

# **SCPI Programming Reference**

## **Programmable DC power supply**

**HDP43XX/44XX Series**

Version: 1.0

## Contents

SCPI Command introduction.....	1
Command format.....	1
Symbol Description.....	1
Command Abbreviations.....	1
SYSTem.....	2
Get Device Model: SYSTem:GET:MODEl?.....	2
Network DHCP Switch: SYSTem:LAN:DHCP.....	2
Network IP Address: SYSTem:LAN:IP.....	2
Network Subnet Mask: SYSTem:LAN:NETMask.....	3
Network Gateway: SYSTem:LAN:GATEWay.....	3
OUTPut Subsystem.....	4
Channel Switch: OUTPut.....	4
Delay to Open the Channel: OUTPut:DELay:RISE.....	4
Delay to Close the Channel: OUTPut:DELay: FALL.....	5
Channel Coupling Switch: OUTPut:COUPle.....	5
Input Suppression Type: OUTPut:INHibit:MODE.....	5
Clear Input Inhibit State: OUTPut:INHibit:CLEar.....	6
Query Input Suppression Status:OUTPut:INHibit:STATe?.....	6
Operation Mode: OUTPut:OPER:MODE.....	7
CURRent Subsystem.....	7
Set Current: CURRent.....	7
Overcurrent Protection Switch: CURRent:PROTection:STATe.....	8
Set Overcurrent Protection Current: CURRent:PROTection.....	8
VOLTage Subsystem.....	9
Set Voltage: VOLTage.....	9
Overvoltage Protection Switch: VOLTage:PROTection:STATe.....	10
Set the Overvoltage Protection Voltage: VOLTage:PROTection.....	10
MEASure Subsystem.....	11
Query Voltage Measurement Value: MEASure:VOLTage?.....	11
Query Current Measurement Value: MEASure:CURRent?.....	11

LIST Subsystem.....	12
Number of List Entries: LIST:COUNT.....	12
List Item Current Value: LIST:CURREnt.....	12
List Entry Voltage Value: LIST:VOLTage.....	13
List Item Duration: LIST:DWELI.....	13
List Entry Stepping Rules: LIST:PACe.....	13
Output at the End of the List: LIST:TERMinate:LAST.....	14
List BOST Settings: LIST:TOUTput:BOSTep.....	14
List EOST Settings: LIST:TOUTput:EOSTep.....	15
List Trigger Source: LIST:TRIGger:SOURce.....	15
List Trigger Delay Time: LIST:TRIGger:DELay.....	16
Number of List Loops: LIST:REPEat:COUNT.....	16
List Switch: LIST:RUN.....	17
List Trigger Signal: LIST:TRIGger.....	17
SENSE Subsystem.....	18
Trend Chart Current Switch: SENSE:DLOG:FUNcTION:CURREnt.....	18
Trend Chart Voltage Switch: SENSE:DLOG:FUNcTION:VOLTage.....	18
Trend Chart Power Switch: SENSE:DLOG:FUNcTION:POWER.....	19
Trend Chart Recording Time: SENSE:DLOG:FUNcTION:TIME.....	19
Trend Graph Automatically Adjusts the Waveform: SENSE:DLOG:FUNcTION:AUTOset.....	20
Trend Switch: SENSE:DLOG:STATe.....	20
DIGital Subsystem.....	20
Digital IO Function: DIGital:PIN<1-3>:FUNcTION.....	20
Digital IO Polarity: DIGital:PIN<1-3>:POLarity.....	21
Digital IO Output: DIGital:OUTPut:DATA.....	21
Digital IO Input: DIGital:INPut:DATA?.....	22

# SCPI Command introduction

SCPI (Standard Commands for Programmable Instruments, abbreviation: SCPI) defines a set of standard syntax and commands for controlling programmable test and measurement instruments. SCPI commands are ASCII strings. It passes into the instrument through the physical transport layer. The command consists of a series of keywords, and some also need to include parameters. In the protocol, the command is specified as the following form: CONFigure. In use, you can write the full name or only the abbreviation that contains only uppercase letters. Usually, the feedback of the instrument to the query command is also ASCII code. When transferring large amounts of data, binary data can also be used.

## Command format

The command is composed of keywords and parameters. The keywords are separated by a colon ":", and the keywords are followed by optional parameter settings. If there is a "?" after the command line, it means the query function. The keyword and the first parameter must be separated by a space.

