.

Return Format The query returns OFF or ON,limit of overvoltage protection,for example, ON,8.800.

Example :PRES:USER1:SET:OVP ON,8.8

:PRES:USER1:SET:OVP?

:PRESet:USER[n]:SET:OTP

Syntax :PRESet:USER[n]:SET:OTP {ON|OFF}

:PRESet:USER[n]:SET:OTP?

Description Enable or disable the over-temperature protection function of the specified user-preset state.

Query the status of the over-temperature protection function of the specified user-preset state.

Parameter

Name	Type	Range	Default
[n]	Discrete	1|2|3|4	None
{ON|OFF}	Bool	ON|OFF	ON

Return Format The query returns ON or OFF.

Example :PRES:USER1:SET:OTP ON

:PRES:USER1:SET:OTP?

:PRESet:USER[n]:SET:SURE

Syntax :PRESet:USER[n]:SET:SURE

Description Confirm the setting of the specified user-preset state.

Parameter

Name	Type	Range	Default
[n]	Discrete	1|2|3|4	None

:PRESet:USER[n]:SET:VOLTage

Syntax :PRESet:USER[n]:SET:VOLTage <voltage>|MINimum|MAXimum}

:PRESet:USER[n]:SET:VOLTage? [MINimum|MAXimum]

Description Set the voltage of the specified user-preset state.

Query the voltage of the specified user-preset state.

Parameter

Name	Type	Range	Default
[n]	Discrete	1|2|3|4	None
<voltage>	Real	Voltage range of the current channel	Refer to the "**Explanation**"

Explanation

- By default, this command sets the parameter of the channel currently selected. If you want to set the parameter of other channel, please use the :INSTrument[:SELEct] command to select the desired channel.
- The voltage default values of each user-preset state are as shown in the table below.

	DP831A			DP832A/DP832		
	CH1	CH2	CH3	CH1	CH2	CH3
USER1	1.500V	3.300V	-5.000V	3.300V	3.300V	1.500V
USER2	3.300V	4.200V	-4.200V	4.200V	4.200V	4.200V
USER3	3.300V	5.000V	-8.000V	5.000V	8.000V	3.300V
USER4	5.000V	12.000V	-12.000V	12.000V	24.000V	5.000V

Return Format The query returns the voltage, for example, 5.000.

Example :PRES:USER1:SET:VOLT 5

:PRES:USER1:SET:VOLT?

:RECorder Commands

Command List:

- :RECorder:DESTination?
- :RECorder:MEMory
- :RECorder:MMEMory
- :RECorder:PERIod
- :RECorder[:STATe]

:RECorder:DESTination?

Syntax :RECorder:DESTination?

Description Query the storage directory of the record file.

Explanation

- Before enabling the recorder, use the :RECorder:MEMory or :RECorder:MMEMory command to set the storage directory.
- After the recording is finished, the instrument stores the record file to the storage directory using the specified filename.

Return Format The query returns the current storage directory, for example, C:\REC 10:RIGOL.ROF.

Related Commands :RECorder:MEMory

:RECorder:MMEMory

:RECorder[:STATe]

:RECorder:MEMory

Syntax :RECorder:MEMory {1|2|3|4|5|6|7|8|9|10},<filename>

Description Set the storage directory of the record file to a storage location of the internal memory.

Parameter

Name	Type	Range	Default
{1|2|3|4|5|6|7|8|9|10}	Discrete	1|2|3|4|5|6|7|8|9|10	10
<filename>	ASCII character string	<name>.ROF; wherein, <name> can be Chinese characters, English characters or numbers with up to 9 characters	RIGOL.ROF

Explanation

- You can use the :RECorder:MMEMory command to set the storage directory of the record file to a directory of the external memory.
- Using the :RECorder:DESTination? command can query the current storage directory.

Example :REC:MEM 5,test1

Related Commands :RECorder:DESTination?

:RECorder:MMEMory

:RECorder[:STATe]

:RECorder:MMEMory

Syntax :RECorder:MMEMory <dest>

Description Set the storage directory of the record file to a directory of the external memory.

Parameter

Name	Type	Range	Default
<dest>	ASCII character string	Directory of the external memory	None

Explanation

- This command is only available when an USB storage device is connected to the power supply.
- You can use the :RECorder:MEMory command to set the storage directory of the record file to a storage location of the internal memory.
- Using the :RECorder:DESTination? command can query the current storage directory.

Example :RECorder:MMEMory D:\ra.ROF

Related Commands :RECorder:DESTination?

:RECorder:MEMory

:RECorder[:STATe]

:RECorder:PERIod

Syntax :RECorder:PERIod <value>

:RECorder:PERIod?

Description Set the record period, namely the time interval at which the instrument samples and records the output of each channel when the recorder is enabled.

Query the record period.

Parameter

Name	Type	Range	Default
<value>	Integer	1s to 99999s	1s

Explanation The query returns an integer from 1 to 99999.

Example

```
:REC:PERI 5     /*Set the record period to 5s*/
:REC:PERI?      /*Query the record period and the query returns 5*/
```

Related Command :RECorder[:STATe]

:RECorder[:STATe]

Syntax :RECorder[:STATe] {ON|OFF}

:RECorder[:STATe]?

Description Enable or disable the recorder.

Query the status of the recorder.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	OFF

Explanation

- When the recorder is enabled, you cannot set the record period and storage directory. The instrument samples and records the output of each channel at the time interval equaling the current record period.
- During the recording, make sure that the output of each channel is enabled. For the channel of which the output is disabled, the corresponding recorded data will be 0.
- When the recorder is disabled, the recording finishes and the instrument stores the record file to the storage directory currently set.

Return Format The query returns ON or OFF.

Example

```
:REC ON      /*Enable the recorder*/
:REC?        /*Query the current status of the recorder and the query returns ON*/
```

Related Commands :RECorder:DESTination?

:RECorder:MEMory

:RECorder:MMEMory

:RECorder:PERIod

:SOURce Commands

The :SOURce commands are used to set the voltage, current, OVP and OCP of the specified channel. Their functions are equivalent to that of :APPLy Command. Although the :APPLy command provides the most straightforward method to program the power supply over the remote interfaces, the :SOURce commands give you more flexibility to change individual parameters.

Command List:

- ◆ [:SOURce[n]]:CURRent[:LEVel][:IMMediate][:AMPLitude]
- ◆ [:SOURce[n]]:CURRent:PROTection[:LEVel]
- ◆ [:SOURce[n]]:CURRent:PROTection:STATe
- ◆ [:SOURce[n]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]
- ◆ [:SOURce[n]]:VOLTage:PROTection[:LEVel]
- ◆ [:SOURce[n]]:VOLTage:PROTection:STATe

Note: When [:SOURce[n]] or [n] is omitted, the parameter of the channel currently selected will be set by default.

[:SOURce[n]]:CURRent[:LEVel][:IMMediate][:AMPLitude]

Syntax

[:SOURce[n]]:CURRent[:LEVel][:IMMediate][:AMPLitude] {<current>|MINimum|MAXimum}

[:SOURce[n]]:CURRent[:LEVel][:IMMediate][:AMPLitude]? [{MINimum|MAXimum}]

Description

Set the current of the specified channel.

Query the current of the specified channel.

Parameter

Name	Type	Range	Default
[n]	Integer	1|2|3	1
<current>	Real	Refer to the "**Explanation**" of :APPLy Command	

Explanation

When [:SOURce[n]] is omitted, the command sets the parameter of the channel currently selected by default.

Return Format

The query returns the current of the specified channel, for example, 1.5000.

Example

:CURR 1.5

:CURR?

Related Command

:APPLy Command

[:SOURce[n]]:CURRent:PROTection[:LEVel]

Syntax

```
[:SOURce[n]]:CURRent:PROTection[:LEVel] {<current>|MINimum|MAXimum}
[:SOURce[n]]:CURRent:PROTection[:LEVel]? [{MINimum|MAXimum}]
```

Description Set the overcurrent protection limit of the specified channel.

Query the overcurrent protection limit of the specified channel.

Parameter

Name	Type	Range	Default
[n]	Integer	1|2|3	1
<current>	Real	Current range of the current channel	Maximum current of the current channel

Explanation

- When the overcurrent protection function is enabled, the output turns off automatically when the output current exceeds the overcurrent protection limit currently set. You can send the :OUTPut:OCP:QUES? command to query whether overcurrent protection occurs to the specified channel currently.
- The function of this command is equivalent to that of the :OUTPut:OCP:VALue command.
- When [:SOURce[n]] is omitted, the command sets the parameter of the channel currently selected by default.

Return Format The query returns the overcurrent protection limit, for example, 5.0000.

Example

```
:CURR:PROT 5
:CURR:PROT?
```

Related Commands

:OUTPut:OCP:QUES?

:OUTPut:OCP[:STATe]

:OUTPut:OCP:VALue

[:SOURce[n]]:CURRent:PROTection:STATe

Syntax

[:SOURce[n]]:CURRent:PROTection:STATe {ON|OFF}

[:SOURce[n]]:CURRent:PROTection:STATe?

Description

Enable or disable the overcurrent protection function of the specified channel.

Query the status of the overcurrent protection function of the specified channel.

Parameter

Name	Type	Range	Default
[n]	Integer	1|2|3	1
{ON|OFF}	Bool	ON|OFF	OFF

Explanation

- When the overcurrent protection function is enabled, the output turns off automatically when the output current exceeds the overcurrent protection limit currently set. You can send the :OUTPut:OCP:QUES? command to query whether overcurrent protection occurs to the specified channel.
- The function of this command is equivalent to that of the :OUTPut:OCP[:STATe] command.
- When [:SOURce[n]] is omitted, the command sets the parameter of the channel currently selected by default.

Return Format

The query returns ON or OFF.

Example

```
:CURR:PROT:STAT ON
:CURR:PROT:STAT?
```

Related Commands

:OUTPut:OCP:QUES?

:OUTPut:OCP[:STATe]

:OUTPut:OCP:VALue

[:SOURce[n]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]

Syntax

[:SOURce[n]]:VOLTage[:LEVel][:IMMediate][:AMPLitude] {<voltage>|MINimum|MAXimum}

[:SOURce[n]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]? [{MINimum|MAXimum}]

Description

Set the voltage of the specified channel.

Query the voltage of the specified channel.

Parameter

Name	Type	Range	Default
[n]	Integer	1|2|3	1
<voltage>	Real	Refer to the “**Explanation**” of :APPLy Command	

Explanation

When [:SOURce[n]] is omitted, the command sets the parameter of the channel currently selected by default.

Return Format

The query returns the voltage of the specified channel, for example, 8.500.

Example

```
:VOLT 8.5
:VOLT?
```

Related Command

:APPLy Command

[:SOURce[n]]:VOLTage:PROTection[:LEVel]

Syntax

[:SOURce[n]]:VOLTage:PROTection[:LEVel] {<voltage>|MINimum|MAXimum}

[:SOURce[n]]:VOLTage:PROTection[:LEVel]? [{MINimum|MAXimum}]

Description

Set the overvoltage protection limit of the specified channel.

Query the overvoltage protection limit of the specified channel.

Parameter

Name	Type	Range	Default
[n]	Integer	1|2|3	1
<voltage>	Real	Voltage range of the current channel	Maximum voltage of the current channel

Explanation

- ➢ When the overvoltage protection function is enabled, the output turns off automatically when the output voltage exceeds the overvoltage protection limit currently set. You can send the :OUTPut:OVP:QUES? command to query whether overvoltage protection occurs to the specified channel currently.
- ➢ The function of this command is equivalent to that of the :OUTPut:OVP:VALue command.
- ➢ When [:SOURce[n]] is omitted, the command sets the parameter of the channel currently selected by default.

Return Format

The query returns the overvoltage protection limit, for example, 8.800.

Example

```
:VOLT:PROT 8.8
:VOLT:PROT?
```

Related Commands

:OUTPut:OVP:QUES?

:OUTPut:OVP[:STATe]

:OUTPut:OVP:VALue

[:SOURce[n]]:VOLTage:PROTection:STATe

Syntax

[:SOURce[n]]:VOLTage:PROTection:STATe {ON|OFF}

[:SOURce[n]]:VOLTage:PROTection:STATe?

Description

Enable or disable the overvoltage protection function of the specified channel.

Query the status of the overvoltage protection function of the specified channel.

Parameter

Name	Type	Range	Default
[n]	Integer	1|2|3	1
{ON|OFF}	Bool	ON|OFF	OFF

Explanation

- When the overvoltage protection function is enabled, the output turns off automatically when the output voltage exceeds the overvoltage protection limit currently set. You can send the :OUTPut:OVP:QUES? command to query whether overvoltage protection occurs to the specified channel currently.
- The function of this command is equivalent to that of the :OUTPut:OVP[:STATe] command.
- When [:SOURce[n]] is omitted, the command sets the parameter of the channel currently selected by default.

Return Format

The query returns ON or OFF.

Example

```
:VOLT:PROT:STAT ON
:VOLT:PROT:STAT?
```

Related Commands

:OUTPut:OVP:QUES?

:OUTPut:OVP[:STATe]

:OUTPut:OVP:VALue

:SYSTem Commands

Command List:

- :SYSTem:BEEPer[:IMMediate]
- :SYSTem:BEEPer:STATe
- :SYSTem:BRIGhtness
- :SYSTem:COMMunicate:GPIB:ADDRess
- :SYSTem:COMMunicate:LAN:APPLy
- :SYSTem:COMMunicate:LAN:AUTOip[:STATe]
- :SYSTem:COMMunicate:LAN:DHCP[:STATe]
- :SYSTem:COMMunicate:LAN:DNS
- :SYSTem:COMMunicate:LAN:GATEway
- :SYSTem:COMMunicate:LAN:IPADdress
- :SYSTem:COMMunicate:LAN:MAC?
- :SYSTem:COMMunicate:LAN:MANualip[:STATe]
- :SYSTem:COMMunicate:LAN:SMASK
- :SYSTem:COMMunicate:RS232:BAUD
- :SYSTem:COMMunicate:RS232:DATABit
- :SYSTem:COMMunicate:RS232:FLOWCrl
- :SYSTem:COMMunicate:RS232:PARItybit
- :SYSTem:COMMunicate:RS232:STOPBit
- :SYSTem:CONTrast
- :SYSTem:ERRor?
- :SYSTem:LANGuage:TYPE
- :SYSTem:LOCal
- :SYSTem:LOCK
- :SYSTem:OTP
- :SYSTem:POWEron
- :SYSTem:RGBBrightness
- :SYSTem:SAVer
- :SYSTem:SELF:TEST:BOARD?
- :SYSTem:SELF:TEST:FAN?
- :SYSTem:SELF:TEST:TEMP?

:SYSTem:BEEPer[:IMMediate]

Syntax :SYSTem:BEEPer[:IMMediate]

Description Send this command and the beeper sounds.

Example :SYST:BEEP

:SYSTem:BEEPer:STATe

Syntax :SYSTem:BEEPer:STATe {ON|OFF}

:SYSTem:BEEPer:STATe?

Description Enable or disable the beeper.

Query the status of the beeper.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	ON

Explanation When the beeper is enabled, the instrument generates prompt sound during front panel operation or when error occurs during remote operation.

Return Format The query returns ON or OFF.

Example :SYST:BEEP:STAT ON

:SYST:BEEP:STAT?

:SYSTem:BRIGhtness

Syntax :SYSTem:BRIGhtness {<brightness>|MINimum|MAXimum}

:SYSTem:BRIGhtness? [{MINimum|MAXimum}]

Description Set the brightness of the screen.

Query the brightness of the screen.

Parameter

Name	Type	Range	Default
<brightness>	Integer	1 to 100	50 (factory setting)

Return Format The query returns an integer, for example, 60.

Example :SYST:BRIG 60

:SYST:BRIG?

:SYSTem:COMMunicate:GPIB:ADDRess

Syntax :SYSTem:COMMunicate:GPIB:ADDRess <gpibaddress>

:SYSTem:COMMunicate:GPIB:ADDRess?

Description Set the GPIB address.

Query the current GPIB address.

Parameter

Name	Type	Range	Default
<gpibaddress>	Integer	0 to 30	2

Explanation Before using the GPIB interface, extend a GPIB interface using the USB-GPIB interface converter; then, connect the instrument and PC using GPIB cable and set the GPIB address.

Return Format The query returns an integer, for example, 7.

Example :SYST:COMM:GPIB:ADDR 7

:SYST:COMM:GPIB:ADDR?

:SYSTem:COMMunicate:LAN:APPLy

Syntax :SYSTem:COMMunicate:LAN:APPLy

Description Apply the network parameters currently set.

:SYSTem:COMMunicate:LAN:AUTOip[:STATe]

Syntax :SYSTem:COMMunicate:LAN:AUTOip[:STATe] {ON|OFF}

:SYSTem:COMMunicate:LAN:AUTOip[:STATe]?

Description Enable or disable the auto IP configuration mode.

Query the status of the auto IP configuration mode.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	ON

Explanation

- In auto IP configuration mode, the instrument acquires the IP address from 169.254.0.1 to 169.254.255.254 and subnet mask 255.255.0.0 according to the current network configuration automatically.
- When all the three configuration modes are set to “On”, the priority order of parameter configuration is “DHCP”, “AutoIP” and “ManualIP”.
- The three IP configuration modes cannot all be set to “Off” at the same time.

Return Format The query returns ON or OFF.

Example :SYST:COMM:LAN:AUTO ON

:SYST:COMM:LAN:AUTO?

Related Commands :SYSTem:COMMunicate:LAN:DHCP[:STATe]

:SYSTem:COMMunicate:LAN:MANualip[:STATe]

:SYSTem:COMMunicate:LAN:DHCP[:STATe]

Syntax :SYSTem:COMMunicate:LAN:DHCP[:STATe] {ON|OFF}

:SYSTem:COMMunicate:LAN:DHCP[:STATe]?

Description Enable or disable the DHCP mode.

Query the status of the DHCP mode.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	ON

Explanation

- In DHCP mode, the DHCP server in the current network assigns network parameters (such as the IP address) for the instrument.
- When all the three configuration modes are set to "On", the priority order of parameter configuration is "DHCP", "AutoIP" and "ManualIP".
- The three IP configuration modes cannot all be set to "Off" at the same time.

Return Format The query returns ON or OFF.

Example :SYST:COMM:LAN:DHCP ON

:SYST:COMM:LAN:DHCP?

