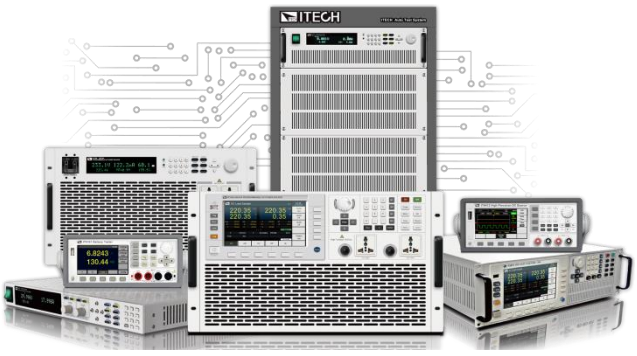
Programmable AC/DC Power Supply IT7900 Series Programming Guide



Model:IT7900
Version:V1.1

Notices

Manual Part Number

IT7900

Revision

First Edition: SEP. 15, 2022

Itech Electronic, Co., Ltd.

Trademarks

Pentium is U.S. registered trademarks of Intel Corporation.

Microsoft, Visual Studio, Windows and MS Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries and regions.

Warranty

The materials contained in this document are provided “as is”, and is subject to change, without prior notice, in future editions. Further, to the maximum extent permitted by applicable laws, ITECH disclaims all warrants, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. ITECH shall not be held liable for errors or for incidental or indirect damages in connection with the furnishing, use or application of this document or of any information contained herein. Should ITECH and the user enter into a separate written agreement with warranty terms covering the materials in this document that conflict with these terms, the warranty terms in the separate agreement shall prevail.

Technology Licenses

The hardware and/or software described herein are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

Restricted permissions of the U.S. government. Permissions for software and technical data which are authorized to the U.S. Government only include those for custom provision to end users. ITECH follows FAR 12.211 (technical data), 12.212 (computer software). DFARS 252.227-7015 (technical data--commercial products) for national defense and DFARS 227.7202-3 (permissions for commercial computer software or computer software documents) while providing the customized business licenses of software and technical data.

Safety Notices

CAUTION

A CAUTION sign denotes a hazard. It calls attention to an operating procedure or practice that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION sign until the indicated conditions are fully understood and met.

WARNING

A WARNING sign denotes a hazard. It calls attention to an operating procedure or practice that, if not correctly performed or adhered to,could result in personal injury or death. Do not proceed beyond a WARNING sign until the indicated conditions are fully understood and met.

NOTE

A NOTE sign denotes important hint. It calls attention to tips or supplementary information that is essential for users to refer to.

Quality Certification and Assurance

We certify that IT7900 power supply meets all the published specifications at time of shipment from the factory.

Warranty

ITECH warrants that the product will be free from defects in material and workmanship under normal use for a period of one (1) year from the date of delivery (except those described in the Limitation of Warranty below).

For warranty service or repair, the product must be returned to a service center designated by ITECH.

- The product returned to ITECH for warranty service must be shipped PREPAID. And ITECH will pay for return of the product to customer.
- If the product is returned to ITECH for warranty service from overseas, all the freights, duties and other taxes shall be on the account of customer.

Limitation of Warranty

This Warranty will be rendered invalid in case of the following:

- Damage caused by circuit installed by customer or using customer own products or accessories;
- Modified or repaired by customer without authorization;
- Damage caused by circuit installed by customer or not operating our products under designated environment;
- The product model or serial number is altered, deleted, removed or made illegible by customer;
- Damaged as a result of accidents, including but not limited to lightning, moisture, fire, improper use or negligence.

Safety Symbols

	Direct current		ON (power on)
	Alternating current		OFF (power off)
	Both direct and alternating current		Power-on state
	Protective conductor terminal		Power-off state
	Earth (ground) terminal		Reference terminal
	Caution, risk of electric shock		Positive terminal
	Warning, risk of danger (refer to this manual for specific Warning or Caution information)		Negative terminal

	Frame or chassis terminal	-	-

Safety Precautions

The following safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or specific warnings elsewhere in this manual will constitute a default under safety standards of design, manufacture and intended use of the instrument. ITECH assumes no liability for the customer's failure to comply with these precautions.

WARNING

- Do not use the instrument if it is damaged. Before operation, check the casing to see whether it cracks. Do not operate the instrument in the presence of inflammable gasses, vapors or dusts.
- The power supply is provided with a three-core power line during delivery and should be connected to a three-core junction box. Before operation, be sure that the instrument is well grounded.
- Make sure to use the power cord supplied by ITECH.
- Check all marks on the instrument before connecting the instrument to power supply.
- Use electric wires of appropriate load. All loading wires should be capable of bearing maximum short-circuit current of power supply without overheating. If there are multiple electronic loads, each pair of the power cord must be capable of bearing the full-loaded rated short-circuit output current.
- Ensure the voltage fluctuation of mains supply is less than 10% of the working voltage range in order to reduce risks of fire and electric shock.
- Do not install alternative parts on the instrument or perform any unauthorized modification.
- Do not use the instrument if the detachable cover is removed or loosen.
- To prevent the possibility of accidental injuries, be sure to use the power adapter supplied by the manufacturer only.
- We do not accept responsibility for any direct or indirect financial damage or loss of profit that might occur when using the instrument.
- This instrument is used for industrial purposes, do not apply this product to IT power supply system.
- Never use the instrument with a life-support system or any other equipment subject to safety requirements.

CAUTION

- Failure to use the instrument as directed by the manufacturer may render its protective features void.
- Always clean the casing with a dry cloth. Do not clean the internals.
- Make sure the vent hole is always unblocked.

Environmental Conditions

The instrument is designed for indoor use and an area with low condensation.

The table below shows the general environmental requirements for the instrument.

Environmental Conditions	Requirements
Operating temperature	0°C to 40°C
Operating humidity	20%-80% (non-condensation)
Storage temperature	-20°C to 70 °C
Altitude	Operating up to 2,000 meters
Pollution degree	Pollution degree 2
Installation category	II

Note

To make accurate measurements, allow the instrument to warm up for 30 min before operation.

Regulatory Markings

CE	The CE mark indicates that the product complies with all the relevant European legal directives. The specific year (if any) affixed refers to the year when the design was approved.
	The instrument complies with the WEEE Directive (2002/96/EC) marking requirement. This affixed product label indicates that you must not discard the electrical/electronic product in domestic household waste.
	This symbol indicates the time period during which no hazardous or toxic substances are expected to leak or deteriorate during normal use. The expected service life of the product is 10 years. The product can be used safely during the 10-year Environment Friendly Use Period (EFUP). Upon expiration of the EFUP, the product must be immediately recycled.

Compliance Information

Complies with the essential requirements of the following applicable European Directives, and carries the CE marking accordingly:

- Electromagnetic Compatibility (EMC) Directive 2014/30/EU
- Low-Voltage Directive (Safety) 2014/35/EU

Conforms with the following product standards:

EMC Standard

IEC 61326-1:2012/ EN 61326-1:2013 [1][2][3]

Reference Standards

CISPR 11:2009+A1:2010/ EN 55011:2009+A1:2010 (Group 1, Class A)
IEC 61000-4-2:2008/ EN 61000-4-2:2009
IEC 61000-4-3:2006+A1:2007+A2:2010/ EN 61000-4-3:2006+A1:2008+A2:2010
IEC 61000-4-4:2004+A1:2010/ EN 61000-4-4:2004+A1:2010
IEC 61000-4-5:2005/ EN 61000-4-5:2006
IEC 61000-4-6:2008/ EN 61000-4-6:2009
IEC 61000-4-11:2004/ EN 61000-4-11:2004

1. The product is intended for use in non-residential/non-domestic environments. Use of the product in residential/domestic environments may cause electromagnetic interference.
2. Connection of the instrument to a test object may produce radiations beyond the specified limit.
3. Use high-performance shielded interface cable to ensure conformity with the EMC standards listed above.