For example:

```
CHANnel1: BWLimit 20M
```

```
CHANnel1: BWLimit?
```

The first-level keyword is CHANnel, and the second-level keyword is BWLimit. The keywords at all levels are also separated by ":", 20M is the parameter, and the keywords are separated by spaces. The question mark "?" indicates a query.

## Symbol Description

The following symbols are not sent with the command.

1. Brace {}

The contents in braces are parameter options. The parameter items are usually separated by a vertical bar "|". When using the command, one of the parameters must be selected.

2. Vertical Line |

The vertical line is used to separate multiple parameter options, and one of the parameters must be selected when using the command.

3. Square Brackets []

The content in square brackets can be omitted.

4. Triangle Brackets <>

The parameter in the triangle brackets must be replaced with a valid value.

## Command Abbreviations

All commands are not case sensitive, you can use all uppercase or lowercase. But if you want to abbreviate, you must enter all the capital letters in the Command format.

For example:

```
CHANnel1:BWLimit?  Abbreviation CHAN:BWL?
```

# SYSTEM

## Get Device Model: SYSTem:GET:MODEI?

### Command format

SYSTem:GET:MODEI?

### Function Description

Query the model of the device.

### Parameter

None

### Description

None

### Return value

None

### For example

SYSTem:GET:MODEI? /\* Return HDP4324B\*/

## Network DHCP Switch: SYSTem:LAN:DHCP

### Command format

SYSTem:LAN:DHCP <swi>

SYSTem:LAN:DHCP?

### Function description

Set or query the status of the network DHCP switch.

### Parameter

swi: OFF | ON | 0 | 1.

### Description

Open the DHCP can automatically obtain IP from DHCP server.

Devices without network ports do not support this command.

### Return value

Query returns OFF | ON

### For example

SYSTem:LAN:DHCP ON /\*Open DHCP\*/

SYSTem:LAN:DHCP? /\*Return ON\*/

## Network IP Address: SYSTem:LAN:IP

### Command format

SYSTem:LAN:IP <addr1, addr2, addr3, addr4>

SYSTem:LAN:IP?

### Function description

Set or query the network IP address.

### Parameter

addr1, addr 2, add 3, addr4: 0~255.

**Description**

This command only works when DHCP is turned off.

Equipment without a network interface does not support this command.

**Return value**

The query returns the IP address.

**For example**

```
SYSTem:LAN:IP 10,0,0,105      /* Set the IP address to 10.0.0.105 */
SYSTem:LAN:IP?               /*Return 10.0.0.105*/
```

## Network Subnet Mask: **SYSTem:LAN:NETMask**

**Command format**

```
SYSTem:LAN:NETMask < addr1, addr2, addr3, addr4>
SYSTem:LAN:NETMask?
```

**Function description**

Set or query the network subnet mask.

**Parameter**

addr1, addr2, addr3, addr4: 0~255.

**Description**

This command only works when DHCP is turned off.

Devices without network ports do not support this command.

**Return value**

The query returns the subnet mask.

**For example**

```
SYSTem:LAN:NETMask 255,0,0,0  /* Set the subnet mask to 255.0.0.0 */
SYSTem:LAN:NETMask?          /*Return 255.0.0.0*/
```

## Network Gateway: **SYSTem:LAN:GATEway**

**Command format**

```
SYSTem:LAN:GATEway < addr1, addr2, addr3, addr4>
SYSTem:LAN:GATEway?
```

**Function description**

Set or query network gateway.

**Parameter**

addr1, addr2, addr3, addr4: 0~255.

**Description**

This command only works when DHCP is turned off.

Devices without network ports do not support this command.

**Return value**

The query returns to the gateway.

**For example**

```
SYSTem:LAN:GATEway 10,0,0,1  /* Set the subnet mask to 10.0.0.1*/
SYSTem:LAN:GATEway?          /*Return 10.0.0.1*/
```

# OUTPut Subsystem

## Channel Switch: OUTPut

### Command format

OUTPut <sw,ch>  
OUTPut? <ch>

### Function description

Set or query the channel switch status.

### Parameter

sw: OFF | ON | 0 | 1.

ch:

3-channel model: (@1) | (@2) | (@3) | (@1,2) | (@1,3) | (@2,3) | (@1,2,3).

4-channel model: Increase on the basis of the 3-channel model (@4) | (@1,4) | (@2,4) | (@3,4) | (@1,2,4) | (@1,3,4) | (@2,3,4) | (@1,2,3,4).

(Ch in the following is equivalent)

### Description

ON: Open channel.

OFF: Close channel.