Related Commands :SYSTem:COMMunicate:LAN:AUTOip[:STATe]

:SYSTem:COMMunicate:LAN:MANualip[:STATe]

:SYSTem:COMMunicate:LAN:DNS

Syntax :SYSTem:COMMunicate:LAN:DNS <dns>

:SYSTem:COMMunicate:LAN:DNS?

Description Set the DNS (Domain Name Service).

Query the current DNS.

Parameter

Name	Type	Range	Default
<dns>	ASCII character string	The format is nnn.nnn.nnn.nnn; the first nnn ranges from 1 to 223 (except 127), the other three range from 0 to 255	None

Explanation

- This command is only available when the manual IP configuration mode is enabled.
- You are recommended to ask your network administrator for an address available.

Return Format The query returns the DNS address, for example, 172.16.3.2.

Example :SYST:COMM:LAN:DNS 172.16.3.2

:SYST:COMM:LAN:DNS?

Related Command :SYSTem:COMMunicate:LAN:MANualip[:STATe]

:SYSTem:COMMunicate:LAN:GATEway

Syntax :SYSTem:COMMunicate:LAN:GATEway <gateway>

:SYSTem:COMMunicate:LAN:GATEway?

Description Set the default gateway.

Query the current default gateway.

Parameter

Name	Type	Range	Default
<gateway>	ASCII character string	The format is nnn.nnn.nnn.nnn; the first nnn ranges from 1 to 223 (except 127), the other three range from 0 to 255	None

Explanation

- ➢ This command is only available when the manual IP configuration mode is enabled.
- ➢ You are recommended to ask your network administrator for a gateway address available.

Return Format The query returns the default gateway, for example, 172.16.3.1.

Example :SYST:COMM:LAN:GATE 172.16.3.1

:SYST:COMM:LAN:GATE?

Related Command :SYSTem:COMMunicate:LAN:MANualip[:STATe]

:SYSTem:COMMunicate:LAN:IPADdress

Syntax :SYSTem:COMMunicate:LAN:IPADdress <ip>

:SYSTem:COMMunicate:LAN:IPADdress?

Description Set the IP address.

Query the current IP address.

Parameter

Name	Type	Range	Default
<ip>	ASCII character string	The format is nnn.nnn.nnn.nnn; the first nnn ranges from 1 to 223 (except 127), the other three range from 0 to 255	None

Explanation

- ➢ This command is only available when the manual IP configuration mode is enabled.
- ➢ You are recommended to ask your network administrator for an address available.

Return Format The query returns the IP address, for example, 172.16.3.128.

Example :SYST:COMM:LAN:IPAD 172.16.3.128

:SYST:COMM:LAN:IPAD?

Related Command :SYSTem:COMMunicate:LAN:MANualip[:STATe]

:SYSTem:COMMunicate:LAN:MAC?

Syntax :SYSTem:COMMunicate:LAN:MAC?

Description Query the MAC address.

Explanation The MAC (Media Access Control) address is also called hardware address and is used to define the location of the network device. For a power supply, the MAC address is unique and is usually used to recognize the instrument when assigning IP address for the instrument. The MAC address (48 bits, namely 6 bytes) is usually expressed in hexadecimal form, for example, 00-2A-A0-AA-E0-56.

Return Format The query returns the MAC address, for example, 00-2A-A0-AA-E0-56.

:SYSTem:COMMunicate:LAN:MANualip[:STATe]

Syntax :SYSTem:COMMunicate:LAN:MANualip[:STATe] {ON|OFF}

:SYSTem:COMMunicate:LAN:MANualip[:STATe]?

Description Enable or disable the manual IP configuration mode.

Query the status of the manual IP configuration mode.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	ON

Explanation

- In manual IP configuration mode, users define the network parameters (such as the IP address).
- When all the three configuration modes are set to “On”, the priority order of parameter configuration is “DHCP”, “AutoIP” and “ManualIP”.
- The three IP configuration modes cannot all be set to “Off” at the same time.

Return Format The query returns ON or OFF.

Related Commands :SYSTem:COMMunicate:LAN:AUTOip[:STATe]

:SYSTem:COMMunicate:LAN:DHCP[:STATe]

:SYSTem:COMMunicate:LAN:SMASK

Syntax :SYSTem:COMMunicate:LAN:SMASK <submask>

:SYSTem:COMMunicate:LAN:SMASK?

Description Set the subnet mask.

Query the current subnet mask.

Parameter

Name	Type	Range	Default
<submask>	ASCII character string	The format is nnn.nnn.nnn.nnn; wherein, the range of nnn is from 0 to 255	None

Explanation

- ➢ This command is only available when the manual IP configuration mode is enabled.
- ➢ You are recommended to ask your network administrator for a subnet mask available.

Return Format The query returns the subnet mask, for example, 255.255.255.0.

Example :SYST:COMM:LAN:SMASK 255.255.255.0

:SYST:COMM:LAN:SMASK?

Related Command :SYSTem:COMMunicate:LAN:MANualip[:STATe]

:SYSTem:COMMunicate:RS232:BAUD

Syntax :SYSTem:COMMunicate:RS232:BAUD {4800|7200|9600|14400|19200|38400|57600|115200|128000}

:SYSTem:COMMunicate:RS232:BAUD?

Description Set the baud rate of the RS232 interface and the unit is Baud.

Query the baud rate of the RS232 interface.

Parameter

Name	Type	Range	Default
{4800|7200|9600|14400|19200|38400|57600|115200|128000}	Discrete	4800|7200|9600|14400|19200|38400|57600|115200|128000	9600

Return Format The query returns the current baud rate, for example, 19200.

Example :SYST:COMM:RS232:BAUD 19200

:SYST:COMM:RS232:BAUD?

:SYSTem:COMMunicate:RS232:DATABit

Syntax :SYSTem:COMMunicate:RS232:DATABit {5|6|7|8}

:SYSTem:COMMunicate:RS232:DATABit?

Description Set the data bit of the RS232 interface.

Query the data bit of the RS232 interface.

Parameter

Name	Type	Range	Default
{5|6|7|8}	Discrete	5|6|7|8	8

Return Format The query returns 5, 6, 7 or 8.

Example :SYST:COMM:RS232:DATAB 8

:SYST:COMM:RS232:DATAB?

:SYSTem:COMMunicate:RS232:FLOWCrl

Syntax :SYSTem:COMMunicate:RS232:FLOWCrl {ON|OFF}

:SYSTem:COMMunicate:RS232:FLOWCrl?

Description Enable or disable the hardware flow control.

Query the status of the hardware flow control.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	OFF

Explanation This power supply uses RTS/CTS hardware flow control mode. The instrument monitors the status of the CTS pin. When the status is “True”, the instrument sends data; when the status is “False”, the instrument stops sending data. The instrument sets the CTS pin to “False” when the input buffer area is almost full and sets the CTS pin to “True” when the input buffer area is available again.

Return Format The query returns ON or OFF.

Example :SYST:COMM:RS232:FLOWC ON

:SYST:COMM:RS232:FLOWC?

:SYSTem:COMMunicate:RS232:PARItybit

Syntax :SYSTem:COMMunicate:RS232:PARItybit {NONE|ODD|EVEN}

:SYSTem:COMMunicate:RS232:PARItybit?

Description Set the parity mode to “None”, “Odd” or “Even”.

Query the current parity mode.

Parameter

Name	Type	Range	Default
{NONE|ODD|EVEN}	Discrete	NONE|ODD|EVEN	NONE

Return Format The query returns NONE, ODD or EVEN.

Example :SYST:COMM:RS232:PARI ODD

:SYST:COMM:RS232:PARI?

:SYSTem:COMMunicate:RS232:STOPBit

Syntax :SYSTem:COMMunicate:RS232:STOPBit {1|2}

:SYSTem:COMMunicate:RS232:STOPBit?

Description Set the stop bit to 1 or 2.

Query the current stop bit.

Parameter

Name	Type	Range	Default
{1|2}	Discrete	1|2	1

Return Format The query returns 1 or 2.

Example :SYST:COMM:RS232:STOPB 2

:SYST:COMM:RS232:STOPB?

:SYSTem:CONTrast

Syntax :SYSTem:CONTrast {<value>|MINimum|MAXimum}

:SYSTem:CONTrast? [{MINimum|MAXimum}]

Description Set the contrast of the screen.

Query the contrast of the screen.

Parameter

Name	Type	Range	Default
<value>	Integer	1 to 100	25 (factory setting)

Return Format The query returns an integer, for example, 50.

Example :SYST:CONT 50

:SYST:CONT?

:SYSTem:ERRor?

Syntax :SYSTem:ERRor?

Description Query and clear the error messages in the error queue.

Explanation Sending the *RST command will clear the error queue.

Return Format The query returns the number and content of the error message, for example, -113,"Undefined header; keyword cannot be found".

Related Command *RST

:SYSTem:LANGuage:TYPE

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

:SYSTem:LANGuage:TYPE?

Description Set the system language to English or Chinese.

Query the system language.

Parameter

Name	Type	Range	Default
{EN|CH}	Discrete	EN|CH	CH

Return Format The query returns English or Chinese.

Example :SYST:LANG:TYPE EN

:SYST:LANG:TYPE?

:SYSTem:LOCal

Syntax :SYSTem:LOCal

Description The power supply returns from remote mode to local mode.

Explanation When the instrument is in remote mode, the front panel keys are locked and ⇆ is displayed in the status bar in the user interface; when this command is sent, the instrument returns to local mode, the front panel keys are available and ⇆ in the status bar in the user interface disappears.

:SYSTem:LOCK

Syntax :SYSTem:LOCK {ON|OFF}

:SYSTem:LOCK?

Description Lock (ON) or unlock (OFF) the front panel.

Query whether the front panel is locked.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	OFF

Explanation DP800 allows users to lock the front panel keys to avoid danger caused by mis-operation. When the front panel is locked, 🔒 is displayed in the status bar in the user interface.

Return Format The query returns ON or OFF.

:SYSTem:OTP

Syntax :SYSTem:OTP {ON|OFF}

:SYSTem:OTP?

Description Enable or disable the over-temperature protection (OTP) function.

Query the status of the over-temperature protection function.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	ON

Explanation When the OTP function is enabled, the instrument turns off the output automatically when the temperature inside the instrument reaches the limit.

Return Format The query returns ON or OFF.

Example :SYST:OTP ON

:SYST:OTP?

:SYSTem:POWEron

Syntax :SYSTem:POWEron {DEFault|LAST}

:SYSTem:POWEron?

Description Set the instrument configuration to be used at power-on to “Default” or “Last”.

Query the instrument configuration to be used at power-on.

Parameter

Name	Type	Range	Default
{DEFault|LAST}	Discrete	DEFault|LAST	DEFault

Explanation LAST: the instrument uses the system configuration before the last power-off at power-on. DEFault: the instrument uses the factory values at power-on (except those parameters that will not be affected by reset; refer to Appendix B: Factory Setting).

Return Format The query returns DEFAULT or LAST.

Example :SYST:POWE LAST

:SYST:POWE?

:SYSTem:RGBBrightness

Syntax :SYSTem:RGBBrightness {<RGBbrightness>|MINimum|MAXimum}

:SYSTem:RGBBrightness? [{MINimum|MAXimum}]

Description Set the RGB brightness of the screen.

Query the RGB brightness of the screen.

Parameter

Name	Type	Range	Default
<RGBbrightness>	Integer	1 to 100	50 (factory setting)

Return Format The query returns an integer from 1 to 100, for example, 47.

Example :SYST:RGBB 47

:SYST:RGBB?

:SYSTem:SAVer

Syntax :SYSTem:SAVer {ON|OFF}

:SYSTem:SAVer?

Description Enable or disable the screen saver function.

Query the status of the screen saver function.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	OFF

Explanation When the screen saver function is enabled, the instrument will enter the screen saver mode automatically after standing by for 25 minutes and will enter the black screen state after another 12.5 minutes.

Return Format The query returns ON or OFF.

Example :SYST:SAV ON

:SYST:SAV?

:SYSTem:SELF:TEST:BOARD?

Syntax :SYSTem:SELF:TEST:BOARD? [{TOP|BOTTOM}]

Description Query the self-test results of TopBoard and BottomBoard.

Parameter

Name	Type	Range	Default
{TOP|BOTTOM}	Discrete	TOP|BOTTOM	None

Explanation TOP: query the self-test result of TopBoard; BOTTOM: query the self-test result of BottomBoard; when the parameter is omitted, the command queries the self-test results of TopBoard and BottomBoard at the same time.

Return Format The query returns PASS or FAIL. When the parameter is omitted, the query returns the self-test results of TopBoard and BottomBoard (separated by comma) at the same time.

Example :SYST:SELF:TEST:BOARD?

Related Command *TST?

:SYSTem:SELF:TEST:FAN?

Syntax :SYSTem:SELF:TEST:FAN?

Description Query the self-test result of the fan.

Return Format The query returns PASS or FAIL.

Related Command *TST?

:SYSTem:SELF:TEST:TEMP?

Syntax :SYSTem:SELF:TEST:TEMP?

Description Query the self-test result of the temperature.

Return Format The query returns the temperature value in ℃, for example, 23.67.

:TIMEr Commands

Command List:

- :TIMEr:CYCLEs
- :TIMEr:ENDState
- :TIMEr:GROUPs
- :TIMEr:PARAmeter
- :TIMEr[:STATe]
- :TIMEr:TEMPlet:CONSTruct
- :TIMEr:TEMPlet:FALLRate
- :TIMEr:TEMPlet:INTErval
- :TIMEr:TEMPlet:INVErt
- :TIMEr:TEMPlet:MAXValue
- :TIMEr:TEMPlet:MINValue
- :TIMEr:TEMPlet:OBJect
- :TIMEr:TEMPlet:PERIod
- :TIMEr:TEMPlet:POINTs
- :TIMEr:TEMPlet:RISERate
- :TIMEr:TEMPlet:SELect
- :TIMEr:TEMPlet:SYMMetry
- :TIMEr:TEMPlet:WIDTh

:TIMEr:CYCLEs

Syntax

```
:TIMEr:CYCLEs {N|I}[,<value>]
:TIMEr:CYCLEs?
```

Description

Set the number of cycles of the timer.

Query the current number of cycles of the timer.

Parameter

Name	Type	Range	Default
{N|I}	Discrete	N|I	N
<value>	Integer	1 to 99999	1

Explanation

- The number of cycles is defined as the number of times that the instrument performs timing output according to the preset voltage/current.
- The total number of groups in each timing output = the number of groups × the number of cycles; wherein, the number of groups is set by the :TIMEr:GROUPs command.

The power supply will terminate the timer function when the total number of groups of outputs is finished. At this point, the state of the power supply depends on the setting of the :TIMEr:ENDState command.

Return Format

The query returns I or N,<value>, for example, N,20.

Example

```
:TIME:CYCLE N,20
:TIME:CYCLE?
```

Related Commands

:TIMEr:ENDState

:TIMEr:GROUPs

:TIMEr:ENDState

Syntax :TIMEr:ENDState {OFF|LAST}

:TIMEr:ENDState?

Description Set the end state of the timer to "Off" or "Last".

Query the current end state of the timer.

Parameter

Name	Type	Range	Default
{OFF|LAST}	Discrete	OFF|LAST	OFF

Explanation

- The end state refers to the state of the instrument after it finishes outputting the total number of groups of voltage/current values when the number of cycles is a specified value.
- OFF: the instrument turns off the output automatically after finishing the output.
- LAST: the instrument stops at the output state of the last group after finishing the output.
- The total number of groups in each timing output = the number of groups × the number of cycles; wherein, the number of groups is set by the :TIMEr:GROUPs command and the number of cycles is set by the :TIMEr:CYCLEs command.

Return Format The query returns OFF or LAST.

Example :TIME:ENDS LAST

:TIME:ENDS?

Related Commands :TIMEr:CYCLEs

:TIMEr:GROUPs

:TIMEr:GROUPs

Syntax :TIMEr:GROUPs <value>

:TIMEr:GROUPs?

Description Set the number of output groups of the timer.

Query the current number of output groups of the timer.

Parameter

Name	Type	Range	Default
<value>	Integer	1 to 2048	1

Explanation

- The number of groups is defined as the number of groups of preset voltage/current values that the power supply outputs in each cycle.
- The total number of groups in each timing output = the number of groups × the number of cycles; wherein, the number of cycles is set by the :TIMEr:CYCLEs command.

The power supply will terminate the timer function when the total number of groups of outputs is finished. At this point, the state of the power supply depends on the setting of the :TIMEr:ENDState command.

Return Format The query returns an integer from 1 to 2048, for example, 25.

Example :TIME:GROUP 25

:TIME:GROUP?

Related Commands :TIMEr:CYCLEs

:TIMEr:ENDState

:TIMEr:PARAmeter

Syntax

:TIMEr:PARAmeter <secnum>,<volt>,<curr>,<time>

:TIMEr:PARAmeter? <firnum>,<timercount>

Description

Set the timer parameters of the specified group.

Query the timer parameters of the specified groups.

Parameter

Name	Type	Range	Default
<secnum>	Integer	0 to 2047	None
<volt>	Real	The voltage range and default value of the current channel; refer to the "**Explanation**" of :APPLy Command	
<curr>	Real	The current range and default value of the current channel; refer to the "**Explanation**" of :APPLy Command	
<time>	Real	1s to 99999s	1s
<firnum>	Integer	0 to 2047	None
<timercount>	Integer	1 to 2048	None

Explanation

- <secnum> is the group number of the timer parameters; <volt>, <curr> and <time> are the voltage, current and time of the timer parameters.
- <firnum> is the group number of the first group of timer parameters to be queried. <timercount> is the total number of groups of timer parameters to be queried.

Return Format

For example, **#90000**00038**1,8.000,1.0000,10;2,6.000,1.0000,10;** wherein, **#90000** is the data block header; 00038 represents the number of bytes following; **1,8.000,1.0000,10;2,6.000,1.0000,10;** are the timer parameters returned. The format of each group of parameters is "number,voltage,current,time" and multiple groups of return values are separated by semicolons.

Example

```
:TIME:PARA 1,8,1,10
:TIME:PARA 2,6,1,10
:TIME:PARA? 1,2   /*Query two groups of timer parameters starting from the first
                    group and the query returns
                 /*#90000000381,8.000,1.0000,10;2,6.000,1.0000,10;*/
```

:TIMEr[:STATe]

Syntax :TIMEr[:STATe] {ON|OFF}

:TIMEr[:STATe]?