Safety Standard

IEC 61010-1:2010/ EN 61010-1:2010

Content

Chapter1 SCPI Introduction

1.1 Overview

SCPI is short for Standard Commands for Programmable Instruments which defines a communication method of bus controller and instrument. It is based on ASCII and supply for testing and measuring instruments. SCPI command is based on hierarchical architecture which also known as tree system. In this system, Relevant Command is returned to a common node or root, so that a subsystem is formed. A part of OUTPut subsystem is listed below:

OUTPut:

• SYNC {OFF|0|ON|1}

• SYNC:

– MODE {NORMal|CARRier}

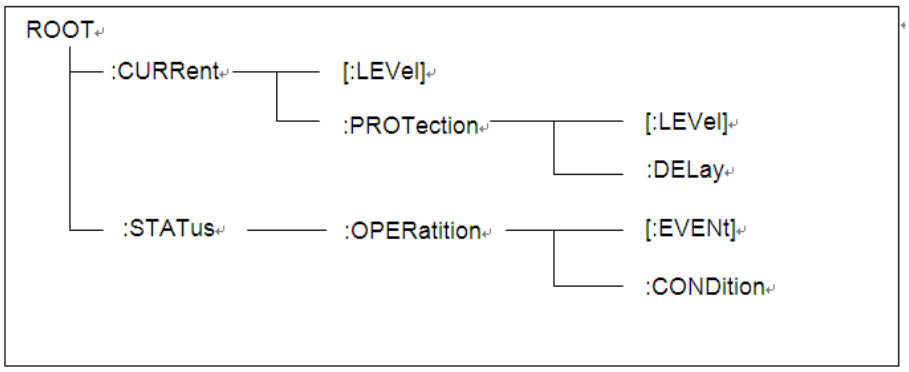
– POLarity {NORMal|INVerted}

OUTPut is the root class keyword, SYNC is the second keyword, MODE and POLarity are the third keyword. Colon(:) is used for separating the command keyword and the next level keyword.

1.2 Command Type of SCPI

SCPI has two types of commands, common and subsystem.

- Common commands generally are not related to specific operation but to controlling overall instrument functions, such as reset, status, and synchronization. All commoncommands consist of a three-letter mnemonic preceded by an asterisk: ***RST *IDN? *SRE 8**.
- Subsystem commands perform specific instrument functions. They are organized into an inverted tree structure with the "root" at the top. The following figure shows a portion of a subsystem command tree, from which you access the commands located along the various paths.



Multiple Commands in a Message

Multiple SCPI commands can be combined and sent as a single message with one message terminator. There are two important considerations when sending several commands within a single message:

- Use a semicolon to separate commands within a message.

- Head paths influence how the instrument interprets commands.

We consider the head path as a string which will be inserted in front of every command of a message. As for the first command of a message, the head path is a null string; for each subsequent command, the head path is a string which is defined to form the current command until and including the head of the last colon separator. A message with two combined commands:

CURR:LEV 3;PROT:STAT OFF

The example indicates the effect of semicolon and explains the concept of head path. Since the head path is defined to be "CURR" after "curr: lev 3", the head of the second command, "curr", is deleted and the instrument explains the second command as:

CURR:PROT:STAT OFF

If "curr" is explicitly included in the second command, it is semantically wrong. Since combining it with the head path will become "CURR:CURR:PROT:STAT OFF", resulting in wrong command.

Movement in the Subsystem

In order to combine commands from different subsystems, you need to be able to reset the header path to a null string within a message. You do this by beginning the command with a colon (:), which discards any previous header path. For example, you could clear the output protection and check the status of the Operation Condition register in one message by using a root specifier as follows:

PROTection:CLEAr;:STATus:OPERation:CONDition?

The following message shows how to combine commands from different subsystems as well as within the same subsystem:

POWer:LEVel 200;PROTection 28; :CURRent:LEVel 3;PROTection:STATe ON

Note the use of the optional header LEVel to maintain the correct path within the voltage and current subsystems, and the use of the root specifier to move between subsystems.

Including Common Commands

You can combine common commands with subsystem commands in the same message. Treat the common command as a message unit by separating it with a semicolon (the message unit separator). Common commands do not affect the header path; you may insert them anywhere in the message.

VOLTage:TRIGgered 17.5;:INITialize;*TRG

OUTPut OFF;*RCL 2;OUTPut ON

Case Sensitivity

Common commands and SCPI commands are not case sensitive. You can use upper or lower, for example:

***RST = *rst**

:DATA? = :data?

:SYSTem:PRESet = :system:preset

Long-Form and Short-Form Versions

A SCPI command word can be sent in its long-form or short-form version. However, the short-form version is indicated by upper case characters. Examples:

:SYSTem:PRESet long-form

:SYST:PRES short form

:SYSTem:PRES long-form and short-form combination

Note that each command word must be in long-form or short-form, and not something in between.

For example, **:SYSTe:PRESe** is illegal and will generate an error. The command will not be executed.

Query

Observe the following precautions with queries:

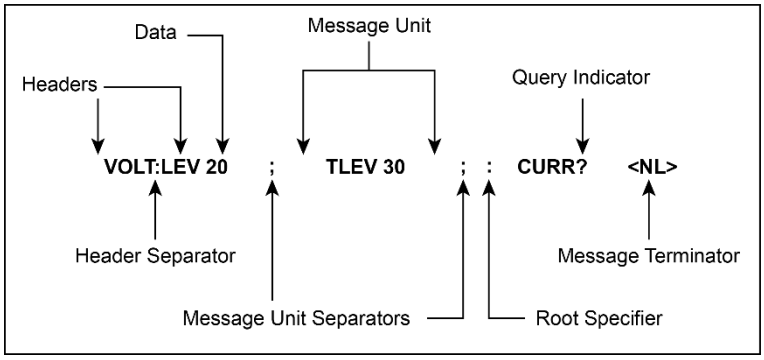
- Set up the proper number of variables for the returned data. For example, if you are reading back a measurement array, you must dimension the array according to the number of measurements that you have placed in the measurement buffer.
- Read back all the results of a query before sending another command to the instrument. Otherwise a Query Interrupted error will occur and the unreturned data will be lost.

1.3 Message Type of SCPI

There are two types of SCPI messages, program and response.

- Program message: A program message consists of one or more properly formatted SCPI commands sent from the controller to the instrument. The message, which may be sent at any time, requests the instrument to perform some action.
- Response message: A response message consists of data in a specific SCPI format sent from the instrument to the controller. The instrument sends the message only when commanded by a program message called a "query."

The next figure illustrates SCPI message structure:



The message unit

The simplest SCPI command is a single message unit consisting of a command header (or keyword) followed by a message terminator. The message unit may include a parameter after the header. The parameter can be numeric or a string.