(@?): Operate one or more channels. For example(@1,2): Operate 1 and 2 channels.

### Return value

The query returns ON or OFF.

### For example

```
OUTP ON,@1      /*Open channel 1*/
OUTP? @1        /*Return ON*/
OUTP ON,@1,2    /* Open channel 1,2*/
OUTP? @1,2      /* Return ON,ON*/
```

## Delay to Open the Channel: OUTPut:DELAy:RISE

### Command format

OUTPut:DELAy:RISE <time,ch>  
OUTPut:DELAy:RISE? <ch>

### Function description

Set or query the channel delay opening time.

### Parameter

time: Channel delay open time, unit: s, Floating point.

ch: Refer to the OUTP command.

### Description

When the channel is opened, the channel will be opened after the set time.

### Return value

The query returns the delay opening time of the channel, unit: s

### For example

```
OUTP:DEL:RISE 5.5,@1 /* Set channel 1 to delay 5.5s to open */
```

OUTP:DEL:RISE? (@1) /\*Return 5.5\*/

## Delay to Close the Channel: OUTPut:DELAy: FALL

### Command format

OUTPut:DELAy:FALL <time,ch>

OUTPut:DELAy:FALL? <ch>

### Function description

Set or query the channel delay closing time.

### Parameter

time: Channel delay closing time, unit: s, Floating point.

ch: Refer to the OUTP command.

### Description

When the channel is closed, the channel will be closed after the set time.

### Return value

The query returns the delayed closing time of the channel, unit: s.

### For example

OUTPut:DELAy:FALL 5.5,(@1) /\* Set channel 1 to delay 5.5s to close \*/

OUTPut:DELAy:FALL? (@1) /\*Return 5.5s \*/

## Channel Coupling Switch: OUTPut:COUPlE

### Command format

OUTPut:COUPlE <sw,ch>

OUTPut:COUPlE? <ch>

### Function description

Set or query the channel coupling switch status.

### Parameter

sw: OFF | ON | 0 | 1.

ch: Refer to the OUTP command.

### Description

ON: Turn on channel coupling.

OFF: Turn off channel coupling.

### Return value

The query returns ON or OFF.

### For example

OUTPut:COUPlE ON,(@1) /\* Turn on the channel 1 coupling switch \*/

OUTPut:COUPlE? (@1) /\*Return ON\*/

## Input Suppression Type: OUTPut:INHibit:MODE

### Command format

OUTPut:INHibit:MODE <mode>

OUTPut:INHibit:MODE?



**Function description**

Set or query input suppression type.

**Parameter**

mode: OFF | LATCHED | LIVE

**Description**

After input suppression occurs, all channel outputs will be turned off.

OFF: Turn off input suppression.

LATCHED: When input suppression occurs, unless the input suppression function is turned off or the OUTPUT:INHIBIT:CLEAr (clear the suppression state, otherwise the suppression state will not be eliminated) command is issued.

LIVE: The input suppression state changes with the input level of the digital IO port (PIN3).

**Return value**

Return OFF、LATCHED or LIVE.

**For example**

```
OUTPUT:INHIBIT:MODE LATCHED /* Power input suppression is set to lock */
OUTPUT:INHIBIT:MODE? /* Return LATCHED*/
```

## Clear Input Inhibit State: OUTPUT:INHIBIT:CLEAr

**Command format**

OUTPUT:INHIBIT:CLEAr

**Function description**

Clear input inhibit state.

**Parameter**

None.

**Description**

If input suppression occurs, issue this command to clear the suppression state and restore the channel to its original state.

**Return value**

None.

**For example**

```
OUTPUT:INHIBIT:CLEAr /* Clear input inhibit state */
```

**Query**

**Input**

**Suppression**

## Status:OUTPUT:INHIBIT:STAtE?

**Command format**

OUTPUT:INHIBIT:STAtE?

**Function description**

Query whether input suppression occurs.

**Parameter**

None.

**Description**

None.

**Return value**

Return 1 if suppression occurs, otherwise return 0.

**For example**

OUTPut:INHibit:STAtE? /\* Return 0 or 1\*/

## Operation Mode: OUTPut:OPER:MODE

**Command format**

OUTPut:OPER:MODE <mode>

OUTPut:OPER:MODE?

**Function description**

Set or query the current operating mode.

**Parameter**

mode: INDEPEND | SERIES | PARALLEL | TRACKING.

**Description**

INDEPEND: Independent.

SERIES: Series, channel 1 and channel 2 are connected in series.