Description Enable or disable the timing output function.

Query the status of the timing output function.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	OFF

Explanation

- Enabling the timer will change the output state of the channel; make sure that the change in the output state will not affect the device connected to the power supply before enabling the timer.
- The timing output is valid only when both the timer and the channel output are enabled.
- When the timer is enabled, the timer parameters cannot be modified.

Return Format The query returns ON or OFF.

Example

```
:TIME ON      /*Enable the timing output*/
:TIME?        /*Query the status of the timing output and the query returns ON*/
```

:TIMEr:TEMPlet:CONSTruct

Syntax :TIMEr:TEMPlet:CONSTruct

Description Send this command and the instrument will create the timer parameters according to the templet currently selected and the parameters set.

:TIMEr:TEMPlet:FALLRate

Syntax :TIMEr:TEMPlet:FALLRate <value>

:TIMEr:TEMPlet:FALLRate?

Description Set the fall index of ExpFall.

Query the fall index of ExpFall.

Parameter

Name	Type	Range	Default
<value>	Integer	0 to 10	0

Explanation When the templet currently selected is ExpRise or ExpFall, the timer parameters created cannot reach the maximum or minimum due to the characteristic of the exponential function. The range of the timer parameters created is related to the rise index or fall index currently set. The larger the rise index or fall index is, the larger the range of the timer parameters will be.

Return Format The query returns an integer from 0 to 10, for example, 5.

Example

```
:TIME:TEMP:FALLR 5
:TIME:TEMP:FALLR?
```

Related Command :TIMEr:TEMPlet:SELect

:TIMEr:TEMPlet:INTErval

Syntax :TIMEr:TEMPlet:INTErval <value>

:TIMEr:TEMPlet:INTErval?

Description Set the time interval.

Query the current time interval.

Parameter

Name	Type	Range	Default
<value>	Integer	1 to 99999	1

Explanation The interval refers to the time required for the instrument to output each group of timer parameters created using the templet currently selected.

Return Format The query returns an integer from 1 to 99999, for example, 15.

Example :TIME:TEMP:INTE 15

:TIME:TEMP:INTE?

Related Command :TIMEr:TEMPlet:SELect

:TIMEr:TEMPlet:INVErt

Syntax :TIMEr:TEMPlet:INVErt {ON|OFF}

:TIMEr:TEMPlet:INVErt?

Description Enable or disable the invert of the templet currently selected.

Query whether the invert of the templet currently selected is enabled.

Parameter

Name	Type	Range	Default
{ON|OFF}	Bool	ON|OFF	OFF

Explanation When the invert is enabled, the instrument will first turn the preset waveform upside down and then create timer parameters.

Return Format The query returns ON or OFF.

Example :TIME:TEMP:INVE ON

:TIME:TEMP:INVE?

Related Command :TIMEr:TEMPlet:SELect

:TIMEr:TEMPlet:MAXValue

Syntax :TIMEr:TEMPlet:MAXValue {<value>|MINimum|MAXimum}

:TIMEr:TEMPlet:MAXValue? [MINimum|MAXimum]

Description Set the maximum voltage or current of the templet currently selected.

Query the maximum voltage or current of the templet currently selected.

Parameter

Name	Type	Range	Default
<value>	Real	Voltage or current range of the channel currently selected	1

Explanation This command sets the maximum voltage or current (depend on the :TIMEr:TEMPlet:OBJect command).

Return Format The query returns the maximum voltage or current, for example, 8.400 or 5.3000.

Example :TIME:TEMP:MAXV 5

:TIME:TEMP:MAXV?

Related Commands :TIMEr:TEMPlet:OBJect

:TIMEr:TEMPlet:SELect

:TIMEr:TEMPlet:MINValue

Syntax :TIMEr:TEMPlet:MINValue {<value>|MINimum|MAXimum}

:TIMEr:TEMPlet:MINValue? [MINimum|MAXimum]

Description Set the minimum voltage or current of the templet currently selected.

Query the minimum voltage or current of the templet currently selected.

Parameter

Name	Type	Range	Default
<value>	Real	Voltage or current range of the channel currently selected	0

Explanation This command sets the minimum voltage or current (depend on the :TIMEr:TEMPlet:OBJect command).

Return Format The query returns the minimum voltage or current, for example, 1.000 or 1.0000.

Example :TIME:TEMP:MINV 1

:TIME:TEMP:MINV?

Related Commands :TIMEr:TEMPlet:OBJect

:TIMEr:TEMPlet:SELect

:TIMEr:TEMPlet:OBJect

Syntax :TIMEr:TEMPlet:OBJect {V|C}[,<value>|MINimum|MAXimum]

:TIMEr:TEMPlet:OBJect? [MINimum|MAXimum]

Description Select the editing object of the templet and set the current or voltage.

Query the editing object of the templet as well as the corresponding current or voltage.

Parameter

Name	Type	Range	Default
{V|C}	Discrete	V|C	V
<value>	Real	Voltage or current range of the channel currently selected	0

Explanation When V is selected, the editing object is set to voltage and <val> sets the current value; when C is selected, the editing object is set to current and <val> sets the voltage value.

Return Format The query returns the editing object and the voltage or current value, for example, V,2.0000.

Example :TIME:TEMP:OBJ V,2

:TIME:TEMP:OBJ?

Related Command :TIMEr:TEMPlet:SELect

:TIMEr:TEMPlet:PERIod

Syntax :TIMEr:TEMPlet:PERIod <value>

:TIMEr:TEMPlet:PERIod?

Description Set the period of Square.

Query the period of Square.

Parameter

Name	Type	Range	Default
<value>	Integer	2 to 99999	10

Return Format The query returns an integer from 2 to 99999, for example, 15.

Example :TIME:TEMP:PERI 15

:TIME:TEMP:PERI?

Related Commands :TIMEr:TEMPlet:SELect

:TIMEr:TEMPlet:WIDTh

:TIMEr:TEMPlet:POINTs

Syntax :TIMEr:TEMPlet:POINTs <value>

:TIMEr:TEMPlet:POINTs?

Description Set the total number of points.

Query the total number of points.

Parameter

Name	Type	Range	Default
<value>	Integer	10 to 2048	10

Explanation

- The total number of points refers to the number of groups of timer parameters created using the templet currently selected.
- When the total number of points (denoted by **P**) and the current number of output groups (denoted by **G**) are different, **P** groups of parameters will be created using the templet; then, the number of output groups will change to **P** automatically.

Return Format The query returns an integer from 10 to 2048, for example, 50.

Example :TIME:TEMP:POINT 50

:TIME:TEMP:POINT?

Related Command :TIMEr:TEMPlet:SELect

:TIMEr:TEMPlet:RISERate

Syntax :TIMEr:TEMPlet:RISERate <value>

:TIMEr:TEMPlet:RISERate?

Description Set the rise index of ExpRise.

Query the rise index of ExpRise.

Parameter

Name	Type	Range	Default
<value>	Integer	0 to 10	0

Explanation When the templet currently selected is ExpRise or ExpFall, the timer parameters created cannot reach the maximum or minimum due to the characteristic of the exponential function. The range of the timer parameters created is related to the rise index or fall index currently set. The larger the rise index or fall index is, the larger the range of the timer parameters will be.

Return Format The query returns an integer from 0 to 10, for example, 5.

Example :TIME:TEMP:RISER 5

:TIME:TEMP:RISER?

Related Command :TIMEr:TEMPlet:SELect

:TIMEr:TEMPlet:SELect

Syntax :TIMEr:TEMPlet:SELect {SINE|SQUARE|RAMP|UP|DN|UPDN|RISE|FALL}

:TIMEr:TEMPlet:SELect?

Description Select the desired templet.

Query the templet currently selected.

Parameter

Name	Type	Range	Default
{SINE|SQUARE|RAMP|UP|DN|UPDN|RISE|FALL}	Discrete	SINE|SQUARE|RAMP|UP|DN|UPDN|RISE|FALL	SINE

Return Format The query returns SINE, SQUARE, RAMP, UP, DN, UPDN, RISE or FALL.

Example :TIME:TEMP:SEL SQUARE

:TIME:TEMP:SEL?

:TIMEr:TEMPlet:SYMMetry

Syntax :TIMEr:TEMPlet:SYMMetry <value>

:TIMEr:TEMPlet:SYMMetry?

Description Set the symmetry of RAMP.

Query the symmetry of RAMP.

Parameter

Name	Type	Range	Default
<value>	Integer	0 to 100	50

Explanation Symmetry refers to the ratio of the duration of the rising edge within a period to the whole period.

Return Format The query returns an integer from 0 to 100, for example, 60.

Example :TIME:TEMP:SYMM 60

:TIME:TEMP:SYMM?

Related Command :TIMEr:TEMPlet:SELect

:TIMEr:TEMPlet:WIDTh

Syntax :TIMEr:TEMPlet:WIDTh <value>

:TIMEr:TEMPlet:WIDTh?

Description Set the pulse width of Square.

Query the pulse width of Square.

Parameter

Name	Type	Range	Default
<value>	Integer	1 to (period-1)	5

Explanation Pulse width refers to the duration of high level within a period.

Return Format The query returns an integer, for example, 14.

Example :TIME:TEMP:WIDT 14

:TIME:TEMP:WIDT?

Related Commands :TIMEr:TEMPlet:PERIod

:TIMEr:TEMPlet:SELect

:TRIGger Commands

Command List:

- :TRIGger:IN[:ENABle]
- :TRIGger:IN:RESPonse
- :TRIGger:IN:SENSitivity
- :TRIGger:IN:SOURce
- :TRIGger:IN:TYPE
- :TRIGger:OUT:CONDition
- :TRIGger:OUT:DUTY
- :TRIGger:OUT[:ENABle]
- :TRIGger:OUT:PERIod
- :TRIGger:OUT:POLArity
- :TRIGger:OUT:SIGNal
- :TRIGger:OUT:SOURce

:TRIGger:IN[:ENABle]

Syntax :TRIGger:IN[:ENABle] [D0|D1|D2|D3,]{ON|OFF}

:TRIGger:IN[:ENABle]? [D0|D1|D2|D3]

Description Enable or disable the trigger input function of the specified data line.

Query the status of the trigger input function of the specified data line.

Parameter

Name	Type	Range	Default
[D0|D1|D2|D3]	Discrete	D0|D1|D2|D3	D0
{ON|OFF}	Bool	ON|OFF	OFF

Explanation

- If [D0|D1|D2|D3] is omitted, the command enables the trigger input function of the data line currently selected.
- After enabling the trigger input function, the specified source under control will turn on the output, turn off the output or toggle the output state according to the setting of the :TRIGger:IN:RESPonse command when the input signal on the specified data line meets the current trigger type.

Return Format The query returns Dn,ON or Dn,OFF; wherein, n=0, 1, 2 or 3.

Example

```
:TRIG:IN D1,ON    /*Enable the trigger input function of D1*/
:TRIG:IN? D1      /*Query the status of the trigger input function of D1 and the
                    query returns D1,ON*/
```

Related Command :TRIGger:IN:RESPonse

:TRIGger:IN:RESPonse

Syntax :TRIGger:IN:RESPonse [D0|D1|D2|D3,]{ON|OFF|ALTER}

:TRIGger:IN:RESPonse? [D0|D1|D2|D3]

Description Set the output response of the trigger input of the specified data line.

Query the output response of the trigger input of the specified data line.

Parameter

Name	Type	Range	Default
[D0|D1|D2|D3]	Discrete	D0|D1|D2|D3	D0
{ON|OFF|ALTER}	Discrete	ON|OFF|ALTER	OFF

Explanation

- If [D0|D1|D2|D3] is omitted, the command sets the output response of the data line currently selected.
- OutpOpen: when the trigger condition is met, turn on the output of the channel currently selected as the source under control.
- OutpClose: when the trigger condition is met, turn off the output of the channel currently selected as the source under control.
- OutpToggle: when the trigger condition is met, toggle the output state of the channel currently selected as the source under control.

Return Format The query returns ON, OFF or ALTER.

Example

```
:TRIG:IN:RESP D1,ON   /*Set the output response of the trigger input of D1 to
                        OutpOpen */
:TRIG:IN:RESP? D1     /*Query the output response of the trigger input of D1 and
                        the query returns ON*/
```

:TRIGger:IN:SENSitivity

Syntax :TRIGger:IN:SENSitivity [D0|D1|D2|D3,]{LOW|MID|HIGH}

:TRIGger:IN:SENSitivity? [D0|D1|D2|D3]

Description Set the trigger sensitivity of the trigger input of the specified data line.

Query the trigger sensitivity of the trigger input of the specified data line.

Parameter

Name	Type	Range	Default
[D0|D1|D2|D3]	Discrete	D0|D1|D2|D3	D0
{LOW|MID|HIGH}	Discrete	LOW|MID|HIGH	LOW

Explanation

- If [D0|D1|D2|D3] is omitted, the command sets the trigger sensitivity of the data line currently selected.
- Selecting relatively lower trigger sensitivity can avoid mis-trigger at the noise.

Return Format The query returns LOW, MID or HIGH.

Example

```
:TRIG:IN:SENS D1,HIGH   /*Set the trigger sensitivity of the trigger input of D1 to
                          high*/
:TRIG:IN:SENS? D1       /*Query the trigger sensitivity of the trigger input of D1
                          and the query returns HIGH*/
```

:TRIGger:IN:SOURce

Syntax :TRIGger:IN:SOURce [D0|D1|D2|D3,][CH1[,CH2[,CH3]]]

:TRIGger:IN:SOURce? [D0|D1|D2|D3]

Description Set the source under control of the trigger input of the specified data line.

Query the source under control of the trigger input of the specified data line.

Parameter

Name	Type	Range	Default
[D0|D1|D2|D3]	Discrete	D0|D1|D2|D3	D0
[CH1[,CH2[,CH3]]]	ASCII character string	One or more of CH1|CH2|CH3	CH1

Explanation

- If [D0|D1|D2|D3] is omitted, the command sets the source under control of the data line currently selected.
- One or more channels can be selected as the source under control at the same time.

Return Format The query returns the source under control. Multiple sources under control are separated by commas, for example, CH1 or CH1,CH2.

Example

```
:TRIG:IN:SOUR D1,CH1   /*Set the source under control of the trigger input of D1
                         to CH1*/
:TRIG:IN:SOUR? D1      /*Query the source under control of the trigger input of
                         D1 and the query returns CH1*/
```

:TRIGger:IN:TYPE

Syntax :TRIGger:IN:TYPE [D0|D1|D2|D3,]{RISE|FALL|HIGH|LOW}

:TRIGger:IN:TYPE? [D0|D1|D2|D3]

Description Set the trigger type of the trigger input of the specified data line.

Query the trigger type of the trigger input of the specified data line.

Parameter

Name	Type	Range	Default
[D0|D1|D2|D3]	Discrete	D0|D1|D2|D3	D0
{RISE|FALL|HIGH|LOW}	Discrete	RISE|FALL|HIGH|LOW	RISE

Explanation

- If [D0|D1|D2|D3] is omitted, the command sets the trigger type of the data line currently selected.
- You can select to trigger on the rising edge (RISE), falling edge (FALL), high level (HIGH) or low level (LOW) of the input signal.

Return Format The query returns RISE, FALL, HIGH or LOW.

Example

```
:TRIG:IN:TYPE D1,FALL   /*Set the trigger type of the trigger input of D1 to the
                          falling edge */
:TRIG:IN:TYPE? D1       /*Query the trigger type of the trigger input of D1 and
                          the query returns FALL*/
```

:TRIGger:OUT:CONDition

Syntax :TRIGger:OUT:CONDition [D0|D1|D2|D3,]{OUTOFF|OUTON|>V|<V|=V|>C|<C|=C|>P|<P|=P|AUTO}[,<value>|MINimum|MAXimum]

:TRIGger:OUT:CONDition? [D0|D1|D2|D3][,MINimum|MAXimum]

Description Set the trigger condition of the trigger output of the specified data line.

Query the trigger condition of the trigger output of the specified data line.

Parameter

Name	Type	Range	Default
[D0|D1|D2|D3]	Discrete	D0|D1|D2|D3	D0
{OUTOFF|OUTON|>V|<V|=V|>C|<C|=C|>P|<P|=P|AUTO}	Discrete	OUTOFF|OUTON|>V|<V|=V|>C|<C|=C|>P|<P|=P|AUTO	OUTOFF
<value>	Real	Voltage/current/power range of the channel currently selected	0.5*maximum voltage/current/power of the channel currently selected

Explanation

- If [D0|D1|D2|D3] is omitted, the command sets the trigger condition of the trigger output of the data line currently selected.
- When OUTOFF, OUTON or AUTO is selected, <value> is omitted; when >V, <V, =V, >C, <C, =C, >P, <P or =P is selected, <value> is the corresponding voltage, current or power.

Return Format When the condition is set to OUTOFF, OUTON or AUTO, the query returns OUTOFF, OUTON or AUTO; when the condition is set to >V, <V, =V, >C, <C, =C, >P, <P or =P, the query returns the condition and the voltage/current/power, for example, >V,8.800.

Example

```
:TRIG:OUT:COND D1,>V,8.8   /*Set the trigger condition of the trigger output of
                              D1 to >V and set the voltage to 8.8V*/
:TRIG:OUT:COND? D1         /* Query the trigger condition of the trigger output of
                              D1 and the query returns >V,8.800*/
```

:TRIGger:OUT:DUTY

Syntax :TRIGger:OUT:DUTY [D0|D1|D2|D3,]<value>

:TRIGger:OUT:DUTY? [D0|D1|D2|D3]

Description Set the duty cycle of the square waveform of the trigger output on the specified data line.

Query the duty cycle of the square waveform of the trigger output on the specified data line.

Parameter

Name	Type	Range	Default
[D0|D1|D2|D3]	Discrete	D0|D1|D2|D3	D0
<value>	Integer	10 to 90	50

Explanation

- If [D0|D1|D2|D3] is omitted, the command sets the duty cycle of the square waveform of the trigger output on the data line currently selected.
- Duty cycle is defined as the percentage that the high level takes up within a whole square waveform period.

Return Format The query returns an integer from 10 to 90.

Example

```
:TRIG:OUT:DUTY D1,60   /*Set the duty cycle of the square waveform of the
                         trigger output on D1 to 60%*/
:TRIG:OUT:DUTY? D1     /*Query the duty cycle of the square waveform of the
                         trigger output on D1 and the query returns 60*/
```

:TRIGger:OUT[:ENABle]

Syntax :TRIGger:OUT[:ENABle] [D0|D1|D2|D3,]{ON|OFF}

:TRIGger:OUT[:ENABle]? [D0|D1|D2|D3]

Description Enable or disable the trigger output function of the specified data line.