ABORt<NL>

VOLTage 20<NL>

Headers

Headers, also referred to as keywords, are instructions recognized by the instrument. Headers may be either in the long form or the short form. In the long form, the header is completely spelled out, such as VOLTAGE, STATUS and DELAY. In the short form, the header has only the first three or four letters, such as VOLT, STAT and DEL.

Query indicator

Following a header with a question mark turns it into a query (**VOLTage?**, **VOLTage:PROTection?**). If a query contains a parameter, place the query indicator at the end of the last header (**VOLTage:PROTection?MAX**).

Message unit separator

When two or more message units are combined into a compound message, separate the units with a semicolon (**STATus:OPERation?;QUEStionable?**).

Root specifier

When it precedes the first header of a message unit, the colon becomes the root specifier. It tells the command parser that this is the root or the top node of the command tree.

Message terminator

A terminator informs SCPI that it has reached the end of a message. Three permitted message terminators are:

- newline (<NL>), decimal 10 or hexadecimal 0X0A in ASCII.
- end or identify (<END>)
- both of the above (<NL><END>).

In the examples of this guide, there is an assumed message terminator at the end of each message.

Command execution rules

- Commands execute in the order that they are presented in the program message.
- An invalid command generates an error and, of course, is not executed.
- Valid commands that precede an invalid command in a multiple command program message are executed.
- Valid commands that follow an invalid command in a multiple command program message are ignored.

1.4 Response Data Type

Character strings returned by query statements may take either of the following forms, depending on the length of the returned string:

- **<CRD>**: character response data. Permits the return of character strings.
- **<AARD>**: arbitrary ASCII response data. Permits the return of undelimited 7-bit ASCII. This data type has an implied message terminator.
- **<SRD>**: string response data. Returns string parameters enclosed in double quotes.
- **<Block>**: arbitrary block data.

Response messages

A response message is the message sent by the instrument to the computer in response to a query command.

Sending a response message

After sending a query command, the response message is placed in the Output Queue. When the instrument is then addressed to talk, the response message is sent from the Output Queue to the computer

Multiple response messages

If you send more than one query command in the same program message, the multiple response messages for all the queries is sent to the computer when the instrument is addressed to talk. The responses are sent in the order that the query commands were sent and are separated by semicolons (;). Items within the same query are separated by commas (,). The following example shows the response message for a program message that contains four single item query commands:

0; 1; 1; 0

Response message terminator (RMT)

Each response is terminated with an LF (line feed) and EOI (end or identify). The following example shows how a multiple response message is terminated:

0; 1; 1; 0; <RMT>

Message exchange protocol

Two rules summarize the message exchange protocol:

- **Rule 1**: You must always tell the instrument what to send to the computer.

  The following two steps must always be performed to send information from the instrument other computer:
  1. Send the appropriate query command(s) in a program message.
  2. Address the instrument to talk.
- **Rule 2**: The complete response message must be received by the computer before another program message can be sent to the instrument.

1.5 Command Format

Formats for command display are as follows:

[SOURce[1|2]:]VOLTage:UNIT {VPP|VRMS|DBM}

[SOURce[1|2]:]FREQuency:CENTer {<frequency>|MINimum|MAXimum|DEFault}

Based on the command syntax, most commands (and certain Parameter) are expressed in both upper and lower cases. Upper case refers to abbreviation of commands. Shorter program line may send commands in abbreviated format. Long-format commands may be sent to ensure better program readability.

For example, both formats of VOLT and VOLTAGE are acceptable in the above syntax statements. Upper or lower case may be used. Therefore, formats of VOLTAGE, volt and Volt are all acceptable. Other formats (such as VOL and VOLTAG) are invalid and will cause errors.

- Parameter options with given command strings are included in the brace ({ }). The brace is not sent along with command strings.
- Vertical stripes (|) separate several parameter options with given command strings. For example, {VPP|VRMS|DBM} indicates that you may assign "APP", "VRMS" or "DBM" in the above commands. Vertical stripes are not sent along with command strings.
- Angle brackets (< >) in the second example indicates that a value must be assigned to the parameter in the brace. For example, the parameter in the angle bracket is <frequency> in the above syntax statements. Angle brackets are not sent along with command strings. You must assign a value (such as "FREQ:CENT 1000") to the parameter, unless you select other options displayed in the syntax (such as "FREQ:CENT MIN").
- Some syntax elements (such as nodes and Parameter) are included in square brackets ([ ]). It indicates that these elements can be selected and omitted. Angle brackets are not sent along with command strings. If no value is assigned to the optional Parameter, the instrument will select a default value. In the above examples, "SOURce[1|2]" indicates that you may refer to source channel 1 by "SOURce" or "SOURce1" or "SOUR1" or "SOUR". In addition, since the whole SOURce node is optional (in the square bracket), you can refer to the channel 1 by omitting the whole SOURce node. It is because the channel 1 is the default channel for SOURce language node. On the other hand, if you want to refer to channel 2, "SOURce2" or "SOUR2" must be used in the program line.

Colon (:)

It is used to separate key words of a command with the key words in next level. As shown below:

APPL:SIN 455E3,1.15,0.0

In this example, APPLy command assigns a sine wave with frequency of 455 KHz, amplitude of 1.15 V and DC offset of 0.0 V.

Semicolon (;)

It is used to separate several commands in the same subsystem and can also minimize typing. For example, to send the following command string:

TRIG:SOUR EXT; COUNT 10

has the same effect as sending the following two commands:

TRIG:SOUR EXT
TRIG:COUNT 10

Question mark (?)

You can insert question marks into a command to query current values of most Parameter. For example, the following commands will trigger to set the count as 10:

TRIG:COUN 10

Then, you may query count value by sending the following command:

TRIG:COUN?

You may also query the allowable minimum or maximum count as follows:

TRIG:COUN?MIN
TRIG:COUN?MAX

Comma (,)

If a command requires several Parameter, then a comma must be used to separate adjacent Parameter.

Space

You must use blank characters, [TAB] or [Space] to separate Parameter with key words of commands.

Common commands (*)

The IEEE-488.2 standard defines a set of common commands that perform functions such as reset, self-test, and status operations. Common commands always start with a asterisk (*) and occupy 3 character sizes, including one or more Parameter. Key words of a command and the first parameter are separated by a space. Semicolon (;) can separate several commands as follows:

***RST; *CLS; *ESE 32; *OPC?**

Command terminator

Command strings sent to the instrument must end with a <Newline> (<NL>) character. IEEE-488 EOI (End or Identify) information can be used as <NL> character to replace termination command string of <NL> character. It is acceptable to place one <NL> after a <Enter>. Termination of command string always resets current SCPI command path to root level.

NOTE

As for every SCPI message with one query sent to the instrument, the instrument will use a <NL> or newline sign (EOI) to terminate response of return. For example, if "DISP:TEXT?" is sent, <NL> will be placed after the returned data string to terminate response. If an SCPI message includes several queries separated by semicolon (such as "DISP?;DISP:TEXT?"), <NL> will terminate response returned after response to the last query. In all cases, the program must read <NL> in response before another command is sent to the instrument, otherwise errors will be caused.

1.6 Data Type

SCPI language defines several data types used for program message and response messages.