PARALLEL: Parallel, channel 1 and channel 2 are connected in parallel.

TRACKING: Tracking (voltage).

**Return value**

Return INDEPEND、SERIES、PARALLEL or TRACKING.

**For example**

OUTPut:OPER:MODE SERIES /\* Set channel 1 and channel 2 in series \*/

OUTPut:OPER:MODE? /\*Return SERIES \*/

## CURRent Subsystem

### Set Current: CURRent

**Command format**

CURRent <curr,chn>

CURRent? <chn>

**Function description**

Set or query the output current value of the channel.

**Parameter**

curr: Current value, unit A, floating point.

chn: 3 channels model:(@1) | (@2) | (@3).

4 channels model: Add an @4 on the basis of the 3 channels model.

(The chn in the following is equivalent)

**Description**

The current value curr can be set range:

3 channels model:  
 Channel 1: 0.002A-3.25A  
 Channel 2: 0.002A-3.25A  
 Channel 3: 0.002A-5.05A  
 4 channels model:  
 Channel 1: 0.002A-3.25A  
 Channel 2: 0.002A-3.25A  
 Channel 3: 0.002A-2.05A  
 Channel 4: 0.002A-1.55A

**Return value**

The query returns the current setting value of the query channel, unit A.

**For example**

```
CURRent 0.5,(@2)    /* Set channel 2 current to 0.5A */
CURRent? (@2)      /*Return 0.5*/
```

**Overcurrent Protection Switch:**  
**CURRent:PROTection:STATe**

**Command format**

```
CURRent:PROTection:STATe <swi,ch>
CURRent:PROTection:STATe? <ch>
```

**Function description**

Set or query the channel overcurrent protection switch status.

**Parameter**

swi: OFF | ON | 0 | 1.  
 ch: Refer to OUP command.

**Description**

**Return value**

Return ON or OFF.

**For example**

```
CURRent:PROTection:STATe ON,(@1,2)
/* Open channel 1 and 2 overcurrent protection */
CURRent:PROTection:STATe? (@1,2)    /*Return ON,ON*/
```

**Set Overcurrent Protection Current :**  
**CURRent:PROTection**

**Command format**

```
CURRent:PROTection <curr,chn>
CURRent:PROTection? <chn>
```

**Function description**

Set or query the channel overcurrent protection current value.

**Parameter**

curr: Overcurrent protection current value, unit A, floating point.

chn: Refer to the CURRent command.

### Description

Overcurrent protection current value curr can be set range:

3 channels model:

Channel 1: 0.002A-3.25A

Channel 2: 0.002A-3.25A

Channel 3: 0.002A-5.05A

4 channels model:

Channel 1: 0.002A-3.25A

Channel 2: 0.002A-3.25A

Channel 3: 0.002A-2.05A

Channel 4: 0.002A-1.55A

### Return value

The query returns the overcurrent protection current setting value of the query channel, unit A.

### For example

```
CURRent:PROTection 1.3,(@2)
```

```
/* Set channel 2 overcurrent protection current to 1.3A */
```

```
CURRent:PROTection? (@2) /*Return 1.3*/
```

## VOLTage Subsystem

### Set Voltage: VOLTage

#### Command format

```
VOLTage <volt,chn>
```

```
VOLTage? <chn>
```

#### Function description

Set or query the output voltage value of the channel.

#### Parameter

volt: Voltage value, unit V, floating point.

chn: Refer to the CURRent command.

#### Description

Voltage value volt can be set range:

3 channels model:

Channel 1: 0V-32.1V

Channel 2: 0V-32.1V

Channel 3: 0V-8.1V

4 channels model:

Channel 1: 0V-32.1V

Channel 2: 0V-32.1V

Channel 3: 0V-8.1V

Channel 4: 0V-16.1V

**Return value**

The query returns the voltage setting value of the query channel, unit V.

**For example**

```
VOLTage 5.5,(@2) /* Set the voltage of channel 2 to 5.5V */
VOLTage? (@2) /*Return 5.5*/
```

**Overvoltage****Protection****Switch:****VOLTage:PROTection:STATe****Command format**

```
VOLTage:PROTection:STATe <swi,ch>
VOLTage:PROTection:STATe? <ch>
```

**Function description**

Set or query the channel overvoltage protection switch status.

**Parameter**

swi: OFF | ON | 0 | 1.  
ch: Refer to OUP command.

**Description**

None.

**Return value**

The query returns ON or OFF.