Query the status of the trigger output function of the specified data line.

Parameter

Name	Type	Range	Default
[D0|D1|D2|D3]	Discrete	D0|D1|D2|D3	D0
{ON|OFF}	Bool	ON|OFF	OFF

Explanation

- If [D0|D1|D2|D3] is omitted, the command enables or disables the trigger output function of the data line currently selected.
- When the trigger output function is enabled, the specified data line outputs level or square waveform according to the setting of the output signal when the output signal of the control source meets the trigger condition set.

Return Format The query returns Dn,ON or Dn,OFF; wherein, n=0, 1, 2 or 3.

Example

```
:TRIG:IN D1,ON   /*Enable the trigger output function of D1 */
:TRIG:IN? D1     /*Query the status of the trigger output function of D1 and the
                   query returns D1,ON*/
```

:TRIGger:OUT:PERIod

Syntax :TRIGger:OUT:PERIod [D0|D1|D2|D3,]<value>

:TRIGger:OUT:PERIod? [D0|D1|D2|D3]

Description Set the period of the square waveform of the trigger output on the specified data line.

Query the period of the square waveform of the trigger output on the specified data line.

Parameter

Name	Type	Range	Default
[D0|D1|D2|D3]	Discrete	D0|D1|D2|D3	D0
<value>	Real	0.0001 to 2.5	1

Explanation If [D0|D1|D2|D3] is omitted, the command sets the period of the square waveform of the trigger output on the data line currently selected.

Return Format The query returns a value from 0.0001 to 2.5.

Example

```
:TRIG:OUT:PERI D1,0.005   /*Set the period of the square waveform of the trigger
                           output on D1 to 5ms*/
:TRIG:OUT:PERI? D1     /*Query the period of the square waveform of the trigger
                        output on D1 and the query returns 0.005000*/
```

:TRIGger:OUT:POLArity

Syntax :TRIGger:OUT:POLArity [D0|D1|D2|D3,]{POSItive|NEGAtive}

:TRIGger:OUT:POLArity? [D0|D1|D2|D3]

Description Set the polarity of the trigger output signal of the specified data line.

Query the polarity of the trigger output signal of the specified data line.

Parameter

Name	Type	Range	Default
[D0|D1|D2|D3]	Discrete	D0|D1|D2|D3	D0
{POSItive|NEGAtive}	Discrete	POSItive|NEGAtive	POSItive

Explanation

- If [D0|D1|D2|D3] is omitted, the command sets the polarity of the trigger output signal of the data line currently selected.
- POSItive: output according to the signal selected using the :TRIGger:OUT:SIGNal command;
  NEGAtive: turn the signal selected using the :TRIGger:OUT:SIGNal command upside down and then output the signal.

Return Format The query returns POSITIVE or NEGATIVE.

Example

```
:TRIG:OUT:POLA D1,NEGAtive   /* Set the polarity of the trigger output signal of
                               D1 to negative */
:TRIG:OUT:POLA? D1           /* Query the polarity of the trigger output signal of
                               D1 and the query returns NEGATIVE*/
```

:TRIGger:OUT:SIGNal

Syntax :TRIGger:OUT:SIGNal [D0|D1|D2|D3,]{LEVEL|SQUARE}

:TRIGger:OUT:SIGNal? [D0|D1|D2|D3]

Description Set the type of the trigger output signal of the specified data line.

Query the type of the trigger output signal of the specified data line.

Parameter

Name	Type	Range	Default
[D0|D1|D2|D3]	Discrete	D0|D1|D2|D3	D0
{LEVEL|SQUARE}	Discrete	LEVEL|SQUARE	LEVEL

Explanation

- If [D0|D1|D2|D3] is omitted, the command sets the type of the trigger output signal of the data line currently selected.
- When LEVEL is selected, the corresponding data line outputs level signal when the trigger condition is met; when SQUARE is selected, the corresponding data line outputs square waveform when the trigger condition is met.

Return Format The query returns LEVEL or SQUARE.

Example

```
:TRIG:OUT:SIGN D1,LEVEL   /* Set the type of the trigger output signal of D1 to
                             level */
:TRIG:OUT:SIGN? D1        /* Query the type of the trigger output signal of D1
                             and the query returns LEVEL*/
```

:TRIGger:OUT:SOURce

Syntax :TRIGger:OUT:SOURce [D0|D1|D2|D3,]{CH1|CH2|CH3}

:TRIGger:OUT:SOURce? [D0|D1|D2|D3]

Description Set the control source of the trigger output of the specified data line.

Query the control source of the trigger output of the specified data line.

Parameter

Name	Type	Range	Default
[D0|D1|D2|D3]	Discrete	D0|D1|D2|D3	D0
{CH1|CH2|CH3}	Discrete	CH1|CH2|CH3	CH1

Explanation

- If [D0|D1|D2|D3] is omitted, the command sets the control source of the trigger output of the data line currently selected.
- Any channel can be selected as the control source of the trigger output.

Return Format The query returns the control source, for example, CH1.

Example

```
:TRIG:IN:SOUR D1,CH1   /* Set the control source of D1 to CH1*/
:TRIG:IN:SOUR? D1      /* Query the control source of D1 and the query returns
                          CH1*/
```

Chapter 3 Application Examples

This chapter provides some application examples of the SCPI commands. A series of SCPI commands are combined to realize the main functions of the power supply.

Note:

1 The examples in this chapter are based on DP831A. For other models, the ranges of some parameters might be different. When using the commands, please make proper adjustment according to the model of your instrument.

2 Before using the examples in this chapter, please select the desired communication interface (USB, LAN, RS232 or GPIB) and make correct connections (refer to the introductions in To Build Remote Communication). Besides, you have to install Ultra Sigma or other PC software for sending commands on your PC.

3 The content enclosed in "/*" and "*/" after each command is annotation for easier understanding and is not a part of the command.

Main topics of this chapter:

CV Output

Requirement

Use the SCPI commands to realize the following functions:
CH1 CV output; set the output voltage to 5V, the output current to 5A and the overcurrent protection limit to 5.3A.

Method 1

```
1    *IDN?                  /*Query the ID character string of the power supply to check whether the
                              remote communication is normal*/
2    :INST CH1              /*Select CH1*/
3    :CURR 5                /*Set the current to 5A*/
4    :CURR:PROT 5.3         /*Set the overcurrent protection limit of CH1 to 5.3A*/
5    :CURR:PROT:STAT ON     /*Enable the overcurrent protection function of CH1*/
6    :VOLT 5                /*Set the voltage to 5V*/
7    :OUTP CH1,ON           /*Enable the output of CH1*/
```

Method 2

```
1    *IDN?                  /*Query the ID character string of the power supply to check whether the
                              remote communication is normal*/
2    :CURR:PROT 5.3         /*Set the overcurrent protection limit of CH1 to 5.3A*/
3    :CURR:PROT:STAT ON     /*Enable the overcurrent protection function of CH1*/
4    :APPL CH1,5,5          /*Select CH1, set the voltage to 5V and current to 5A*/
5    :OUTP CH1,ON           /*Enable the output of CH1*/
```

Track Function

Some channels of DP800 support the track function, including CH2 and CH3 of DP831A as well as CH1 and CH2 of DP832A/DP832.

Requirement

Use the SCPI commands to realize the following functions by taking DP831A as an example:
Enable the track function of CH3; change the voltage setting value of CH3 from -5V to -30V; at this point, the voltage setting value of CH2 changes accordingly.

Method

```
1    *IDN?                  /*Query the ID character string of the power supply to check whether the
                              remote communication is normal*/
2    :OUTP:TRAC CH3,ON      /*Enable the track function of CH3*/
3    :APPL CH3,-5,1         /*Set the voltage of CH3 to -5V and the current to 1A*/
4    :APPLy? CH2,VOLTage    /*Query the voltage of CH2 and the query returns 5.000*/
5    :APPL CH3,-30,1        /*Change the voltage of CH3 to -30V*/
6    :APPLy? CH2,VOLTage    /*Query the voltage of CH2 and the query returns 30.000*/
```

Timing Output

Requirement

Use the SCPI commands to realize the following functions:

- Set the timer parameters of CH1: set the number of groups to 25, the number of cycles to 20 and the end state to last; use the Sine templet to create the timer parameters; set the editing object to voltage and the current to 2A; set the templet maximum to 8V and the templet minimum to 0V; set the total number of points to 25 and the time interval to 5s; enable the invert.
- Save the timer parameters edited.
- Enable the timing output.

Method

```
1    *IDN?                    /*Query the ID character string of the power supply to check whether the
                                remote communication is normal*/
2    :INST CH1                /*Select CH1*/
3    :TIME:GROUP 25           /*Set the number of groups to 25*/
4    :TIME:CYCLE N,20         /*Set the number of cycles to 20*/
5    :TIME:ENDS LAST          /*Set the end state to last*/
6    :TIME:TEMP:SEL SINE      /*Select Sine templet*/
7    :TIME:TEMP:OBJ V,2       /*Set the editing object to voltage and set the current to 2A*/
8    :TIME:TEMP:MAXV 8        /*Set the maximum to 8V*/
9    :TIME:TEMP:MINV 0        /*Set the minimum to 0V*/
10   :TIME:TEMP:POINT 25      /*Set the total number of points to 25*/
11   :TIME:TEMP:INTE 5        /*Set the time interval to 5s*/
12   :TIME:TEMP:INVE ON       /*Enable the invert*/
13   :TIME:TEMP:CONST         /*Create the timer parameters*/
14   :MEM:STOR RTF,1          /*Save the timer parameters edited in internal memory*/
15   :OUTP CH1,ON             /*Enable the output of CH1*/
16   :TIME ON                 /*Enable the timing output*/
```

Delay Output

Requirement

Use the SCPI commands to realize the following functions:

- Set the delayer parameters of CH1: set the number of groups to 25, the number of cycles to 20 and the end state to last; select 1 0 pattern to generate state; set the time generation method to monotonic increase, the time base value to 2s and the step to 5s; set the stop condition to “>Volt” and the voltage to 8V.
- Save the delayer parameters edited.
- Enable the delay output.

Method

```
1    *IDN?                      /*Query the ID character string of the power supply to check whether
                                  the remote communication is normal*/
2    :INST CH1                  /*Select CH1*/
3    :DELAY:GROUP 25            /*Set the number of groups to 25*/
4    :DELAY:CYCLE N,20          /*Set the number of cycles to 20*/
5    :DELAY:ENDS LAST           /*Set the end state to last*/
6    :DELAY:STAT:GEN 10P        /*Select 1 0 pattern to generate state*/
7    :DELAY:TIME:GEN INC,2,5    /*set the time generation method to monotonic increase, the time base
                                  value to 2s and the step to 5s */
8    :DELAY:STOP >V,8           /*Set the stop condition to “>Volt” and the voltage to 8V*/
9    :MEM:STOR RDF,1            /*Save the delayer parameters edited in internal memory*/
10   :OUTP CH1,ON               /*Enable the output of CH1*/
11   :DELAY ON                  /*Enable the delay output*/
```

To Use the Recorder

Requirement

Use the SCPI commands to realize the following functions:

Set the record period to 2s and the storage directory of the record file to C:\REC 1:RIGOL.ROF; enable the recorder, wait for about 2 minutes and disable the recorder.

Method

```
1    *IDN?                      /*Query the ID character string of the power supply to check whether the
                                   remote communication is normal*/
2    :OUTP CH1,ON               /*Enable the output of CH1; otherwise, the recorded data of CH1 will be 0*/
3    :OUTP CH2,ON               /*Enable the output of CH2; otherwise, the recorded data of CH2 will be 0*/
4    :OUTP CH3,ON               /*Enable the output of CH3; otherwise, the recorded data of CH3 will be 0*/
5    :REC:PERI 2                /*Set the record period to 2s*/
6    :REC:MEM 1,RIGOL.ROF       /*Set the storage directory of the record file to C:\REC 1:RIGOL.ROF*/
7    :REC ON                    /*Enable the recorder*/
      /*Wait for about 2 minutes……*/
8    :REC OFF                   /*Disable the recorder*/
```

To Use the Analyzer

Requirement

Use the SCPI commands to realize the following functions:

- Open the C:\REC 1:RIGOL.ROF file; set the start time to 1s and the end time to 100s.
- Execute the analysis.
- Read the analysis results.

Method

```
1    *IDN?                      /*Query the ID character string of the power supply to check whether the
                                   remote communication is normal*/
2    :ANAL:MEM 1                /*Open the C:\REC 1:RIGOL.ROF file*/
3    :ANAL:STARTT 1             /*Set the start time to 1s*/
4    :ANAL:ENDT 100             /*Set the end time to 100s*/
5    :ANAL:ANAL                 /*Execute the analysis*/
6    :ANAL:RES?                 /*Read the analysis results*/
```

To Use the Monitor

Requirement

Use the SCPI commands to realize the following functions:

- Monitor CH1.
- Set the monitor condition: >Volt, >Curr or >Power; set the voltage to 5V, the current to 3A and the power to 15W; set the stop mode to OutpOff, Warning and Beeper.
- Enable the monitor.

Method

```
1    *IDN?                          /*Query the ID character string of the power supply to check whether
                                      the remote communication is normal*/
2    :INST CH1                      /*Select CH1*/
3    :MONI:VOLT:COND >V,OR          /*Set the voltage monitor condition to ">Volt" and the logic relation to
                                      "OR" */
4    :MONI:VOLT 5                   /*Set the voltage of the monitor condition to 5V*/
5    :MONI:CURR:COND >C,OR          /*Set the current monitor condition to ">Curr" and the logic relation to
                                      "OR"*/
6    :MONI:CURR 3                   /*Set the current of the monitor condition to 3A*/
7    :MONI:POWER:COND >P            /*Set the power monitor condition to ">Power"*/
8    :MONI:POWER 15                 /*Set the power of the monitor condition to 15W*/
9    :MONI:STOP OUTOFF,ON           /*Enable the "OutpOff" stop mode*/
10   :MONI:STOP WARN,ON             /*Enable the "Warning" stop mode*/
11   :MONI:STOP BEEPER,ON           /*Enable the "Beeper" stop mode*/
12   :MONI ON                       /*Enable the monitor*/
```

To Use the Trigger

Trigger Input

Requirement

Use the SCPI commands to realize the following functions:

- Set the trigger input parameters of D0: set the source under control to CH1, the trigger type to falling edge, the output response to OutpOpen and the trigger sensitivity to low.
- Enable the trigger input function of D0.

Method

```
1    *IDN?                      /*Query the ID character string of the power supply to check whether the
                                  remote communication is normal*/
2    :TRIG:IN:SOUR D0,CH1       /*Set the source under control of the trigger input of D0 to CH1*/
3    :TRIG:IN:TYPE D0,FALL      /*Set the trigger type of the trigger input of D0 to falling edge*/
4    :TRIG:IN:RESP D0,ON        /*Set the output response of the trigger input of D0 to OutpOpen*/
5    :TRIG:IN:SENS D0,LOW       /*Set the trigger sensitivity of the trigger input of D0 to low*/
6    :TRIG:IN D0,ON             /*Enable the trigger input function of D0*/
```

Trigger Output

Requirement

Use the SCPI commands to realize the following functions:

- Set the trigger output parameters of D1: set the control source to CH2, the trigger condition to "Volt>5V", the output signal to square waveform (0.5s period and 60% duty cycle) and the polarity of the signal to negative.
- Enable the trigger output function of D1.

Method

```
1    *IDN?                         /*Query the ID character string of the power supply to check whether
                                     the remote communication is normal*/
2    :TRIG:OUT:SOUR D1,CH2         /*Set the control source of the trigger output of D1 to CH2*/
3    :TRIG:OUT:COND D1,>V,5        /*Set the trigger condition of the trigger output of D1 to "Volt>5V"*/
4    :TRIG:OUT:SIGN D1,SQUARE      /*Set the trigger output signal of D1 to square waveform*/
5    :TRIG:OUT:PERI D1,0.5         /*Set the period of the square waveform of the trigger output on D1
                                     to 0.5s*/
6    :TRIG:OUT:DUTY D1,60          /*Set the duty cycle of the square waveform of the trigger output on
                                     D1 to 60%*/
7    :TRIG:OUT:POLA D1,NEGAtive    /*Set the polarity of the trigger output signal of D1 to negative*/
8    :TRIG:OUT D1,ON               /*Enable the trigger output function of D1*/
```

Chapter 4 Programming Demos

This chapter provides the demos for programming and controlling the power supply using SCPI commands under Excel, Matlab, LabVIEW, Visual Basic and Visual C++ environment on the basis of NI-VISA.

NI-VISA (National Instrument-Virtual Instrument Software Architecture) is an advanced application programming interface developed by NI (National Instrument) for communicating with various instrument buses. It can communicate with instrument in the same method regardless of the type of the instrument interface (GPIB, USB, LAN/Ethernet or RS232).

The instruments communicate with NI-VISA via various interfaces are called “resources”. The VISA descriptor (namely the resource name) is used to describe the accurate name and location of the VISA resource. If LAN interface is currently used for communicating with the instrument, the VISA descriptor is :TCPIP0::172.16.2.13::INSTR. Before programming, please acquire the correct VISA descriptor.

Main topics of this chapter:

Programming Preparations

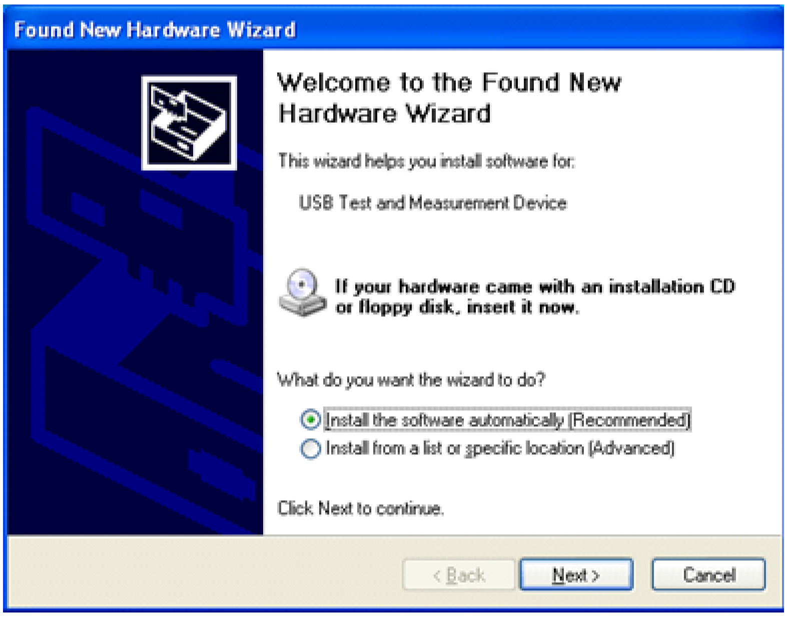
Before programming, you need to make the following preparations:

1 Make sure that your PC has installed the NI-VISA library (can be downloaded from NI website: http://www.ni.com/visa/). Here, the default installation path is C:\Program Files\IVI Foundation\VISA.