- Numerical parameter

Commands requiring numerical parameter support the notations of all common decimal notations, including optional signs, decimal points, scientific notation, etc. Special values of numerical parameter are also acceptable, such as MIN, MAX and DEF. In addition, suffixes for engineering units can also be sent together with numerical parameter (including M, k, m or u). If the command accepts only some specific values, the instrument will automatically round the input parameter to acceptable values. The following commands require numerical parameter of frequency value:

[SOURce[1|2]:]FREQuency:CENTer {<Frequency>|MINimum|MAXimum}

– **<NR1>**: represents an integer value, such as 273;
– **<NR2>**: represents a real number in floating-point format, such as .273;
– **<NR3>**: represents a real number in scientific notation, such as 2.73E+2;
– **<Nrf>**: The extensible form includes <NR1>, <NR2> and <NR3>;
– **<Nrf+>**: The extensible decimal form includes <Nrf>, MIN, MAX and DEF. MIN and MAX are the minimum and maximum finite number. Within the range of the parameter definition, DEF is the default of the parameter.

- Discrete parameter

  Discrete parameter are used for settings with limited number of programming values (such as IMMediate, EXTernal or BUS). They can use short and long format like key words of commands. They may be expressed in both upper and lower case. The query response always returns uppercase Parameter in short format. The following commands require discrete parameter in voltage unit:

  [SOURce[1|2]:]VOLTage:UNIT {VPP|VRMS|DBM}

- Boolean parameter

  Boolean parameter refer to true or false binary conditions. In case of false conditions, the instrument will accept "OFF" or "0". In case of true conditions, the instrument will accept "ON" or "1". In query of Boolean settings, the instrument will always return "0" or "1". Boolean parameter are required by the following commands:

  DISPlay {OFF|0|ON|1}

- ASCII string parameter

  String parameter may actually include all ASCII character sets. Character strings must start and end with paired quotation marks; and single quotation marks or double quotation marks are both allowed. Quotation mark separators may also act as one part of a string, they can be typed twice without any character added between them. String parameter is used in the following command:

  DISPlay:TEXT <quoted string>

  For example, the following commands display message of "WAITING..." (without quotation marks) on the front panel of the instrument.

  DISP:TEXT "WAITING..."

  Single quotation marks may also be used to display the same message.

  DISP:TEXT 'WAITING...'

  – **<SPD>**: string program data. String parameters enclosed in single or double quotes.
  – **<CPD>**: character program data.

1.7 Remote Interface Connection

IT7900 series power supplies are equipped with three communication interfaces as standard: USB, LAN and CAN, and two optional interfaces: RS232 and GPIB. Users can choose any one to realize communication with the computer. For detailed introduction of remote interface connection, please refer to the content in

the user manual.

Chapter2 Example of Common Commands

This chapter displays the programming examples to remotely control IT7800 power supply using SCPI commands.

If the user want to change the settings of the instrument, for instance, the output setting value, the command SYST:REM must be sent to the instrument after finishing the connection between the instrument and PC.

Example 1: Identify the power supply

You can verify that you are communicating with the correct IT7600 power supply.

To query the identification of the power supply, enter the following command:

-> *IDN?

To check the error queue of the power supply, enter the following command:

-> SYST:ERR?

Example 2:Common output commands

The following common commands to help you quickly implement common operations. For more command information, refer to the corresponding section. You can omit the small write parts in the following commands.

```
SYSTem:REMote            // Set the power supply to remote operation mode.
*IDN?                    //Query the identification of the power supply
SYSTem:FUNCtion ONE      //Set the power supply mode to 1-phase
SYSTem:FUNCtion THRee    //Set the power supply mode to 3-phase
FUNCtion AC              //Set the output mode to AC mode
FUNCtion DC,CH1          //Set the output mode of CH1 to DC mode
VOLTage 220              //Set the voltage RMS to 220V
FREQuency 60.0           //Set the frequency to 60Hz
CURRent:PROTection:RMS 90    //Set the current RMS protection value to 90A
CURRent:PROTection:PEAK 270  //Set the current Peak protection value to 270A
CURR 30                  //Set the current RMS limit of single phase
CURRent:DC 30            //Set the current limit of DC output
VOLTage 20               //Set the voltage RMS
VOLTage:DC 300,220,110   //Set the Vdc of three channels to 300/220/110V
OUTPut ON                //Turn on the output
MEASure:VOLTage?         //Read back the Vac RMS measurement
MEASure:CURRent?         //Read back the Iac RMS measurement
MEASure:POWer?           //Read back the power measurement
MEASure:FREQuency?       //Read back the frequency measurement
SYSTem:ERRor?            //Query the error information
SYSTem:CLEar             //Clear the error information
```

OUTPut:PROTection:CLEar //Clear the protection status

Example 3: List function commands

The following is a detailed for editing a List of waveforms. The parameters of the waveforms can be modified according to actual needs.

SYST:REM // Set the power supply to remote operation mode.

SYST:FUNC ONE //Set the power supply mode to single mode

FUNC AC //Set the output mode to AC mode

LIST:CREate //Create a new list file

LIST:REP 5 //List file cycles 5 times

LIST:STEP -1, "10, 1000,0.0,1000,PHASe,90,50,1000,Sine,TIME,10,0,OFF"

// Add a step, notice the content in red, the amplitude of DC must be 0.0V under AC mode.（Three phase: "10,10,10,1000,1000,1000,0.0,0.0,0.0,1000,1000,1000,PHASe,356.9,120,1000,240,1000,50,1000,Sine,Sine,Sine,TIME,10,0,OFF"）

LIST:STEP? 1 //Query the parameter of step1

LIST:STEP? 2 //Query the parameter of step2

LIST:SAVE "one-ac.csv" //Save the list file as one-ac.csv

SYST:ERR? //Query the error information

Example 4: THD Wave commands

SYST:REM // Set the power supply to remote operation mode.

SYST:FUNC ONE //Set the power supply mode to single mode

FUNC AC //Set the output mode to AC mode

WAV:EDIT:THD:CLE // Clear memory, otherwise incorrect data may be saved

WAV:EDIT:THD:DATA 2, "8.8,0.0" // Edit 2 THD waveform values and phase angles

WAV:EDIT:THD:DATA 3, "8.8,0.0" // Edit 3 THD waveform values and phase angles

WAV:EDIT:THD:DATA 4, "8.8,0.0" // Edit 4 THD waveform values and phase angles

WAV:EDIT:THD:DATA 5, "8.8,0.0" // Edit 5 THD waveform values and phase angles

WAV:EDIT:THD:SAVE:LOC "5.csv" // Save the present edit THD wave data to the local file, and specify the file name (if the file name is the same, it is considered to modify the present waveform, if the file name does not exist, it is considered to create a new file)

WAVE "5.csv" //Select THD wave 5.csv

VOLTage 220 //Set voltage 220V

OUTP 1 //Enable the output

FETC:VOLT:HARM? A,4 // Measurement of the voltage harmonic amplitude of phase A, 4th harmonic

FETC:VOLT:HARM:DIST? A,4 // Measurement of phase A, voltage harmonic component of the 4th harmonic

FETC:VOLT:HARM:THD? A // Measurement of total harmonic distortion of phase A voltage

FETC:ARR:VOLT:HARM:? A,10 //This command is used to measure the harmonic amplitude of each harmonic of the voltage. the value of NR1 ranges from 0 to 50, the 0th represents the DC component, the 1st represents the fundamental wave, and the 2nd to 50th are the harmonic components. In this command, if you assign a value of 10 to NR1, the result will list the harmonic amplitude of the voltage from 0th to 10th.

Chapter3 SCPI status register

You can get the current status of the power supply by reading the operation status registers. The power supply records the different status of the instrument through the four status register group, the four status register group are: status byte register, standard event register, query status register and operation status register. Status byte register records the information of the other status register.