**For example**

```
VOLTage:PROTection:STATe ON,(@1,2)
/* Open channel 1 and 2 overvoltage protection
*/
VOLTage:PROTection:STATe? (@1,2) /*Return ON,ON*/
```

**Set the Overvoltage Protection Voltage:****VOLTage:PROTection****Command format**

```
VOLTage:PROTection <volt,chn>
VOLTage:PROTection? <chn>
```

**Function description**

Set or query the channel overvoltage protection voltage value.

**Parameter**

volt: Overvoltage protection voltage value, unit V, floating point.  
chn: Refer to the CURRent command.

**Description**

Overvoltage protection voltage value volt can be set range:

3 channels model:

Channel 1: 0V-32.1V

Channel 2: 0V-32.1V

Channel 3: 0V-8.1V  
 4 channels model:  
 Channel 1: 0V-32.1V  
 Channel 2: 0V-32.1V  
 Channel 3: 0V-8.1V  
 Channel 4: 0V-16.1V

**Return value**

The query returns the overvoltage protection voltage setting value of the query channel, unit V.

**For example**

```
VOLTage:PROTection 30.5,(@2)
/* Set the overvoltage protection voltage of channel 2 to 30.5V
*/
VOLTage:PROTection? (@2) /*Return 30.5*/
```

## MEASure Subsystem

**Query Voltage Measurement Value :**

### MEASure:VOLTage?

**Command format**

MEASure:VOLTage? <ch>

**Function description**

Query the measured value of the channel voltage.

**Parameter**

ch: Refer to OUTP command.

**Description**

None.

**Return value**

The query returns the measured voltage value of the channel, unit V.

**For example**

MEASure:VOLTage? (@1)

**Query Current Measurement Value :**

### MEASure:CURRent?

**Command format**

MEASure:CURRent? <ch>

**Function description**

Query channel current measurement value.

**Parameter**

chn: Refer to OUTF command.

**Description**

None.

**Return value**

The query returns the measured current value of the channel, unit V.

**For example**

MEASure:CURRent?(@1)

## LIST Subsystem

### Number of List Entries: LIST:COUNT

**Command format**

LIST:COUNT <nums,chn>

LIST:COUNT? <chn>

**Function description**

Set or query the number of channel list entries.

**Parameter**

nums: Integer (1~50)

chn: Refer to the CURRent command.

**Description**

None

**Return value**

The query returns the number of list entries of the query channel.

**For example**

LIST:COUNT 2,(@1) /\* Set channel 1 list entry to 2 \*/

LIST:COUNT? (@1) /\*Return 2\*/

### List Item Current Value: LIST:CURRENT

**Command format**

LIST:CURRENT <curr1,curr2,...,curr?,chn>

LIST:CURRENT? <chn>

**Function description**

Set or query the current value of the channel list entry.

**Parameter**

curr?: Current value, unit A, floating point.

chn: Refer to the CURRent command.

**Description**

The number of currents set must be equal to the number of list entries.

**Return value**

The query returns the current value of each list item of the query channel, unit A.

**For example**

```
LIST:CURRent 0.5,0.6,@1 /* Set channel 1 list entry current */
LIST:CURRent? (@1) /*Return 0.5,0.6*/
```

## List Entry Voltage Value: LIST:VOLTage

### Command format

```
LIST:VOLTage <volt1,volt2,...,volt?,chn>
LIST:VOLTage? <chn>
```

### Function description

Set or query the voltage value of the channel list entry.

### Parameter

volt?: Voltage value, unit V, floating point.  
chn: Refer to the CURRent command.

### Description

The number of voltages set must be equal to the number of list entries and correspond to the list entries one-to-one.

### Return value

The query returns the voltage value of each list item of the query channel, unit V.

### For example

```
LIST:VOLTage 5.5,6.6,@1 /* Set the channel 1 list entry voltage */
LIST:VOLTage? (@1) /*Return 5.5,6.6*/
```

## List Item Duration: LIST:DWELI

### Command format

```
LIST:DWELI <time1,time2,...,time?,chn>
LIST:DWELI? <chn>
```

### Function description

Set or query the duration of channel list entries.

### Parameter

time?: Time value, unit s, floating point.  
chn: Refer to the CURRent command.

### Description

The number of set times must be equal to the number of list entries.

### Return value

The query returns the duration of each list item of the query channel, unit s.

### For example

```
LIST:DWELI 3.5,3.5,@1 /* Set the duration of channel 1 list entry */
LIST:DWELI? (@1) /*Return 3.5,3.5*/
```

## List Entry Stepping Rules: LIST:PACE

### Command format

```
LIST:PACE <type,ch>
```



LIST:PACE? <ch>

**Function description**

Set or query the step rule between the channel lists.