2 Here, the USB interface of the power supply is used to communicate with the PC and please use USB cable to connect the USB DEVICE interface at the rear panel of the power supply to the PC.
You can also use LAN, RS232 or GPIB to communicate with PC. Note that the end mark of the command sent through RS232 interface is "\r\n".

3 Turn on the instrument after connecting the power supply and PC.

4 At this point, the "**Found New Hardware Wizard**" dialog box appears on the PC. Please follow the instructions to install the "USB Test and Measurement Device".



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

USB Test and Measurement Device

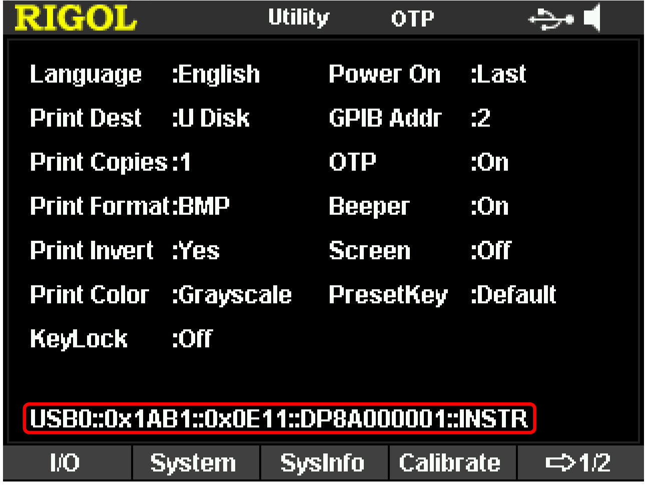
If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

● Install the software automatically (Recommended)
○ Install from a list or specific location (Advanced)

Click Next to continue.

5 Acquire the USB VISA descriptor of the power supply: press **Utility** and the VISA descriptor is displayed at the bottom of the interface, as shown in the figure below. Here, the VISA descriptor of the power supply is USB0::0x1AB1::0x0E11::DP8A000001::INSTR.



```
Language     :English     Power On   :Last
Print Dest   :U Disk      GPIB Addr  :2
Print Copies :1           OTP        :On
Print Format :BMP         Beeper     :On
Print Invert :Yes         Screen     :Off
Print Color  :Grayscale   PresetKey  :Default
KeyLock      :Off

USB0::0x1AB1::0x0E11::DP8A000001::INSTR
```

By now, the programming preparations are finished.

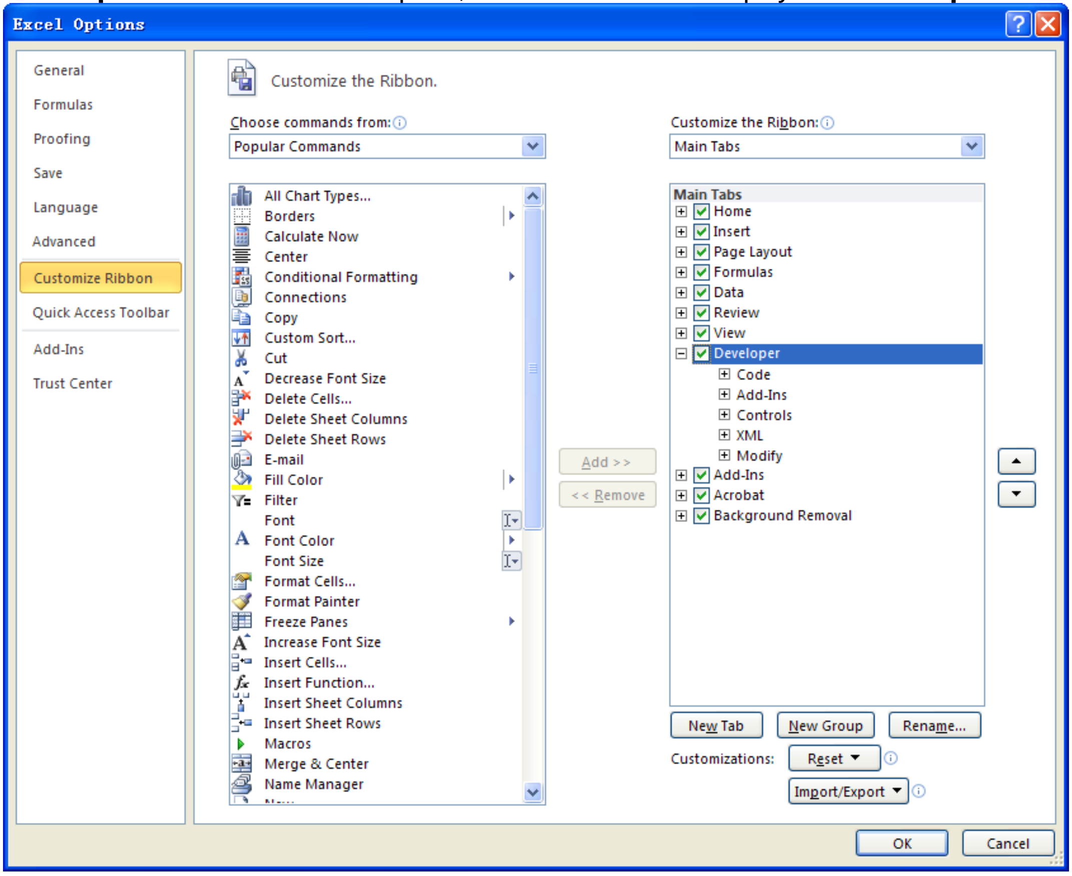
Excel Programming Demo

The program used in this demo: Microsoft Excel 2010

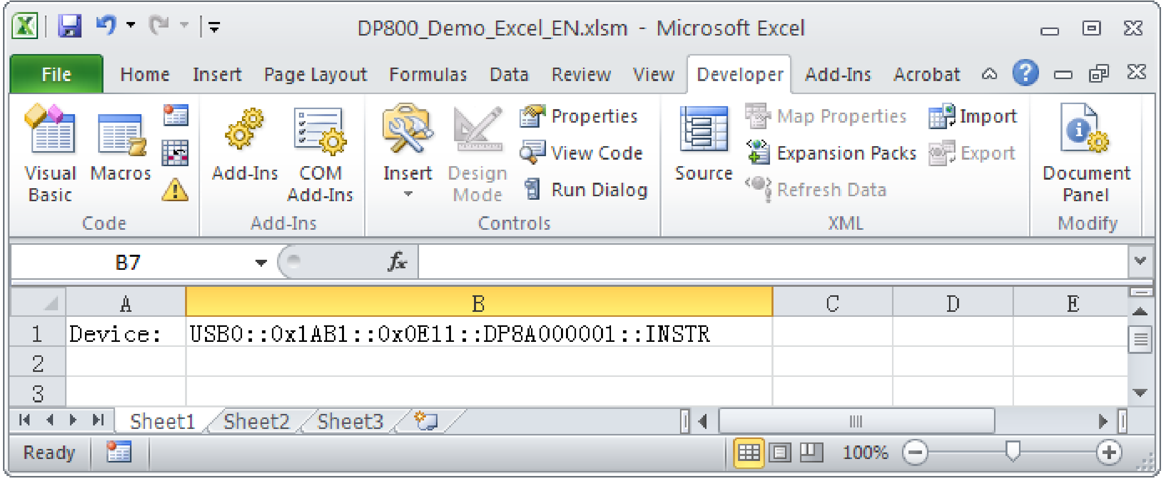
The function realized in this demo: send the *IDN? Command to read the device information.

1 Create a new Excel file that enables the Macros. In this demo, the file is named as DP800_Demo_Excel.xlsm.

2 Run the DP800_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 right, check **Developer** and click **OK**. At this point, the Excel menu bar displays the **Developer** menu.

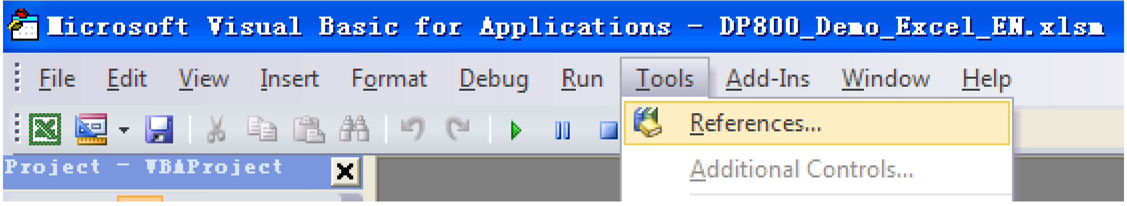


3 Enter the VISA descriptor into a cell of the file as shown in the figure below. Click the **Developer** menu and select the **Visual Basic** option to open the Microsoft Visual Basic.

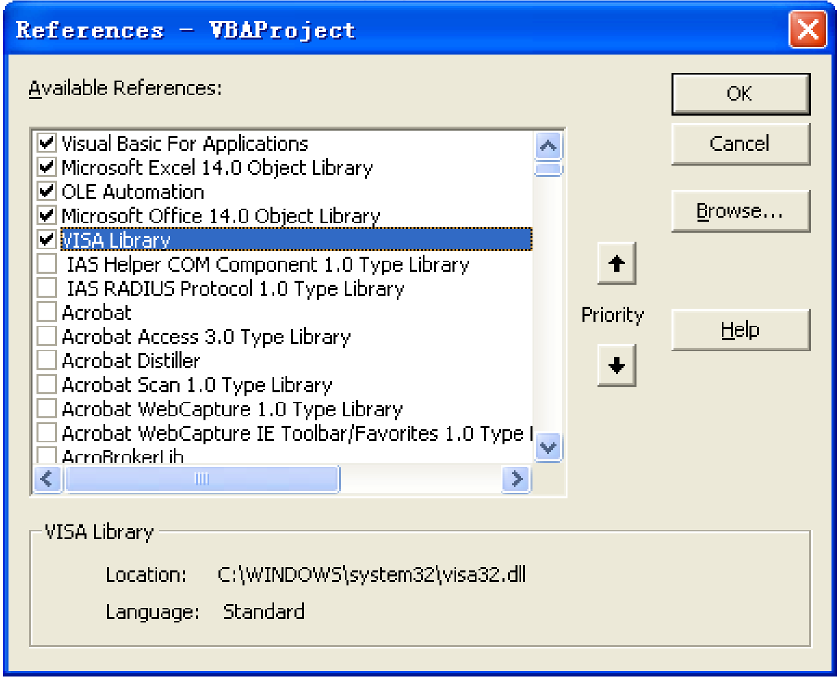


Device: USB0::0x1AB1::0x0E11::DP8A000001::INSTR

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


Microsoft Visual Basic for Applications - DP800_Demo_Excel_EN.xlsm
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
References...
Additional Controls...
Project - VBAProject

Select **VISA Library** in the pop-up dialog box and click **OK** to refer to the VISA Library.



Available References:

- ☑ Visual Basic For Applications
- ☑ Microsoft Excel 14.0 Object Library
- ☑ OLE Automation
- ☑ Microsoft Office 14.0 Object Library
- ☑ VISA Library
- □ IAS Helper COM Component 1.0 Type Library
- □ IAS RADIUS Protocol 1.0 Type Library
- □ Acrobat
- □ Acrobat Access 3.0 Type Library
- □ Acrobat Distiller
- □ Acrobat Scan 1.0 Type Library
- □ Acrobat WebCapture 1.0 Type Library
- □ Acrobat WebCapture IE Toolbar/Favorites 1.0 Type l
- □ AcroBrokerLib

VISA Library

Location: C:\WINDOWS\system32\visa32.dll

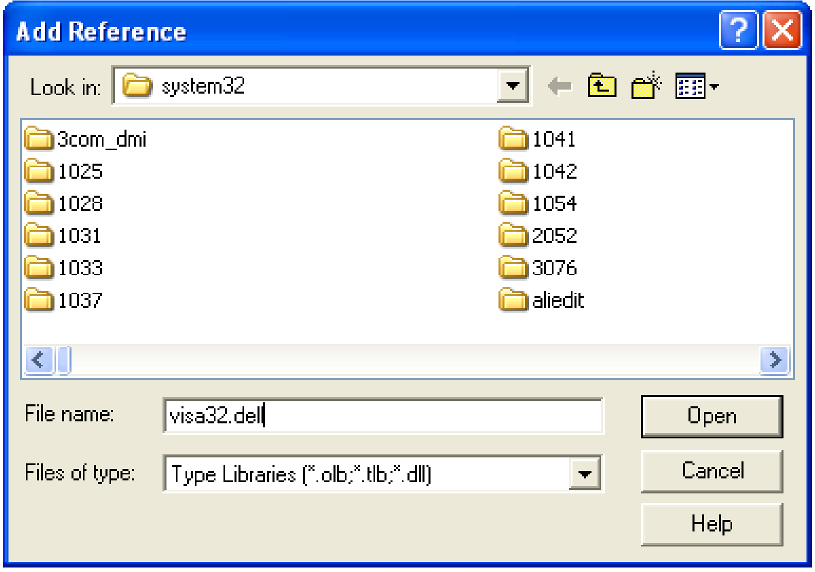
Language: Standard

Explanation:

If you cannot find VISA Library in the list at the left of the figure above, please follow the method below to find it.

(1) Make sure that your PC has installed the NI-VISA library.

(2) Click **Browse...** at the right and set the search range to **C:\WINDOWS\system32** and the filename to **visa32.dll**, as shown in the figure below.



Look in: system32

File name: visa32.dell

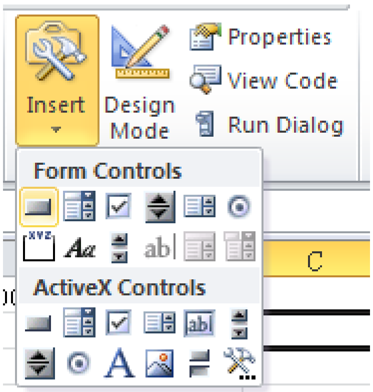
Files of type: Type Libraries (*.olb;*.tlb;*.dll)

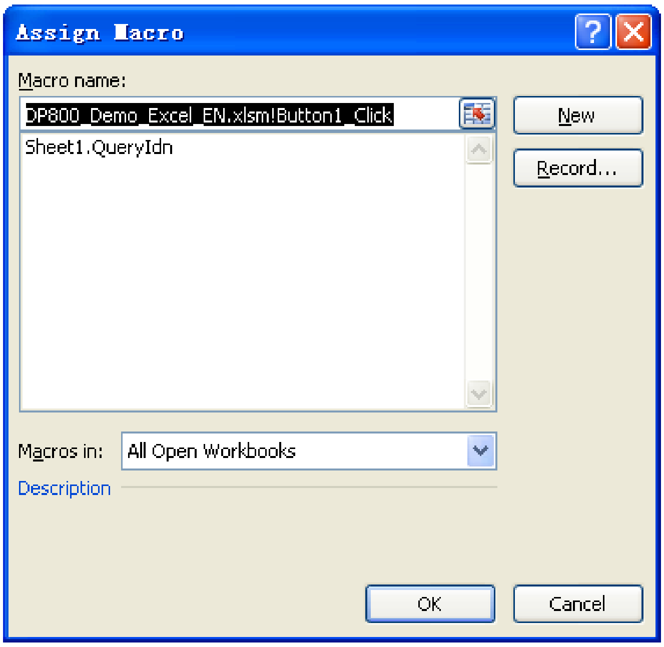
5 Click **View Code** in the **Developer** menu to enter the Microsoft Visual Basic interface. Add the following codes and save the file.

Note: If the Excel file created at step 2 does not enable the Macros, at this point, the prompt message “The following features cannot be saved in macro-free workbooks”will be displayed. In this situation, please save the Excel file as a file using the Macros.

```
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, 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, 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
```

6 Add button control: click **Insert** in the **Developer** menu, select the desired button in **Form Controls** and put it into the cell of the Excel. At this point, the **Assign Macro** interface is displayed, select “Sheet1.QueryIdn” and click “OK”.




Assign Macro

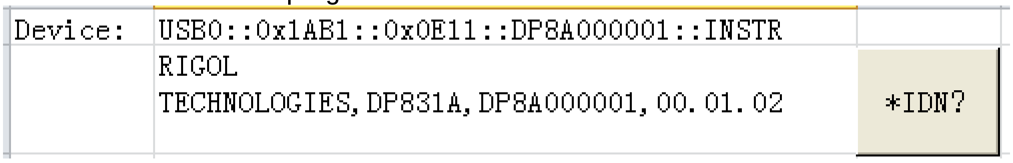
Macro name: DP800_Demo_Excel_EN.xlsm!Button1_Click

Sheet1.QueryIdn

Macros in: All Open Workbooks

By default, the button mane is “Button 1”. Right-click the button and select **Edit Text** in the pop-up menu to change the button name to “*IDN?”.

7 Click ***IDN?** to run the program.



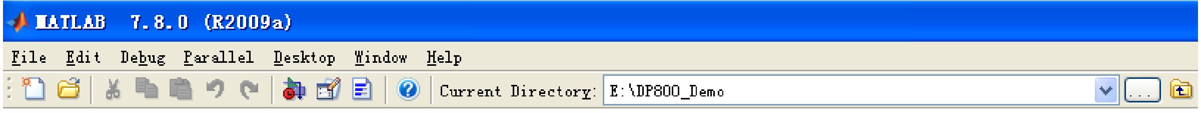
```
Device:     USB0::0x1AB1::0x0E11::DP8A000001::INSTR
            RIGOL
            TECHNOLOGIES,DP831A,DP8A000001,00.01.02
```

Matlab Programming Demo

The program used in this demo: MATLAB R2009a

The function realized in this demo: read the output voltage, current and power measured internally on CH1.