The following table describes the status signals.

Bit name	Bit	Decimal value	Definition
Questionable Status Register			
OV	0	1	Output is disabled by the over-voltage protection.
OC_Rms	1	2	Output is disabled by the over-current protection.RMS
OC_Peak	2	4	Output is disabled by the over-current protection. PEAK
OP_POSITIVE	3	8	Output is disabled by the over power protection. POSITIVE
OP_NEGATIVE	4	16	Output is disabled by the over power protection. NEGATIVE
UV	5	32	Output is disabled by the under-voltage protection.
OT	6	64	Output is disabled by the over-temperature protection.
UC	7	128	Output is disabled by the under-current protection.
Errsense	8	256	Sense error
Share	9	512	Current sharing failure
Rvs	10	1024	Output terminals reversed
INH	11	2048	External output inhibited
PS	12	4096	Error protection bit (protect shutdown)
OSC	13	8192	Loop oscillation failure
UNR	14	16384	Unknown internal fault of the instrument
OC_Dc	15	32768	Capacitor overcurrent
Operation Status Register			
ACQ-WTG	0	1	Meter is waiting for trigger
ARB-WTG	1	2	ARB is Waiting for trigger
DLOG-WTG	2	4	DLOG is Waiting for trigger
ACQ-Active	3	8	ACQ has been triggered and is being executed
ARB-Active	4	16	ARB has been triggered and is being executed
DLOG-Active	5	32	DLOG has been triggered and is being executed
OFF	6	64	The status of the instrument is off.

Bit name	Bit	Decimal value	Definition
CC	7	128	Output is in constant current.
CV	8	256	Output is in constant voltage.
CW	9	512	Output is in constant current.
CR	10	1024	Output is in constant resistance.
CC-	11	2048	Negative current limit
CP-	12	4096	Negative power limit
CAL	13	8192	Calibration is in progress
DEV_Type0	14	16384	Device type 0,Used in conjunction with device type 1
DEV_Type1	15	32768	Device type 1
WORK_Mode0	16	65536	Work mode 0
WORK_Mode1	17	131072	Work mode 1
Normal	18	262144	Normal mode
List	19	524288	List mode
Sweep	20	1048576	Sweep mode
EXT_Analog	21	2097152	External analog control mode
Law	22	4194304	Law mode
SURGE_Sag	23	8388608	Surge/Trap mode
DIMMING	24	16777216	Dimming mode
EXT_FREQ_LOCK	25	33554432	External frequency lock
Standard Event Register			
OPC	0	1	All commands before and including *OPC have been executed.
QYE	2	4	The instrument tried to read the output buffer but it was empty, a new command line was received before a previous query has been read, or both the input and output buffers are full.
DDE	3	8	A device-specific error, including a self-test error, calibration error or other device-specific error occurred.
EXE	4	16	An execution error occurred.
CME	5	32	A command syntax error occurred.
PON	7	128	Power has been cycled since the last time the event register was read or cleared.
Status Byte Register			
QUES	3	8	This bit is set to 1 when any one status of enabled query status register changes.
MAV	4	16	Output buffer available.
ESB	5	32	Bit ESB is set to 1 when the status of an enabled standard event status.
RQS/MSS	6	64	Register changes.

Bit name	Bit	Decimal value	Definition
OPER	7	128	If the status of enabled operation register changes, then this bit is set to1.

Chapter4 SYSTem Commands

SYSTem:VERSion?

This command queries the SCPI version of the instrument.

Syntax

SYSTem:VERSion?

Arguments

None

Reset value

Not applicable

Example

- > SYST:VERS?
< - "1993.1"

Returns

SRD

SYSTem:ERRor?

This command reads the error code and error information.

Syntax

SYSTem:ERRor?

Arguments

None

Example

- > SYST:ERR?
< - 0,"NO_ERR"

Note

- " - >" indicates the commands that you send to instrument.
- "< -" indicates the response from instrument.

Returns

Error Code	Description
0	No error

Error Code	Description
101	DESIGN ERROR: Too many numeric suffices in Command Spec
110	No Input Command to parse
114	Numeric suffix is invalid value
116	Invalid value in numeric or channel list, e.g. out of range
117	Invalid number of dimensions in a channel list
120	Parameter of type Numeric Value overflowed its storage
130	Wrong units for parameter
140	Wrong type of parameter(s)
150	Wrong number of parameters
160	Unmatched quotation mark (single/double) in parameters
165	Unmatched bracket
170	Command keywords were not recognized
180	No entry in list to retrieve (- number list or channel list)
190	Too many dimensions in entry to be returned in parameters
191	Too many char
-150	String data error
-151	Invalid string data [e.g., END received before close quote]
-158	String data not allowed
-160	Block data error
-161	Invalid block data [e.g., END received before length satisfied]
-168	Block data not allowed
-170	Expression error
-171	Invalid expression
-178	Expression data not allowed
-200	Execution error [generic]
-221	Settings conflict [check current device state]

Error Code	Description
-222	Data out of range [e.g., too large for this device]
-223	Too much data [out of memory; block, string, or expression too long]
-224	Illegal parameter value [device-specific]

Error Code	Description
-225	Out of memory
-230	Data Corrupt or Stale
-270	Macro error
-272	Macro execution error
-273	Illegal macro label
-276	Macro recursion error
-277	Macro redefinition not allowe
-310	System error [generic]
-350	Too many errors [errors beyond 9 lost due to queue overflow]
-499	sets Standard Event Status Register bit #2
-400	Query error [generic]
-410	Query INTERRUPTED [query followed by DAB or GET before response complete]
-430	Query DEADLOCKED [too many queries in command string]
-440	Query UNTERMINATED [after indefinite response]
1	Module Initialization Lost
2	Mainframe Initialization Lost
3	Module Calibration Lost
4	Non-volatile RAM STATE section checksum failed
5	Non-volatile RAM RST section checksum failed

Error Code	Description
10	RAM selftest

Error Code	Description
11	CVDAC selftest 1
12	CVDAC selftest 2
13	CCDAC selftest 1
14	CCDAC selftest 2
20	Input Down
40	Flash write failed
41	Flash erase failed
80	Digital I/O selftest error
213	RS232 buffer overrun error
216	RS232 receiver framing error
217	RS232 receiver parity error
218	RS232 receiver overrun error
220	Front panel uart overrun
221	Front panel uart framing
222	Front panel uart parity
223	Front panel buffer overrun
224	Front panel timeout
225	Front Crc Check error
226	Front Cmd Error

Error Code	Description
401	CAL switch prevents calibration
402	CAL password is incorrect
403	CAL not enabled
404	Computed readback cal constants are incorrect
405	Computed programming cal constants are incorrect
406	Incorrect sequence of calibration commands

Error Code	Description
407	CV or CC status is incorrect for this command
408	Output mode switch must be in NORMAL position
600	Lists inconsistent [lists have different list lengths]
601	Too many sweep points
602	Command only applies to RS232 interface
603	FETCH of data that was not acquired
604	Measurement overrange
605	Command not allowed while list initiated
610	Corrupt update data
611	Not Updating

SYSTem:PRESet

This command is used to make the instrument in a state suitable for panel operation(reset).

Syntax

SYSTem:PRESet

Example

SYST:PRES

SYSTem:POSetup <CPD>

This command is used to set and query some parameters or working status when the instrument is powered on.