**Parameter**

type: DWELL | TRIGGER.

ch: Refer to OUTP command.

**Description**

DWELL: After executing the current item, automatically start the next item.

TRIGGER: After executing the current item, wait for the trigger signal on the digital IO port to execute the next item.

**Return value**

Return DWELL | TRIGGER

**For example**

LIST:PACE TRIGGER,(@1) /\* Set the channel 1 list step rule to TRIGGER \*/

LIST:PACE? (@1) /\*Return TRIGGER \*/

## Output at the End of the List: LIST:TERMinate:LAST

**Command format**

LIST:TERMinate:LAST <swi,ch>

LIST:TERMinate:LAST? <ch>

**Function description**

Set or query the output of the channel after the execution of the list is completed.

**Parameter**

swi: OFF | ON | 0 | 1.

ch: Refer to OUTP command.

**Description**

ON: When the list ends, the output voltage or current remains at the value of the last item in the list.

OFF: When the list ends, the output voltage or current returns to the settings that were valid before the list was started.

**Return value**

Return OFF | ON

**For example**

LIST:TERMinate:LAST ON,(@1)

LIST:TERMinate:LAST? (@1)

## List BOST Settings: LIST:TOUTput:BOSTep

**Command format**

LIST:TOUTput:BOSTep <swi1,swi2,...,swin,chn>

LIST:TOUTput:BOSTep? <chn>

**Function description**

Set or query the switch status of the BOST function of the list item.

**Parameter**

Swi?: OFF | ON | 0 | 1.

chn: Refer to the CURRent command.

#### **Description**

ON: When the list item starts to output, it will output a pulse on the digital IO port set to TOUTPUT.

OFF:Close.

#### **Return value**

Return OFF | ON

#### **For example**

```
LIST:TOUTput:BOSTep ON,ON,(@1)          /* Turn on the BOST of the channel 1 list entry */
LIST:TOUTput:BOSTep? (@1)                /*Return ON,ON*/
```

## **List EOST Settings: LIST:TOUTput:EOSTep**

#### **Command format**

LIST:TOUTput:EOSTep <swi1,swi2,...,swin,chn>

LIST:TOUTput:EOSTep? <chn>

#### **Function description**

Set or query the switch status of the EOST function of the list item.

#### **Parameter**

Swi?: OFF | ON | 0 | 1.

chn: Refer to the CURRent command.

#### **Description**

ON: When the list entry ends, a pulse will be output on the digital IO port set as TOUTPUT.

OFF:Close

#### **Return value**

Return OFF | ON

#### **For example**

```
LIST:TOUTput:EOSTep ON,ON,(@1)          /* Enable EOST for channel 1 list entry */
LIST:TOUTput:EOSTep? (@1)                /*Return ON,ON*/
```

## **List Trigger Source: LIST:TRIGger:SOURce**

#### **Command format**

LIST:TRIGger:SOURce <source,ch>

LIST:TRIGger:SOURce? <ch>

#### **Function description**

Set or query the trigger source to start execution of the list.

#### **Parameter**

source: KEY | IO | RMT.

ch: Refer to OUTP command.

#### **Description**

KEY: Press the key to trigger, that is, the list will be executed after pressing the

RUN key.

IO: Digital IO port level trigger, that is, after pressing the RUN key, it will execute after waiting for the pulse list of the IO port.

RMT: Remote command trigger, that is, after pressing the RUN key, wait for the LIST:TRIGger command to be executed.

**Return value**

Return KEY | IO | RMT

**For example**

```
LIST:TRIGger:SOURce KEY,(@1)
```

*/\* Set the trigger source of the channel 1 list to key trigger \*/*

```
LIST:TRIGger:SOURce? (@1) /*Return KEY*/
```

## List Trigger Delay Time: LIST:TRIGger:DElay

**Command format**

```
LIST:TRIGger:DElay <time,ch>
```

```
LIST:TRIGger:DElay? <ch>
```

**Function description**

Set or query the trigger delay time of the list.

**Parameter**

time: 0.0~3600.0s

ch: Refer to OUTP command.

**Description**

After the trigger signal occurs, the trigger delay timer starts timing, and the list starts to output after the timing ends.

**Return value**

Query return delay time, unit s.