1 Run the Matlab software and modify the current directory (namely modify the **Current Directory** at the top of the software). In this demo, the current directory is modified to E:\DP800_Demo.


MATLAB 7.8.0 (R2009a)
File Edit Debug Parallel Desktop Window Help
Current Directory: E:\DP800_Demo

2 Click **File → New → Blank M-File** in the Matlab interface to create an empty M file.

3 Add the following codes in the M file:

```
dp800 = visa( 'ni','USB0::0x1AB1::0x0E11::DP8A000001::INSTR' ); %Create VISA object

fopen( dp800 );   %Open the VISA object created

fprintf(dp800, ':MEAS:ALL? CH1' );   %Send request

meas_CH1 = fscanf(dp800);   %Read data

fclose(dp800);   %Close the VISA object

display(meas_CH1)    %Display the device information read
```

4 Save the M file under the current directory. In this demo, the M file is named as DP800_Demo_MATLAB.m.

5 Run the M file and the following running result is displayed in the command window.

```
meas_CH1 =

7.9833,0.0484,0.386
```

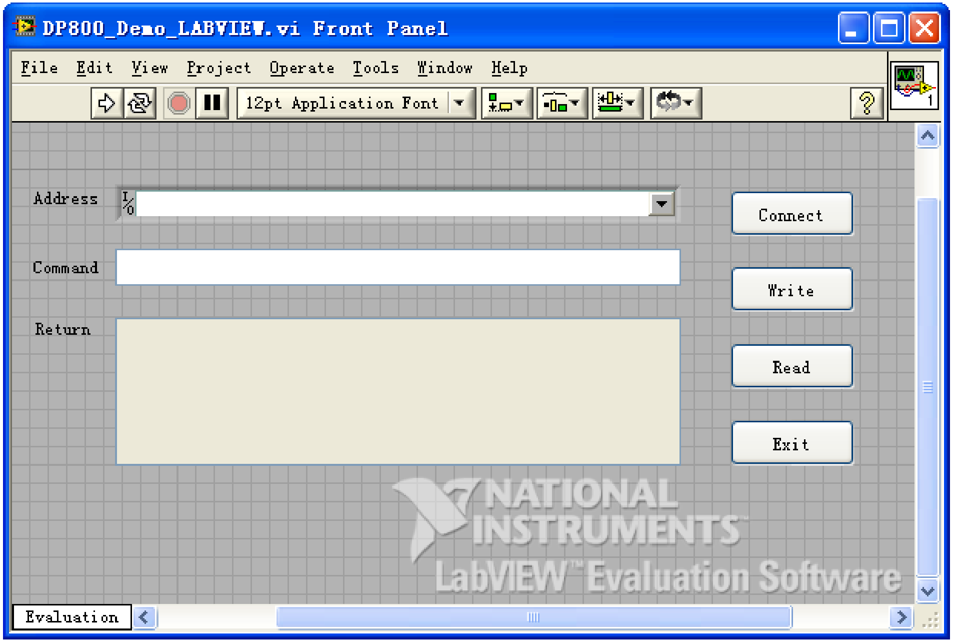
LabVIEW Programming Demo

The program used in this demo: LabVIEW 2009

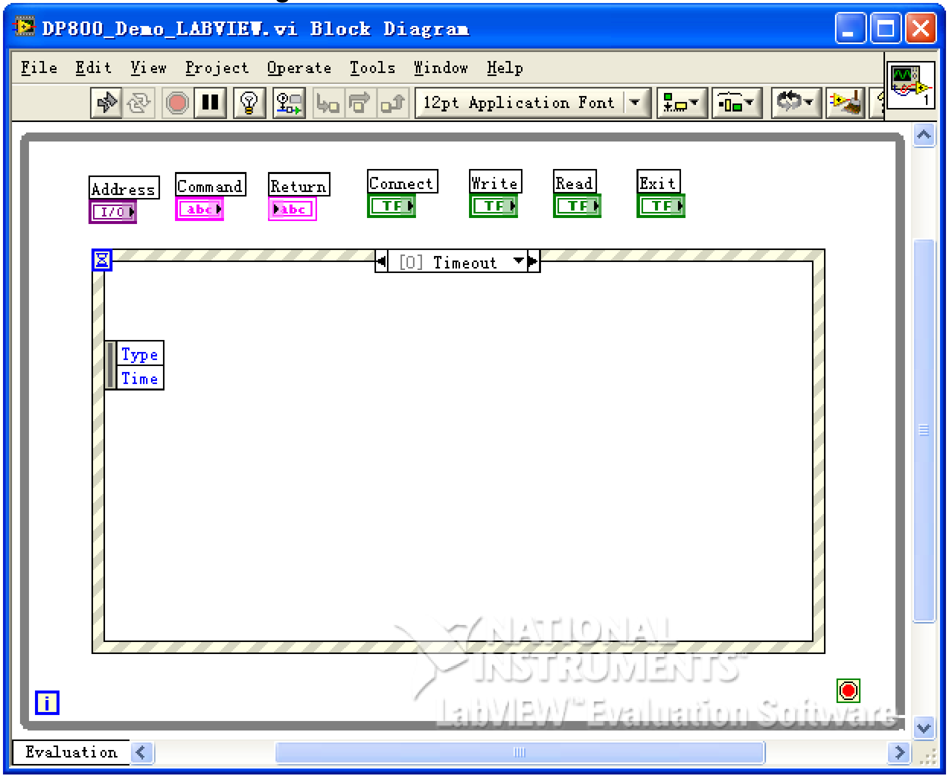
The functions realized in this demo: search for the instrument address, connect the instrument, send command and read the return value.

1 Run LabVIEW 2009, create a VI file and name it as DP800_Demo_LABVIEW.

2 Add controls in the front panel interface, including the **Address** bar, **Command** bar and **Return** bar as well as the **Connect**, **Write**, **Read** and **Exit** buttons.

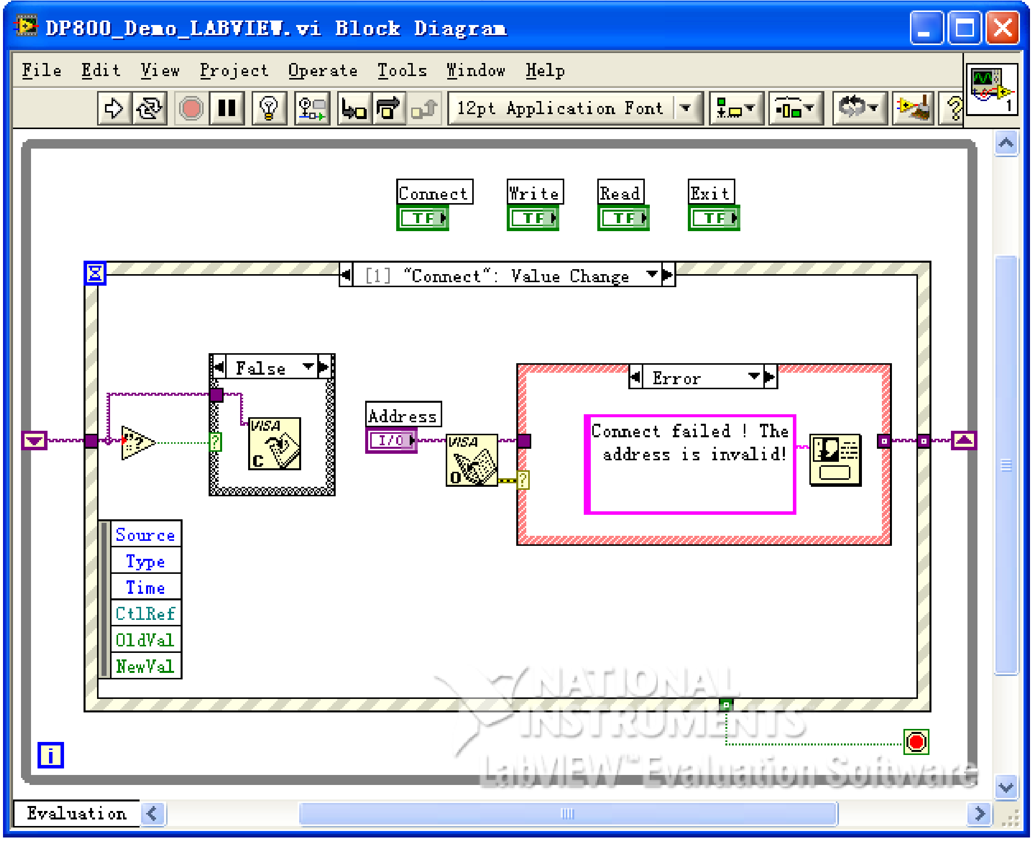

DP800_Demo_LABVIEW.vi Front Panel
File Edit View Project Operate Tools Window Help
12pt Application Font
Address
Command
Return
Connect
Write
Read
Exit
NATIONAL INSTRUMENTS
LabVIEW Evaluation Software
Evaluation

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



4 Add events, including connecting instrument, write operation, read operation and exit.

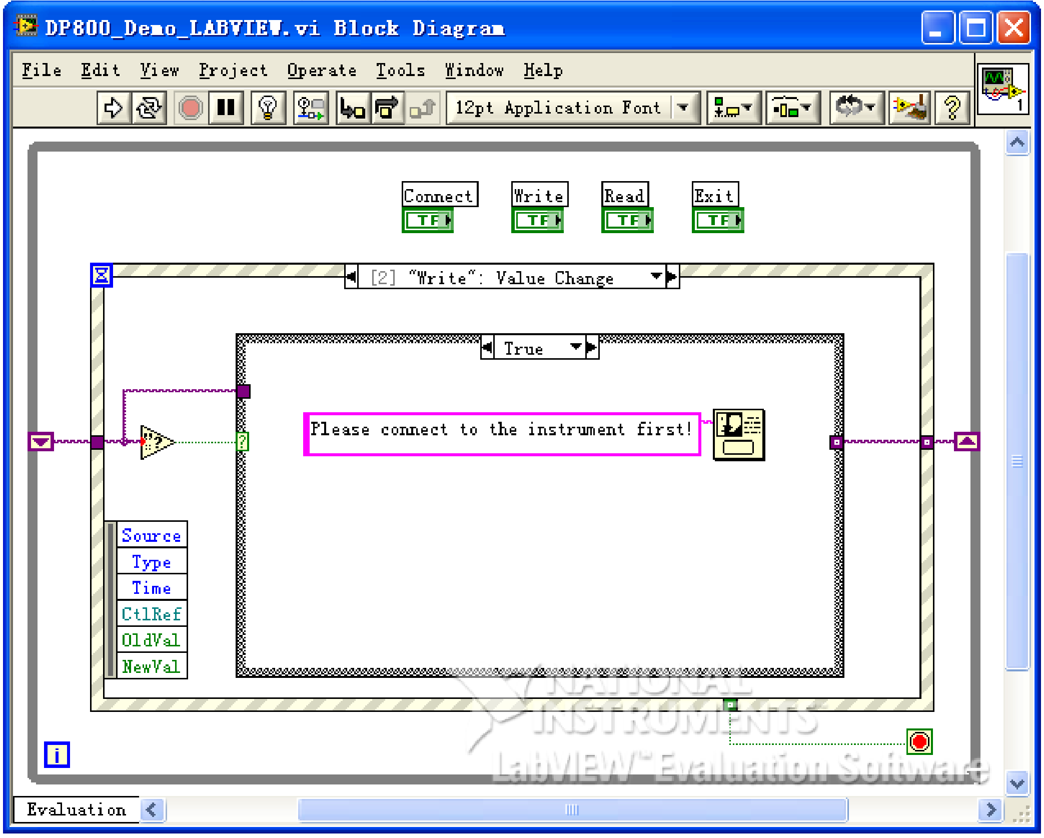
(1) Connect the instrument (including error processing):



DP800_Demo_LABVIEW.vi Block Diagram
File Edit View Project Operate Tools Window Help
12pt Application Font
Connect
Write
Read
Exit
TF
[1] "Connect": Value Change
False
VISA
Address
I/O
Error
Source
Type
Time
CtlRef
OldVal
NewVal
Evaluation
NATIONAL INSTRUMENTS LabVIEW Evaluation Software

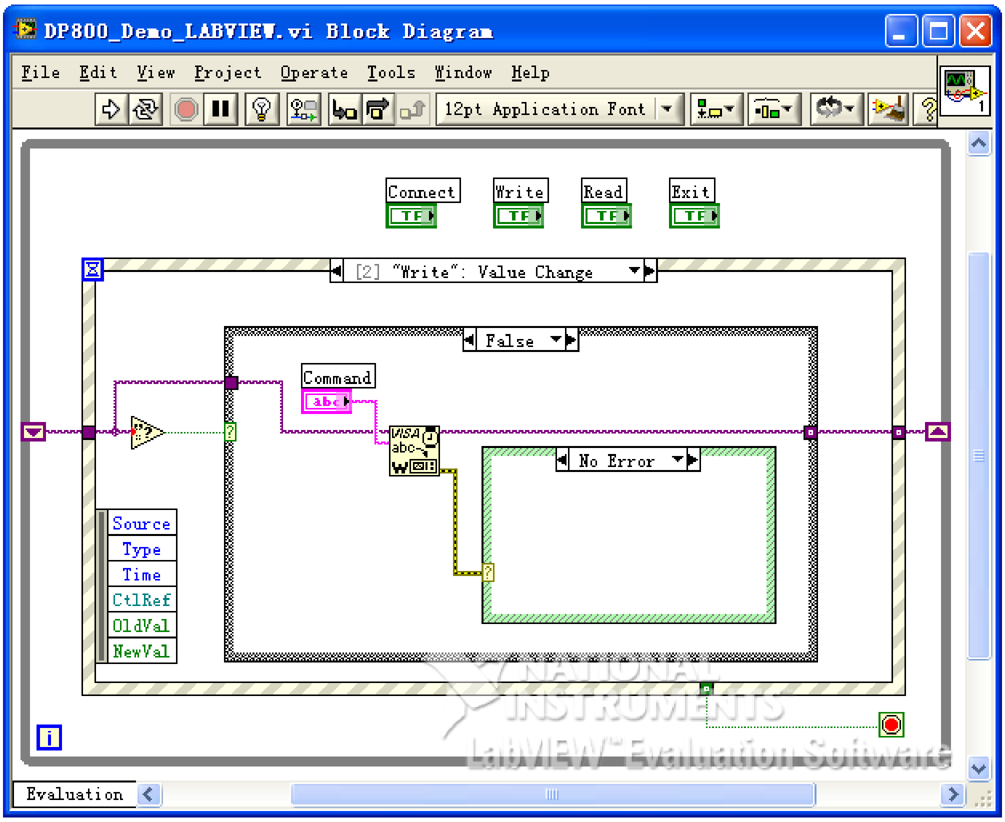
Connect failed ! The address is invalid!

(2) Write operation (including error judgment):

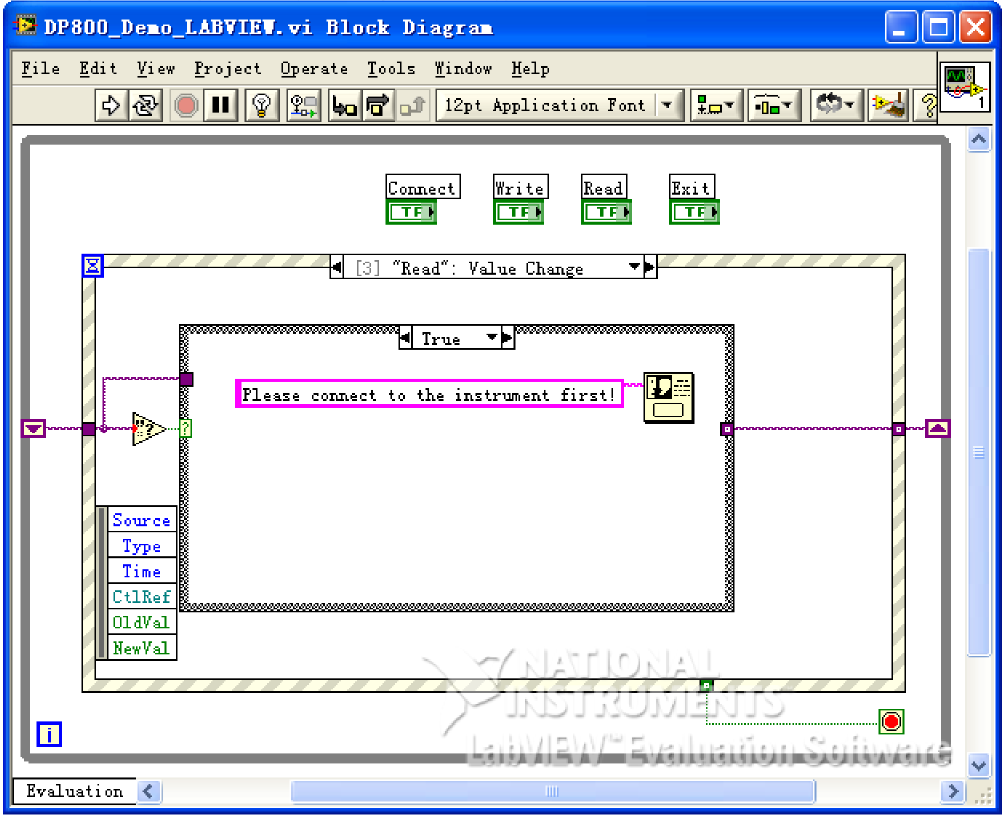


DP800_Demo_LABVIEW.vi Block Diagram
File Edit View Project Operate Tools Window Help
12pt Application Font
Connect
Write
Read
Exit
TF
[2] "Write": Value Change
True
Source
Type
Time
CtlRef
OldVal
NewVal
Evaluation
NATIONAL INSTRUMENTS LabVIEW Evaluation Software

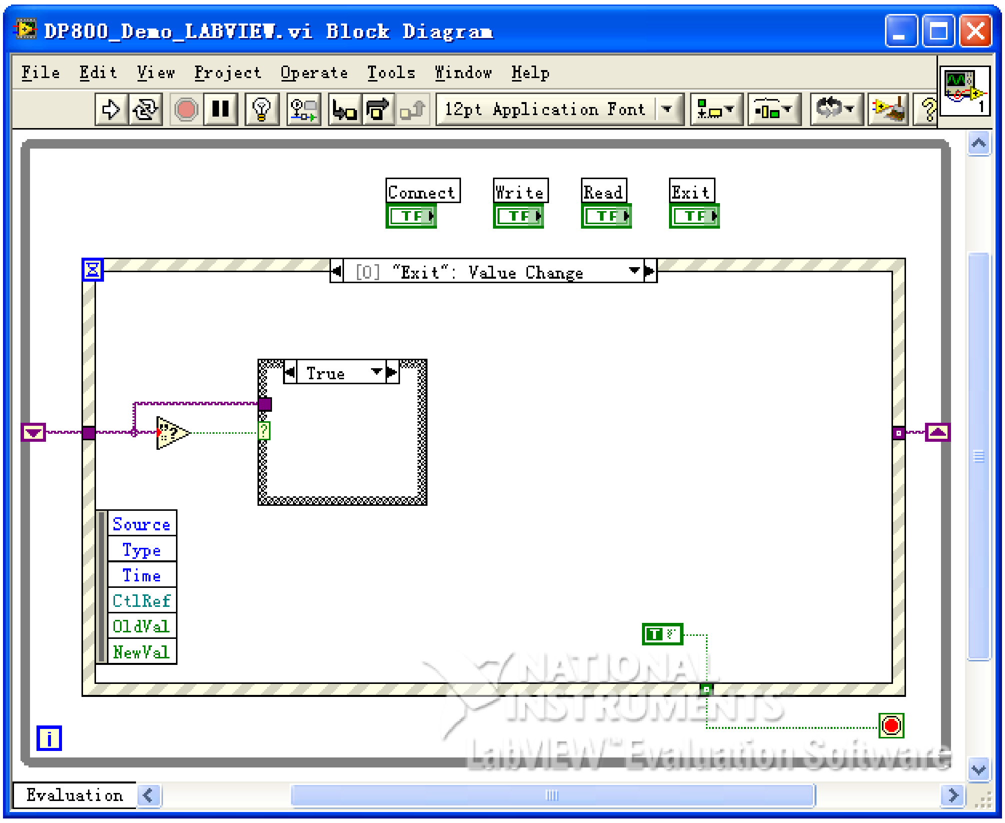
Please connect to the instrument first!