- RST: The default value indicates that the factory initialization value is displayed when the instrument is powered on. The specific parameters are described in *RST.
- LAST_ON: Indicates when powered on, the instrument will remain the same parameter settings as last time you turned off the instrument.
- LAST_OFF: Indicates when powered on, the instrument will remain the same settings as last time you turned off the instrument, but the output state is Off.

Syntax

SYSTem:POSetup <CPD>

Arguments

RST|LAST|LAST_OFF

Default value

RST

Returns

None

Example

SYST:POS RST

Query syntax

SYSTem:POSetup?

SYSTem:CLEar

This command is used to clear the error queue.

Syntax

SYSTem:CLEar

Example

SYST:CLE

SYSTem:REMote

This command takes the instrument out of front-panel control mode and switches it to remote control mode.

Syntax

SYSTem:REMote

Example

SYST:REM

SYSTem:LOCal

This command is used to switch the power supply into the control from the front panel.

Syntax

SYSTem:LOCal

Example

SYST:LOC

SYSTem:RWLock

This command locks the power supply in remote control mode. When this command is executed, pressing the LOCAL button does not switch the instrument to local control mode.

Syntax

SYSTem:RWLock

Arguments

None

Reset value

Not applicable

Example

SYST:RWL

SYSTem:BEEPer:IMMediate

This command tests the beeper function of the power supply. If it passes the test, a beep is issued.

Syntax

SYSTem:BEEPer:IMMediate

Arguments

None

Reset value

Not applicable

Example

SYST:BEEP:IMM

SYSTem:BEEPer[:STATe] <CPD>

This command enables or disables the beeper function of the power supply.

Syntax

SYSTem:BEEPer[:STATe] <CPD>

Arguments

OFF|ON

Default value

ON

Returns

OFF/ON

Example

SYST:BEEP OFF

Related syntax

SYSTem:BEEPer[:STATe]?

SYSTem:COMMunicate:USB:TYPE <CPD>

This command is used to set and query the USB interface type. IT7900 series power supply USB can be set as the host or device type, used to set the USB interface for communication or as a storage disk. When set as host type, it is used as a storage disk, and when set as device type, it is used as a communication interface.

Syntax

SYSTem:COMMunicate:USB:TYPE <CPD>

Arguments

DEVice|HOST

Defaults

HOST

Returns

DEVice|HOST

Example

SYST:COMM:USB:TYPE HOST

Query syntax

SYSTem:COMMunicate:USB:TYPE?

SYSTem:COMMunicate:SELect <CPD>

This command is used to set and query the communication method. This series instrument comes standard with four communication interfaces: USB, LAN, VCP and CAN, and supports two optional communication interfaces: GPIB, RS-232. And the RS232 and GPIB options can be selected only after the communication board corresponding to RS232 and GPIB is successfully inserted into the corresponding position on the rear panel of the instrument.

Syntax

SYSTem:COMMunicate:SELect <CPD>

Arguments

TMC|VCP

Defaults

VCP

Returns

TMC|VCP

Example

SYST:COMM:SEL VCP //Set the USB communication interface to VCP

Query syntax

SYSTem:COMMunicate:SELect?

SYSTem:COMMunicate:GPIB:ADDRess <NR1>

This command sets and queries the GPIB address of the power supply.

Syntax

SYSTem:COMMunicate:GPIB:ADDRess <NR1>

Arguments

<NR1>

Settable range :1~30

Default value

1

Returns

<NR1>

Example

SYST:COMM:GPIB:ADDR 2

Query syntax

SYSTem:COMMunicate:GPIB:ADDRess?

SYSTem:COMMunicate:SERial:BAUDrate <CPD>

This command sets and queries the baud rate of the serial port.

Syntax

SYSTem:COMMunicate:SERial:BAUDrate <CPD>

Arguments

<CPD>

115200|57600|38400|19200|9600|4800

Default value

9600

Returns

<CPD>

Example

SYST:COMM:SER:BAUD 4800

Query syntax

SYSTem:COMMunicate:SERial:BAUDrate?

SYSTem:COMMunicate:LAN:IP[:CONFiguration] <SPD>

This command is used to set and query the IP address of the instrument.

Syntax

SYSTem:COMMunicate:LAN:IP[:CONFiguration] <SPD>

Arguments

<SPD>

Defaults

"192.168.0.11"

Returns

<SPD>

Example

SYST:COMM:LAN:IP "192.168.0.11"

Query syntax

SYSTem:COMMunicate:LAN:IP[:CONFiguration]?

SYSTem:COMMunicate:LAN:IP[:CONFiguration]:MODE <CPD>

This command is used to set and query the IP mode of the LAN port.

- MANual: the user manually sets the IP-related parameters.

- AUTO: the system automatically configures IP related parameters.

Syntax

SYSTem:COMMunicate:LAN:IP[:CONFiguration]:MODE <CPD>

Arguments

<CPD>

AUTO|MANual

Default value

MANual

Returns

AUTO|MANual

Example

SYST:COMM:LAN:IP:MODE AUTO //Set the IP mode of the LAN interface to automatic configuration mode.

Query syntax

SYSTem:COMMunicate:LAN:IP[:CONFiguration]:MODE?

SYSTem:COMMunicate:LAN:SMASk <SPD>

This command is used to set and query the subnet mask of the LAN.

Syntax

SYSTem:COMMunicate:LAN:SMASk <SPD>

Arguments

<SPD>

Default value

"255.255.255.0"

Returns

<SPD>

Example

SYST:COMM:LAN:SMAS "255.255.255.1" //This command is used to set the subnet mask of the LAN.

Query syntax

SYSTem:COMMunicate:LAN:SMASk?

SYSTem:COMMunicate:LAN:DGATeway <SPD>

This command is used to set and query the gateway address of the LAN communication.

Syntax

SYSTem:COMMunicate:LAN:DGATeway <SPD>

Arguments

<SPD>

Default value

"192.168.0.1"

Example

SYST:COMM:LAN:DGAT "192.168.0.1"

Query syntax

SYST:COMM:LAN:DGAT?

SYSTem:COMMunicate:LAN:RAWSocketport <port>

This command is used to set and query the socket port of the LAN communication.

Syntax

SYSTem:COMMunicate:LAN:RAWSocketport <port>

Arguments

<SPD>

Default value

"30000"

Example

SYSTem:COMMunicate:LAN:RAWSocketport 30001 //Set the socket port to 30001

Query syntax

SYSTem:COMMunicate:LAN:RAWSocketport? //Query the socket port of LAN interface.

SYSTem:FUNCtion <CPD>

This command is used to set and query the present power supply mode, including single-phase/three-phase/reverse/multi-channel.

Syntax

SYSTem:FUNCtion <device>

Arguments

ONE|THRee|DIFFerence|MULTichannel

Default value

ONE

Returns

ONE|THRee|DIFFerence|MULTichannel

Example

SYST:FUNC ONE

Query syntax

SYSTem:FUNCtion?

SYSTem:BRIGhtness:LEVel <NR1>

This command is used to set and query the screen brightness of the present power supply, the setting range is 1-10.

Syntax

SYSTem:BRIGhtness:LEVel <NR1>

Arguments

1-10

Returns

<NR1>

Default value

8

Example

SYST:BRIG:LEVel 10

Query syntax

SYSTem:BRIGhtness:LEVel?

SYSTem:TOUCh[:STATe] <CPD>

This command is used to set and query the touch screen status of the screen, 0|OFF means the touch screen is turned off, 1|ON means the touch screen is turned on.