**For example**

```
LIST:TRIGger:DElay 1.5,(@1)
```

*/\* Set the trigger delay of the channel 1 list to 1.5s \*/*

```
LIST:TRIGger:DElay? (@1) /*Return 1.5*/
```

## Number of List Loops: LIST:REPEat:COUNT

**Command format**

```
LIST:REPEat:COUNT <nums,ch>
```

```
LIST:REPEat:COUNT? <ch>
```

**Function description**

Set or query the number of cycles of the list.

**Parameter**

nums: 0~9999, Integer.

ch: Refer to OUTP command.

**Description**

If nums is set to 0, the list is output in a continuous loop, if nums is not 0, it represents the number of loops.

**Return value**

The query returns the number of loops, if it returns 0, it is a continuous loop output.

**For example**

```
LIST:REPEat:COUNT 10,(@1)
/* Set the number of cycles of channel 1 list to
10*/
LIST:REPEat:COUNT? (@1) /*Return 10*/
```

**List Switch: LIST:RUN****Command format**

```
LIST:RUN <swi,chn>
LIST:RUN? <chn>
```

**Function description**

Set or query the status of the list switch.

**Parameter**

swi: OFF | ON | 0 | 1  
chn: Refer to the CURRent command.

**Description**

The function is the same as pressing the RUN/STOP button in the menu.

**Return value**

Return OFF | WAIT | RUNNING  
OFF: List closed.  
WAIT: List waiting to be triggered.  
RUNNING: List is running.

**For example**

```
LIST:RUN ON,@1 /* Open channel 1 list */
LIST:RUN? (@1)
```

**List Trigger Signal: LIST:TRIGger****Command format**

```
LIST:TRIGger <chn>
```

**Function description**

Send the list trigger signal of the corresponding channel.

**Parameter**

chn: Refer to the CURRent command.

**Description**

If the corresponding channel list is in WAIT state and the trigger source is RMT, the list starts to execute, otherwise it is invalid.

**Return value**

None

**For example**

```
LIST:TRIGger (@1) /* Send channel 1 list trigger signal */
```

# SENSe Subsystem

**Trend**                      **Chart**                      **Current**                      **Switch:**

## SENSe:DLOG:FUNCTION:CURRENT

### Command format

SENSe:DLOG:FUNCTION:CURRENT <swi,chn>

SENSe:DLOG:FUNCTION:CURRENT? <chn>

### Function description

Set or query the switch status of the current trend of the corresponding channel.

### Parameter

swi: OFF | ON | 0 | 1

chn: Refer to the CURRENT command.

### Description

ON: Open the current trend graph of the corresponding channel.

OFF:Close.

### Return value

Return OFF | ON

### For example

SENSe:DLOG:FUNCTION:CURRENT ON,(@1)

*/\* Open channel 1 current trend graph \*/*

SENSe:DLOG:FUNCTION:CURRENT? (@1)                      */\*Return ON\*/*

**Trend**                      **Chart**                      **Voltage**                      **Switch:**

## SENSe:DLOG:FUNCTION:VOLTage

### Command format

SENSe:DLOG:FUNCTION:VOLTage <swi,chn>

SENSe:DLOG:FUNCTION:VOLTage? <chn>

### Function description

Set or query the switch status of the corresponding channel voltage trend.

### Parameter

swi: OFF | ON | 0 | 1

chn: Refer to the CURRENT command.

### Description

ON: Open the voltage trend graph of the corresponding channel.

OFF:Close.

### Return value

查询返回 OFF | ON

### For example

SENSe:DLOG:FUNCTION:VOLTage ON,(@1)

/\* Open the channel 1 voltage trend graph \*/  
SENSe:DLOG:FUNCTION:VOLTage? (@1) /\*Return ON\*/

**Trend                      Chart                      Power                      Switch:**

## **SENSe:DLOG:FUNCTION:POWER**

### **Command format**

SENSe:DLOG:FUNCTION:POWER <swi,chn>  
SENSe:DLOG:FUNCTION:POWER? <chn>

### **Function description**

Set or query the switch status of the corresponding channel power trend.

### **Parameter**

swi: OFF | ON | 0 | 1  
chn: Refer to the CURRent command.

### **Description**

ON: Open the power trend graph of the corresponding channel  
OFF:Close.

### **Return value**

Return OFF | ON

### **For example**

SENSe:DLOG:FUNCTION:POWER ON,(@1)  
/\* Open the channel 1 power trend graph \*/  
SENSe:DLOG:FUNCTION:POWER? (@1) /\*Return ON\*/

**Trend                      Chart                      Recording                      Time                      :**

## **SENSe:DLOG:FUNCTION:TIME**

### **Command format**

SENSe:DLOG:FUNCTION:TIME <time>  
SENSe:DLOG:FUNCTION:TIME?