DP800_Demo_LABVIEW.vi Block Diagram
File Edit View Project Operate Tools Window Help
12pt Application Font
Connect
Write
Read
Exit
[2] "Write": Value Change
False
Command
abc
No Error
Source
Type
Time
CtlRef
OldVal
NewVal
Evaluation
NATIONAL INSTRUMENTS LabVIEW™ Evaluation Software

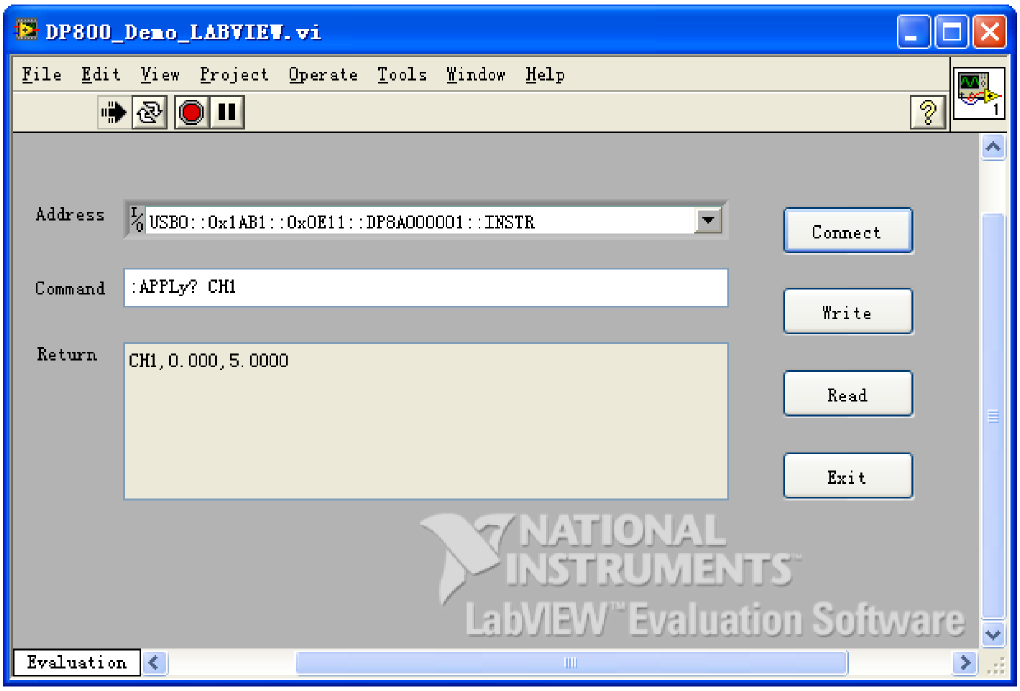
(3) Read operation (including error processing):


DP800_Demo_LABVIEW.vi Block Diagram
File Edit View Project Operate Tools Window Help
12pt Application Font
Connect
Write
Read
Exit
[3] "Read": Value Change
True
Please connect to the instrument first!
Source
Type
Time
CtlRef
OldVal
NewVal
Evaluation
NATIONAL INSTRUMENTS LabVIEW™ Evaluation Software

(4) Exit:



5 Run the program and the interface as shown in the figure below is displayed. Click the **Address** dropdown box and select the VISA resource name; click **Connect** to connect the instrument; enter the command into the **Command** textbox and click **Write** to write the command into the instrument. If the command is a query command, click **Read** and the return value is displayed in the **Return** textbox.


DP800_Demo_LABVIEW.vi
File Edit View Project Operate Tools Window Help
Connect
Write
Read
Exit
NATIONAL INSTRUMENTS
LabVIEW Evaluation Software
Evaluation

Address USB0::0x1AB1::0x0E11::DP8A000001::INSTR

Command :APPLy? CH1

Return CH1,0.000,5.0000

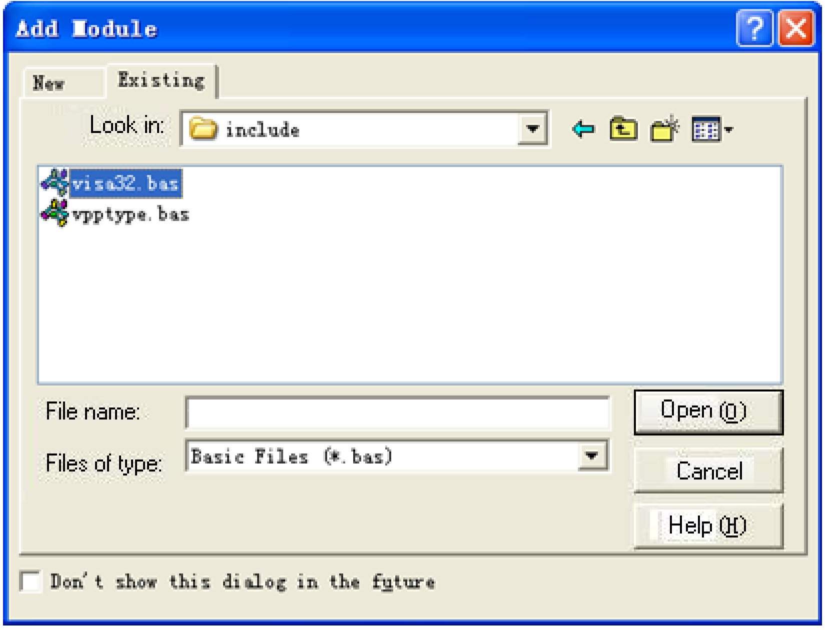
Visual Basic Programming Demo

The program used in this demo: Visual Basic 6.0

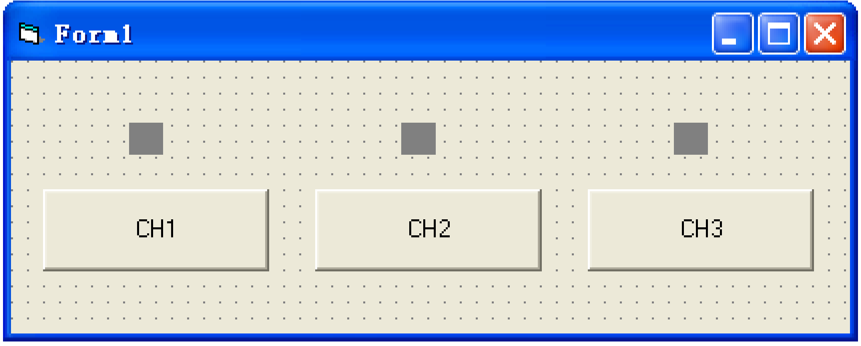
The function realized in this demo: enable the three channels of the power supply and show the colors of the channels.

1 Build a standard application program project (Standard EXE) and name it as DP800_Demo_VB.

2 Click the **Existing** tab of **Project→Add Module**. Search for the **visa32.bas** file in the **include** folder under the **NI-VISA** installation path and add the file.



3 Add three **CommandButton** controls to represent CH1, CH2 and CH3 respectively. Add three Text controls (Label1(0), Label1(1) and Label1(2)) to represent the status of the three channels respectively (by default, the Text control is gray; when the channel is enabled, it displays the color of the channel). The layout of the controls is as shown in the figure below.



4 Open the **General** tab in **Project→Project1 Properties** and select **Form1** in the **Startup Object** dropdown box.

5 Double-click **CH1** to enter the programming environment. Add the following codes to control CH1, CH2 and CH3. The codes of CH1 are as shown below; the codes of CH2 and CH3 are similar.

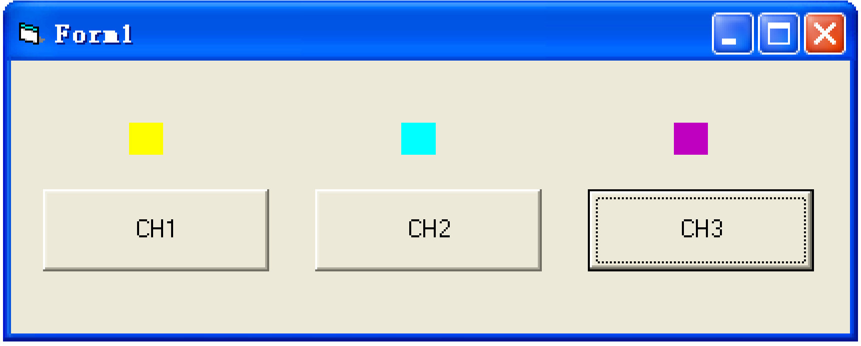
```
Dim defrm As Long
Dim vi As Long
Dim strRes As String * 200
Dim list As Long
```

```
    Dim nmatches As Long
    Dim matches As String * 200

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

    ' Turn on the device
    Call viOpen(defrm, matches, 0, 0, vi)

    ' Send command to query the CH1 status
    Call viVPrintf(vi, ":OUTP? CH1" + Chr$(10), 0)

    ' Acquire the status of CH1
    Call viVScanf(vi, "%t", strRes)

    If strRes = "ON" Then

    ' Send the setting command
    Call viVPrintf(vi, ":OUTP CH1,OFF" + Chr$(10), 0)
    Label1(0).ForeColor = &H808080   'Gray

    Else

    Call viVPrintf(vi, ":OUTP CH1,ON" + Chr$(10), 0)
    Label1(0).ForeColor = &HFFFF&   'Yellow

    End If

    ' Turn off the resource
    Call viClose(vi)
    Call viClose(defrm)
```

6 Running results

1) Click **CH1** to enable CH1 and the control above **CH1** turns yellow;
2) Click **CH2** to enable CH2 and the control above **CH2** turns blue;
3) Click **CH3** to enable CH3 and the control above **CH3** turns rosy.

The running results are as shown in the figure below.



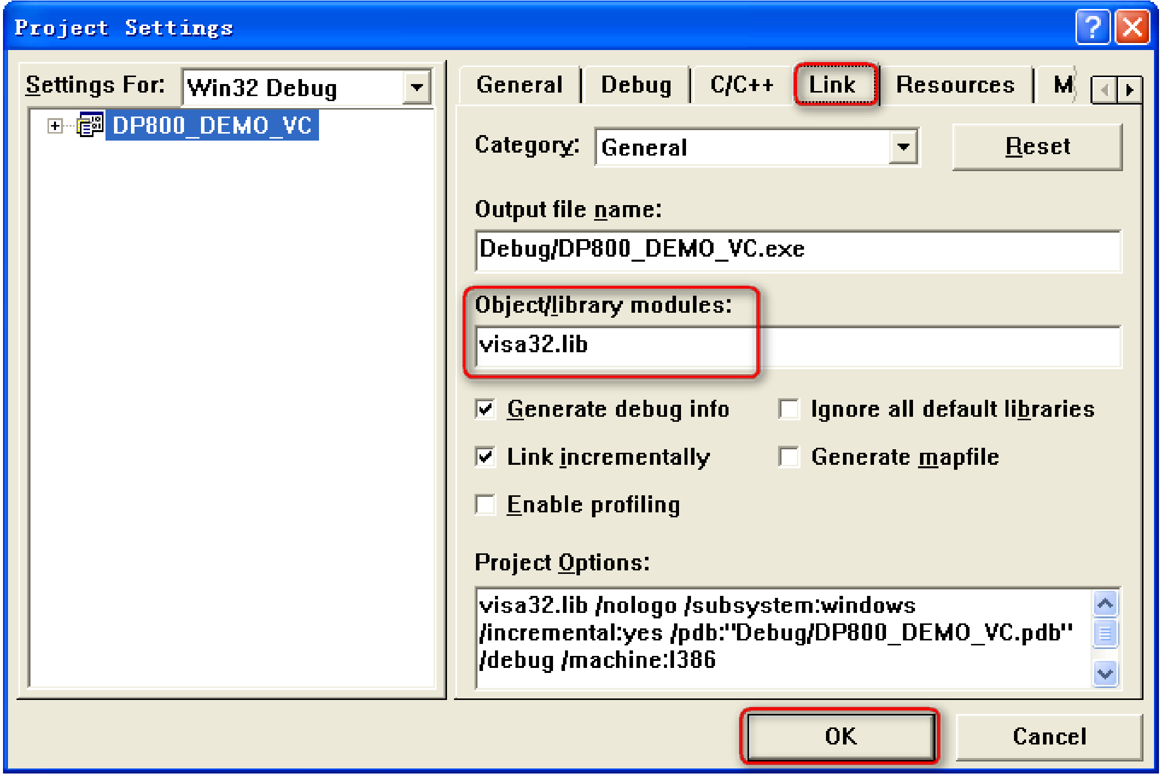
Visual C++ Programming Demo

The program used in this demo: Microsoft Visual C++ 6.0

The functions realized in this demo: search for the instrument address, connect the instrument, send command and read the return value.

1 Run Microsoft Visual C++ 6.0, create a MFC project based on dialog box and name it as DP800_Demo_VC。

2 Click **Project→Settings** and add **visa32.lib** in the **Link** tab in the pop-up interface manually.



Settings For: Win32 Debug

Category: General

Output file name: Debug/DP800_DEMO_VC.exe

Object/library modules: visa32.lib

☑ Generate debug info ☐ Ignore all default libraries

☑ Link incrementally ☐ Generate mapfile

☐ Enable profiling

Project Options:

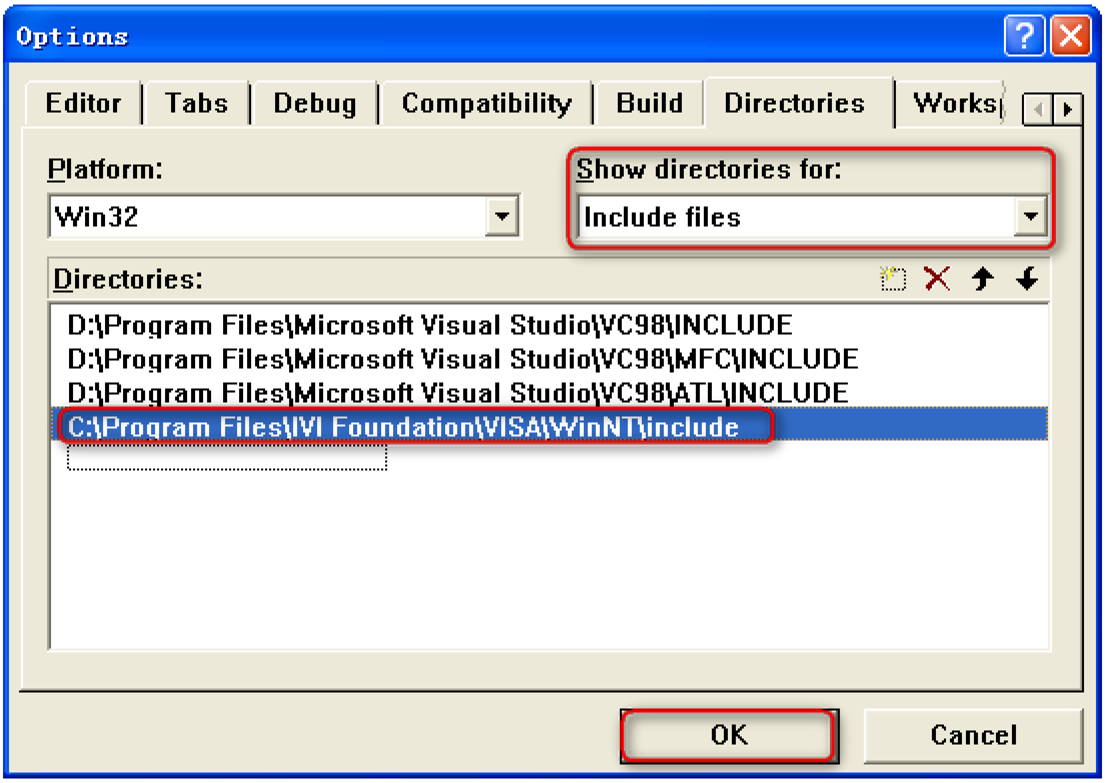
```
visa32.lib /nologo /subsystem:windows
/incremental:yes /pdb:"Debug/DP800_DEMO_VC.pdb"
/debug /machine:I386
```

3 Click **Tools→Options** and add the **Include** and **Lib** paths in the **Directories** tab in the pop-up interface.

Select **Include files** in **Show directories for** and double-click the blank in **Directories** to add the path of **Include**: C:\Program Files\IVI Foundation\VISA\WinNT\include.