Syntax

SYSTem:TOUCh[:STATe] <CPD>

Arguments

<CPD>

0|OFF|1|ON

Returns

0|1

Example

SYST:TOUC 0

Query syntax

SYST:TOUC?

SYSTem:LANGuage <CPD>

This command is used to set and query the language display of the power supply, Chinese or English can be selected.

Syntax

SYSTem:LANGuage <CPD>

Arguments

ENGLish|CHINese

Example

SYST:LANG CHIN

Query syntax

SYSTem:LANGuage?

SYSTem:SOFT:KEYBoard[:STATe] <CPD>

This command is used to set and query the UI soft keyboard switch of the present power supply.

Syntax

SYSTem:SOFT:KEYBoard[:STATe] <CPD>

Arguments

0|OFF|1|ON

Example

SYST:SOFT:KEYB 1

Query syntax

SYSTem:SOFT:KEYBoard[:STATe]?

SYSTem:KNOB:IMMediate:EFFective <CPD>

This command is used to set and query whether the parameters adjusted by the knob take effect immediately.

Syntax

SYSTem:KNOB:IMMediate:EFFective <CPD>

Arguments

0|OFF|1|ON

Default value

0

Returns

0|1|

Example

SYST:KNOB:IMM:EFF 1

Query syntax

SYSTem:KNOB:IMMediate:EFFective?

Chapter5 SOURce Commands

[SOURce:]FUNCtion <CPD1>[,CPD2]

This command is used to set and query the working mode of the power supply.

Syntax:

FUNCtion <CPD1>[,CPD2]

Arguments:

CPD1:DC|AC|DCAC|ACDC

CPD2:CH1|CH2|CH3

CPD1:No DC or DCAC under three-phase

CPD2: Select a channel, this parameter needs to be input only in multi-channel mode

Query syntax:

FUNCtion? [CH1|CH2|CH3]

Returns:

<CPD1>:DC|AC|DCAC|ACDC

<CPD2>:CH1|CH2|CH3

*RST:

AC

Example

```
FUNC AC              //Set the instrument work mode to AC mode
FUNC AC,CH1          //Set the CH1 to AC mode in multi-channel mode.
FUNC? CH1            //Query the channel 1 work mode
```

[SOURce:]FUNCtion:MODE <CPD>

This command is used to set and query the functional mode of the power supply.

Syntax:

FUNCtion:MODE <mode>

Arguments:

NORMal|LIST|SWEep

Query syntax:

FUNCtion:MODE?

Returns:

NORMal|LIST|SWEep

*RST:

NORMal

Example

FUNC:MODE LIST

[SOURce:]CURRent:PROTection:RMS <NRf+> [,CH1|CH2|CH3]

Set the RMS current protection value of the power supply.

Note: the second parameter [,CH1|CH2|CH3] is optional and only valid in multi-channel mode.

Syntax:

CURRent:PROTection:RMS <NRf+> [,CH1|CH2|CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

CURRent:PROTection:RMS? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

Returns:

MAXimum|MINimum|DEFault

Example

CURR:PROT:RMS 30 //Set the RMS protection level to 30A

CURR:PROT:RMS 10,1 // Set the CH1 RMS protection level to 30A under multi-channel mode

CURR:PROT:RMS? //Query the RMS protection level

CURR:PROT:RMS? 1 //Query the CH1 RMS protection level

[SOURce:]CURRent:PROTection:DELay <NRf+> [,CH1|CH2|CH3]

Set the RMS current protection delay time, the unit is: s

Syntax:

CURRent:PROTection:DELay <NRf+> [,CH1|CH2|CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

CURRent:PROTection:DELay? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

Returns:

MAXimum|MINimum|DEFault

Example

CURR:PROT:DEL 0.5

CURR:PROT:DEL 10,1

[SOURce:]CURRent:PROTection:DELay? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

Query the RMS current protection delay time.

Note: both parameters of this query command are optional.

Syntax:

CURRent:PROTection:DELay? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

Arguments:

MAXimum|MINimum|DEFault

Returns:

MAXimum|MINimum|DEFault

Example

CURR:PROT:DEL?

CURR:PROT:DEL? MAX|MIN

CURR:PROT:DEL? 1, MAX|MIN

[SOURce:]CURRent:PROTection:MODE <CPD> [,CH1|CH2|CH3]

This command sets and queries the mode of RMS current protection.

SHUTup: If the measured RMS current value is greater than or equal to the set value and greater than the protection delay time, protection is generated and the output of the machine is turned off.

LIMit: In the same way, when the protection is generated, the machine will still output with the Limit value.

Syntax:

CURRent:PROTection:MODE <CPD> [,CH1|CH2|CH3]

Arguments:

LIMit|SHUTup

Query syntax:

CURRent:PROTection:MODE? [CH1|CH2|CH3]

Returns:

LIMit|SHUTup

[SOURce:]CURRent:PROTection:PEAK <NRf+> [,CH1|CH2|CH3]

This command sets the peak current protection value of the power supply.

Note: the second parameter [,CH1|CH2|CH3] is optional and only valid in multi-channel mode.

Syntax:

CURRent:PROTection:PEAK <NRf+> [,CH1|CH2|CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

CURRent:PROTection:PEAK? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

Returns:

MAXimum|MINimum|DEFault

Example

CURR:PROT:PEAK 30

CURR:PROT:PEAK 10,1

[SOURce:]CURRent:PROTection:PEAK? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

This command is used to query the peak current protection value of the power supply.

Note: both parameters of this query command are optional.

Syntax:

CURRent:PROTection:PEAK? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

Arguments:

MAXimum|MINimum|DEFault

Returns:

MAXimum|MINimum|DEFault

Example

CURR:PROT:PEAK?

CURR:PROT:PEAK? MAX|MIN

CURR:PROT:PEAK? 1,MAX|MIN

[SOURce:]CURRent:PROTection:PEAK:DELay<NRf+> [,CH1|CH2|CH3]

This command sets the peak current protection delay time, the unit is: s.

Syntax:

CURRent:PROTection:PEAK:DELay <NRf+> [,CH1|CH2|CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

CURRent:PROTection:PEAK:DELay? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

Returns:

MAXimum|MINimum|DEFault

Example

CURR:PROT:PEAK:DEL 0.5

CURR:PROT:PEAK:DEL 10,1

[SOURce:]CURRent:PROTection:PEAK:DELay? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

This command queries the peak current protection delay time.

Note: both parameters of this query command are optional.

Syntax:

[SOURce]CURRent:PROTection:PEAK:DELay? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

Arguments:

MAXimum|MINimum|DEFault

Returns:

MAXimum|MINimum|DEFault

Example

CURRent:PROTection:PEAK:DELay?

CURRent:PROTection:PEAK:DELay? MAX|MIN

CURRent:PROTection:PEAK:DELay? 1, MAX|MIN

[SOURce:]CURRent:LIMit:HIGH <NRf+> [,CH1|CH2|CH3]

This command sets and queries the upper limit of RMS current. This common is valid in DC mode and DC+AC mode.

Note: the second parameter is optional and only valid in multi-channel mode.

Syntax:

CURRent:LIMit:HIGH <NRf+> [,CH1|CH2|CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

CURRent:LIMit:HIGH? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

Returns:

MAXimum|MINimum|DEFault

Example:

CURR:LIMit:HIGH 5 //Set the power supply current limit value to 5A.