### **Function description**

Set or query the total duration of trend graph recording.

### **Parameter**

time: Unit s, floating point.

### **Description**

None

### **Return value**

The query returns the total time of the trend graph record, unit s.

### **For example**

SENSe:DLOG:FUNCTION:TIME 60 /\* Set the trend chart recording time to 60s \*/  
SENSe:DLOG:FUNCTION:TIME? /\*Return 60\*/

## Trend Graph Automatically Adjusts the Waveform :

### SENSe:DLOG:FUNCTION:AUTOset

**Command format**

SENSe:DLOG:FUNCTION:AUTOset

**Function description**

Automatically adjust the trend graph waveform.

**Parameter**

None

**Description**

None

**Return value**

None

**For example**

SENSe:DLOG:FUNCTION:AUTOset

## Trend Switch: SENSe:DLOG:STATE

**Command format**

SENSe:DLOG:STATE <swi>

SENSe:DLOG:STATE?

**Function description**

Set or query the switch status of the trend graph.

**Parameter**

swi: OFF | ON | 0 | 1

**Description**

ON: Trend chart starts to record.

OFF: Close trend graph.

**Return value**

Return OFF | ON

**For example**

SENSe:DLOG:STATE ON /\* Open trend chart \*/

SENSe:DLOG:STATE? /\*Return ON\*/

## DIGital Subsystem

### Digital IO Function: DIGital:PIN<1-3>:FUNCTION

**Command format**

DIGital:PIN<1-3>:FUNCTION <func>

DIGital:PIN<1-3>:FUNCTION?

**Function description**

Set or query the digital IO port function.

**Parameter**

func: DIO | DINPUT | TOUTPUT | TINPUT | OFFCOUPLE | ONCOUPLE | INHIBIT

**Description**

DIO: Ordinary digital IO port for input/output.

DINPUT: Only as input, output is ignored.

TOUTPUT: Trigger output, BOST and EOST for the list.

TINPUT: Trigger input, can be selected as the source of the trigger signal.

OFFCOUPLE: Synchronize the output Off state of each channel (couple on), only one pin can be configured for this function, and the polarity of the pin is fixed.

ONCOUPLE: Synchronize the output On state of each channel (couple on), only one pin can be configured for this function, and the polarity of the pin is fixed.

INHIBIT: Refer to OUTPut:INHibit:MODE command.

**Return value**

Return DIO | DINPUT | TOUTPUT | TINPUT | OFFCOUPLE | ONCOUPLE | INHIBIT

**For example**

```
DIGital:PIN1:FUNCTION DINPUT      /* Set PIN1 pin function to DINPUT */
DIGital:PIN1:FUNCTION?             /*Return DINPUT */
```

## Digital IO Polarity: DIGital:PIN<1-3>:POLarity

**Command format**

DIGital:PIN<1-3>:POLarity <pol>

DIGital:PIN<1-3>:POLarity?

**Function description**

Set or query the polarity of the digital IO port.

**Parameter**

pol: NEGATIVE | POSITIVE

**Description**

NEGATIVE: When In is 1, the actual level on the pin is low.

POSITIVE: When In is 1, the actual level on the pin is high.

**Return value**

Return NEGATIVE | POSITIVE

**For example**

```
DIGital:PIN1:POLarity POSITIVE    /* Set PIN1 pin polarity to POSITIVE */
DIGital:PIN1:POLarity?            /*Return POSITIVE */
```

## Digital IO Output: DIGital:OUTPut:DATA

**Command format**

DIGital:OUTPut:DATA <data>

**Function description**



Set digital IO port output.

**Parameter**

data: Output Data (0~7)

**Description**

When the digital control port is configured as DIO, this command sets the output data on the digital control port. Among the binary weighted values written to the port, the pins are controlled according to the following bit allocation:

Pin	1	2	3
pin	0	1	2

**Return value**

None

**For example**

DIGital:OUTPut:DATA 3 /\* Set PIN1 output 1, PIN2 output 1, PIN3 output 0 \*/

## Digital IO Input: DIGital:INPut:DATA?

**Command format**

DIGital:INPut:DATA?

**Function description**

Query digital IO port input.

**Parameter**

None

**Description**

Pin	1	2	3
pin	0	1	2

**Return value**

Return 0~7

**For example**

DIGital:INPut:DATA?