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

Note:
The two pathes added here are related to the NI-VISA installation path on your PC. Here, the NI-VISA is installed under C:\Program Files\IVI Foundation\VISA.



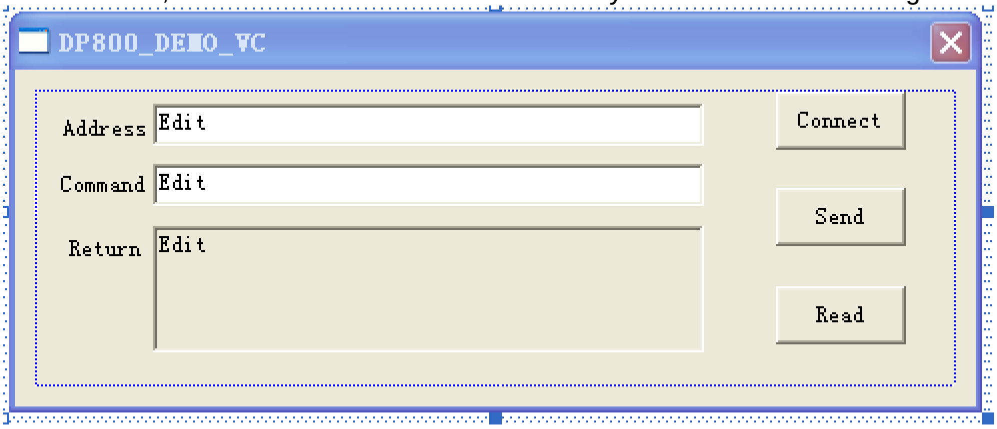
Platform: Win32

Show directories for: Include files

Directories:

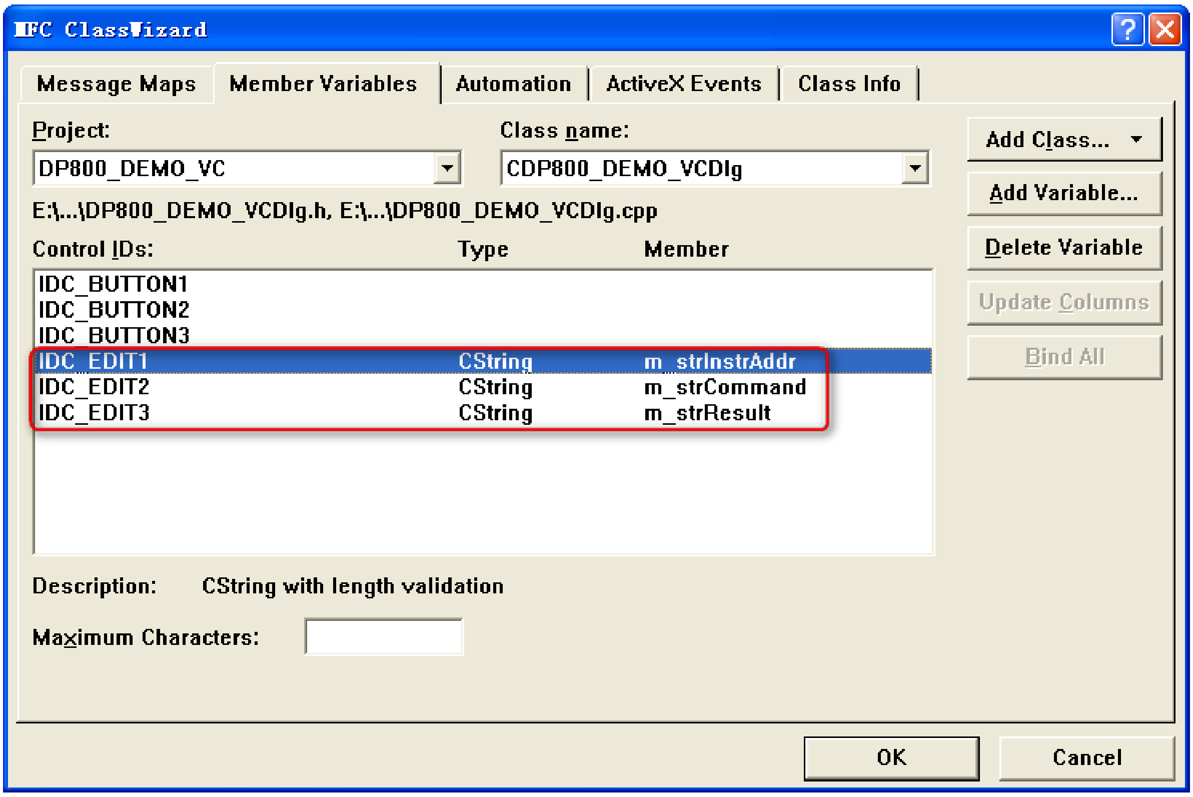
```
D:\Program Files\Microsoft Visual Studio\VC98\INCLUDE
D:\Program Files\Microsoft Visual Studio\VC98\MFC\INCLUDE
D:\Program Files\Microsoft Visual Studio\VC98\ATL\INCLUDE
C:\Program Files\IVI Foundation\VISA\WinNT\include
```

4 Add the **Text**, **Edit** and **Button** controls and the layout is as shown in the figure below.



5 Click **View→ClassWizard** and add the control varibles in the **Member Variables** tab in the pop-up interface.

Instrument address: CString m_strInstrAddr
Command: CString m_strCommand
Return value: CString m_strResult



Project: DP800_DEMO_VC

Class name: CDP800_DEMO_VCDlg

E:\...\DP800_DEMO_VCDlg.h, E:\...\DP800_DEMO_VCDlg.cpp

Control IDs:	Type	Member
IDC_BUTTON1		
IDC_BUTTON2		
IDC_BUTTON3		
IDC_EDIT1	CString	m_strInstrAddr
IDC_EDIT2	CString	m_strCommand
IDC_EDIT3	CString	m_strResult

Description: CString with length validation

Maximum Characters:

6 Encapsulate the write and read operations of VISA.

1) Encapsulate the write operation of VISA for easier operation.

```
bool CDP800_DEMO_VCDlg::InstrWrite(CString strAddr, CString strContent) //write function
{
    ViSession defaultRM,instr;
    ViStatus status;
    ViUInt32 retCount;
    char * SendBuf = NULL;
    char * SendAddr = NULL;
    bool bWriteOK = false;
    CString str;

    //Change the address's data style from CString to char*
    SendAddr = strAddr.GetBuffer(strAddr.GetLength());
    strcpy(SendAddr,strAddr);
    strAddr.ReleaseBuffer();

    //Change the command's data style from CString to char*
    SendBuf = strContent.GetBuffer(strContent.GetLength());
    strcpy(SendBuf,strContent);
    strContent.ReleaseBuffer();

    //open the VISA instrument
    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        AfxMessageBox("No VISA instrument was opened !");
        return false;
    }

    status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);
    //write command to the instrument
    status = viWrite(instr, (unsigned char *)SendBuf, strlen(SendBuf), &retCount);

    //close the instrument
    status = viClose(instr);
    status = viClose(defaultRM);

    return bWriteOK;
}
```

2) Encapsulate the read operation of VISA for easier operation.

```
bool CDP800_DEMO_VCDlg::InstrRead(CString strAddr, CString *pstrResult)
//Read from the instrument
{
    ViSession defaultRM,instr;
    ViStatus status;
    ViUInt32 retCount;
    char * SendAddr = NULL;
    unsigned char RecBuf[MAX_REC_SIZE];
    bool bReadOK = false;
    CString str;

    //Change the address's data style from CString to char*
    SendAddr = strAddr.GetBuffer(strAddr.GetLength());
    strcpy(SendAddr,strAddr);
    strAddr.ReleaseBuffer();

    memset(RecBuf,0,MAX_REC_SIZE);

    //open the VISA instrument
    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        // Error Initializing VISA...exiting
        AfxMessageBox("No VISA instrument was opened !");
        return false;
    }

    //open the instrument
    status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL, &instr);

    //read from the instrument
    status = viRead(instr, RecBuf, MAX_REC_SIZE, &retCount);

    //close the instrument
    status = viClose(instr);
    status = viClose(defaultRM);

    (*pstrResult).Format("%s",RecBuf);

    return bReadOK;
}
```

7 Add the control message response code.

1) Connect the instrument

```
void CDP800_DEMO_VCDlg::OnConnect()
{
    // TODO: Add your control notification handler code here
    ViStatus status;
    ViSession defaultRM;
    ViString expr = "?*";
    ViPFindList findList = new unsigned long;
    ViPUInt32 retcnt = new unsigned long;
    ViChar instrDesc[1000];
    CString strSrc = "";
    CString strInstr = "";
    unsigned long i = 0;
    bool bFindDP = false;

    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        // Error Initializing VISA...exiting
        MessageBox("No VISA instrument was opened ! ");
        return ;
    }

    memset(instrDesc,0,1000);

    // Find resource
    status = viFindRsrc(defaultRM,expr,findList, retcnt, instrDesc);

    for (i = 0;i < (*retcnt);i++)
    {
        // Get instrument name
        strSrc.Format("%s",instrDesc);
        InstrWrite(strSrc,"*IDN?");
        ::Sleep(200);
        InstrRead(strSrc,&strInstr);

    // If the instrument(resource) belongs to the DP series then jump out from the loop
        strInstr.MakeUpper();
        if (strInstr.Find("DP") >= 0)
        {
            bFindDP = true;
            m_strInstrAddr = strSrc;
            break;
        }

        //Find next instrument
        status =   viFindNext(*findList,instrDesc);
    }

    if (bFindDP == false)
    {
        MessageBox("Didn't find any DP!");
    }
    UpdateData(false);
}
```

2) Write operation

```
void CDP800_DEMO_VCDlg::OnSend()
{
    // TODO: Add your control notification handler code here
    UpdateData(true);
    if (m_strInstrAddr.IsEmpty())
    {
        MessageBox("Please connect to the instrument first!");
    }
    InstrWrite(m_strInstrAddr,m_strCommand);
    m_strResult.Empty();
    UpdateData(false);
}
```

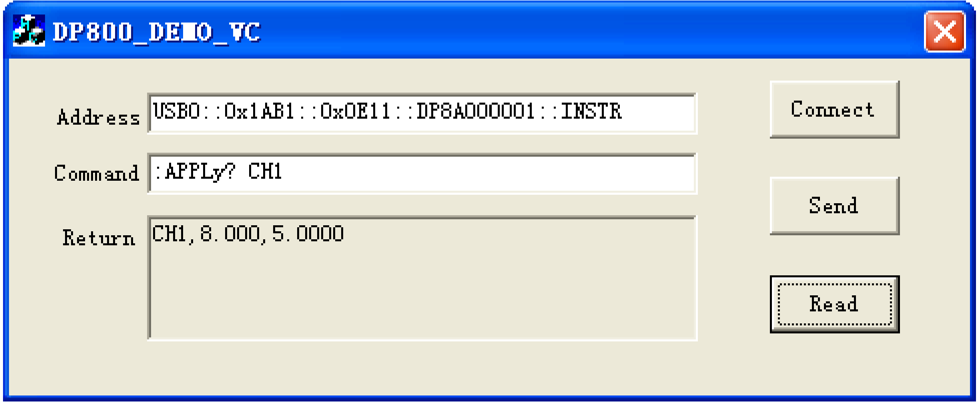
3) Read operation

```
void CDP800_DEMO_VCDlg::OnRead()
{
    // TODO: Add your control notification handler code here
    UpdateData(true);
    InstrRead(m_strInstrAddr,&m_strResult);
    UpdateData(false);
}
```

8 Running results.

1) Click **Connect** to search for the power supply and connect it;
2) Enter the command in to the **Command** textbox, for example, :APPLy? CH1;
3) Click **Send** to send the command;
4) Click **Read** to read the return value.

The running results are as shown in the figure below.



DP800_DEMO_VC

Address USB0::0x1AB1::0x0E11::DP8A000001::INSTR

Command :APPLy? CH1

Return CH1,8.000,5.0000

Chapter 5 Appendix

Appendix A: Command List

  - :PRESet:KEY
  - :PRESet:USER[n]:SET:CURRent
  - :PRESet:USER[n]:SET:DEFault
  - :PRESet:USER[n]:SET:TRACk
  - :PRESet:USER[n]:SET:OCP
  - :PRESet:USER[n]:SET:OVP
  - :PRESet:USER[n]:SET:OTP
  - :PRESet:USER[n]:SET:SURE
  - :PRESet:USER[n]:SET:VOLTage
- :RECorder Commands
  - :RECorder:DESTination?
  - :RECorder:MEMory
  - :RECorder:MMEMory
  - :RECorder:PERIod
  - :RECorder[:STATe]
- :SOURce Commands
  - [:SOURce[n]]:CURRent[:LEVel][:IMMediate][:AMPLitude]
  - [:SOURce[n]]:CURRent:PROTection[:LEVel]
  - [:SOURce[n]]:CURRent:PROTection:STATe
  - [:SOURce[n]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]
  - [:SOURce[n]]:VOLTage:PROTection[:LEVel]
  - [:SOURce[n]]:VOLTage:PROTection:STATe
- :SYSTem Commands
  - :SYSTem:BEEPer[:IMMediate]
  - :SYSTem:BEEPer:STATe
  - :SYSTem:BRIGhtness
  - :SYSTem:COMMunicate:GPIB:ADDRess
  - :SYSTem:COMMunicate:LAN:APPLy
  - :SYSTem:COMMunicate:LAN:AUTOip[:STATe]
  - :SYSTem:COMMunicate:LAN:DHCP[:STATe]
  - :SYSTem:COMMunicate:LAN:DNS
  - :SYSTem:COMMunicate:LAN:GATEway
  - :SYSTem:COMMunicate:LAN:IPADdress
  - :SYSTem:COMMunicate:LAN:MAC?
  - :SYSTem:COMMunicate:LAN:MANualip[:STATe]
  - :SYSTem:COMMunicate:LAN:SMASK
  - :SYSTem:COMMunicate:RS232:BAUD
  - :SYSTem:COMMunicate:RS232:DATABit
  - :SYSTem:COMMunicate:RS232:FLOWCrl

:TRIGger:OUT:CONDition

:TRIGger:OUT:DUTY

:TRIGger:OUT[:ENABle]

:TRIGger:OUT:PERIod

:TRIGger:OUT:POLArity

:TRIGger:OUT:SIGNal

:TRIGger:OUT:SOURce

Appendix B: Factory Setting

Parameter	Factory Setting	
Channel Parameters		
	DP831A	**DP832A**
CH1 Setting Values	0.000V, 5.0000A	0.000V, 3.000A
CH2 Setting Values	00.000V, 2.0000A	0.000V, 3.000A
CH3 Setting Values	-00.000V, 2.0000A	0.000V, 3.000A
CH1 Limits	8.800V, 5.5000A	33.000V, 3.300A
CH2 Limits	33.000V, 2.2000A	33.000V, 3.300A
CH3 Limits	-33.000V, 2.2000A	5.500V, 3.300A
CH1 OVP\OCP On/Off	OFF/OFF	OFF/OFF
CH2 OVP\OCP On/Off	OFF/OFF	OFF/OFF
CH3 OVP\OCP On/Off	OFF/OFF	OFF/OFF
CH1 Track On/Off	None	OFF
CH2 Track On/Off	OFF	OFF
CH3 Track On/Off	OFF	None
CH1 Output On/Off	OFF	
CH2 Output On/Off	OFF	
CH3 Output On/Off	OFF	
Display		
Luminance*	50%	
Contrast*	25%	
RGB Brightness*	50%	
Display Mode	Normal	
System Setting		
Language*	Chinese	
Power-on Setting*	Default	
Print Destination	U Disk	
OTP	On	
Beeper	On	
Screen Saver	Off	
Keyboard Lock Password*	Off	
Preset	Default	
I/O*		
GPIB Address	2	
RS232		
Baud Rate	9600	
Data Bit	8	
Stop Bit	1	
Parity Bit	None	
Hardware Flow Control	Off	
LAN		
DHCP	On	
Auto IP	On	
Manual IP	Off	
Timer		
Channel	CH1	
Timer On/Off	Off	
Output Groups	1	
Timer Parameters	Volt: 1V; Curr: 1A; Time: 1s	

Cycles	1
End State	Off
Object	Volt
Templet	Sine
Delayer	
Channel	CH1
Delayer On/off	Off
Output Groups	1
Delayer Parameter	State: Off, On alternately
Cycles	1
End State	Off
State Generation	0 1Pat
Time Generation	
Generation Method	FixTime
On Delay	1s
Off Delay	1s
Time Base Value	1s
Step	1s
Stop Condition	None
Recorder	
Recorder Switch	Off
Record Period	1s
Destination	C:\REC 10:RIGOL.ROF
Analyzer	
Channel Number	CH1
Analysis Object	Volt
Display	Curve
Current Time	1s
Start Time	1s
End Time	2s
Monitor	
Channel	CH1
Monitor Switch	Off
Monitor Condition	>Volt
Voltage	Half of the rated value
Current	Half of the rated value
Power	The product of voltage times current
Output Off	True
Warning	True
Beeper	True
Trigger	
Direction	In
Trigger Input	
Data Line	D0
Source under Control	CH1
Trigger Type	RiseEdge
Output Response	OutpClose
Sensitivity	Low
Enable	No
Trigger Output	
Data Line	D0

Control Source	CH1
Trigger Condition	OutpClose
Output Signal	Level
Polarity	Pos
Enable	No
Store	
Browser	Directory
File Type	StateFile

Note*: these parameters don't change when the instrument is restored to its factory settings (restarting the instrument when "Default" is selected in **Utility** → **System** →**PowerOn** or sending the *RST command can restore the instrument to its factory settings).

Appendix C: Warranty

RIGOL warrants that its products mainframe and accessories will be free from defects in materials and workmanship within the warranty period.

If a product is proven to be defective within the respective period, **RIGOL** guarantees the free replacement or repair of products which are approved defective. To get repair service, please contact with your nearest **RIGOL** sales and service office.

RIGOL does not provide any other warranty items except the one being provided by this summary and the warranty statement. The warranty items include but not being subjected to the hint guarantee items related to tradable characteristic and any particular purpose. **RIGOL** will not take any responsibility in cases regarding to indirect, particular and ensuing damage.

Appendix D: Any Question or Comment?

If you have any question or comment on our document, please mail to: service@rigol.com