CURR:LIMit:HIGH 5 ,CH1 //In multi-channel mode, set the current limit value of channel 1 to 5A.

[SOURce:]CURRent:LIMit:LOW <NRf+> [,CH1|CH2|CH3]

This command sets and queries the lower limit of RMS current. This common is valid in DC mode and DC+AC mode.

Note: the second parameter is optional and only valid in multi-channel mode.

Syntax:

CURRent:LIMit:LOW <NRf+> [,CH1|CH2|CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

CURRent:LIMit:LOW? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

Returns:

MAXimum|MINimum|DEFault

Example:

CURR:LIMit:LOW 1 //Set the current lower limit to 1A

CURR:LIMit:LOW 1 ,1 //Under multi-channel mode, set the current lower limit to 1A

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude][:AC] <NRf+>[,NRf+][,NRf+]

Set and query the upper limit of RMS current, which has the same function as the command [SOURce:]CURRent:LIMit:HIGH <NRf+> [,CH1|CH2|CH3].

Syntax:

CURRent[:LEVel][:IMMediate][:AMPLitude][:AC] <NRf+>[,NRf+][,NRf+]

Arguments:

<NRf+>

MAXimum|MINimum|DEFault

Query syntax:

CURRent[:LEVel][:IMMediate][:AMPLitude][:AC]?

Returns:

MAXimum|MINimum|DEFault

Related syntax:

[SOURce:]CURRent:LIMit:HIGH <NRf+> [,CH1|CH2|CH3]

Example:

CURR 30 //Under single mode, set the current RMS to 30A

CURR 30,20,10 //Under multi-channel or 3-three phase mode, set the current RMS to 30A,20A,10A

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]:DC <NRf+>,[NRf+],[NRf+]

DC current limit value setting command in normal mode.

<NRf+>: DC current setting value.

Syntax:

CURRent[:LEVel][:IMMediate][:AMPLitude]:DC <NRf+>,[NRf+],[NRf+]

Query syntax:

CURRent[:LEVel][:IMMediate][:AMPLitude]:DC?

Returns

<NRf+>

Example:

CURRent:DC 30

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude][:AC] <NRf+>[,NRf+][,NRf+]

This command is used to set and query the AC voltage RMS.

In the three-phase mode, the three parameters of the command will be set to the corresponding A/B/C three-phase. If the command contains one Arguments, the A/B/C will be set at the same time (in this case, whether it is balanced or unbalanced,all so).

Syntax:

VOLTage[:LEVel][:IMMediate][:AMPLitude][:AC] <NRf+>[,NRf+][,NRf+]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

VOLTage[:LEVel][:IMMediate][:AMPLitude][:AC]? [MAXimum|MINimum|DEFault]

Returns:

MAXimum|MINimum|DEFault

Example:

VOLTage 220 //Under single mode, set the Voltage RMS to 220V

VOLTage 20,30,40 //Under multi-channel or 3-three phase mode, set the voltage RMS to 20V,30V,40V

[SOURce:]VOLTage:LIMit:HIGH <NRf+> [,CH1|CH2|CH3]

This command sets and query the upper limit of RMS voltage.

Note: the second parameter is optional and only valid in multi-channel mode.

Syntax:

VOLTage:LIMit:HIGH <NRf+> [,CH1|CH2|CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

VOLTage:LIMit:HIGH? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

Returns:

MAXimum|MINimum|DEFault

Example:

VOLTage:LIMit:HIGH 300 //Set the voltage upper limit to 300V

VOLTage:LIMit:HIGH 300,1 //Set the voltage upper limit of CH1 or A phase to

300V

VOLTage:LIMit:HIGH? //Query the voltage upper limit

[SOURce:]VOLTage:LIMit:LOW <NRf+> [,CH1|CH2|CH3]

This command sets and query the lower limit of RMS voltage.

Note: the second parameter is optional and only valid in multi-channel mode.

Syntax:

VOLTage:LIMit:LOW <NRf+> [,CH1|CH2|CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

VOLTage:LIMit:LOW? [CH1|CH2|CH3],[MAXimum|MINimum|DEFault]

Returns:

MAXimum|MINimum|DEFault

Example:

VOLTage:LIMit:LOW -11 //set the lower voltage to -11V

VOLTage:LIMit:LOW -11,1 //set lower voltage of CH1 to -11V

VOLTage:LIMit:LOW? //Query the lower volage setting

[SOURce:]PVOLtage[:LEVel][:IMMediate][:AMPLitude][:AC] <phase>,<NRf+>

It is used to set and query the AC voltage of a certain phase in normal mode.

phase: phase identification.

NRf+: AC voltage setting value.

Syntax:

PVOLtage[:LEVel][:IMMediate][:AMPLitude][:AC] <phase>,<NRf+>

Arguments:

A|B|C

Query syntax:

PVOLtage[:LEVel][:IMMediate][:AMPLitude][:AC]? <phase>, [MAXimum|MINimum|DEFault]

Returns:

A|B|C,<NRf+>

Example:

PVOLtage B,300.0

PVOLtage B,MAX

PVOLT? B

PVOLT? B,MAX

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]:DC <NRf+>,[NRf+],[NRf+]

DC voltage setting command in normal mode.

<NRf+>: DC voltage setting value.

Syntax:

VOLTage[:LEVel][:IMMediate][:AMPLitude]:DC <NRf+>,[NRf+],[NRf+]

Query syntax:

VOLTage[:LEVel][:IMMediate][:AMPLitude]:DC?

*RST:

0V

Example

VOLTage:DC 300.0

VOLTage:DC?

VOLT:DC? MIN

[SOURce:]PVOLtage[:LEVel][:IMMediate][:AMPLitude]:DC <phase>,<NRf+>

In normal mode, set and query the DC voltage value of a phase.

phase: phase identification.

NRf+: DC voltage setting value.

Syntax:

PVOLtage [:LEVel][:IMMediate][:AMPLitude]:DC <phase>,<NRf+>

Query syntax:

PVOLtage[:LEVel][:IMMediate][:AMPLitude]:DC? <phase>, [MAXimum|MINimum|DEFault]

Example

PVOLTage:DC C,30.0

PVOLTage:DC? B

PVOLT:DC? B,MIN

[SOURce:]VOLTage:SLOPe[:AC][:IMMediate] <NRf+>,[NRf+],[NRf+]

This command is used to set and query the AC voltage slope in normal mode.

<NRf+>: AC voltage slope setting value

Syntax:

VOLTage:SLOPe[:AC][:IMMediate] <NRf+>,[NRf+],[NRf+]

Query syntax:

VOLTage:SLOPe[:AC][:IMMediate]? [MAXimum|MINimum|DEFault]

Example

VOLTage:SLOPe 300.0

VOLT:SLOP 300.0,500,800

VOLT:SLOP?

VOLT:SLOP? MAX

[SOURce:]PVOLtage:SLOPe[:IMMediate][:AC] <phase>,<NRf+>

Set and query the AC voltage slope of a phase in normal mode.

phase: phase identification.

NRf+: AC voltage slope setting value.

Syntax:

PVOLtage:SLOPe[:IMMediate][:AC] <phase>,<NRf+>

Query syntax:

PVOLtage:SLOPe[:AC][:IMMediate]? <phase>,[MAXimum|MINimum|DEFault]

Example

PVOLT:SLOP C,300.0

PVOLT:SLOP?

PVOLT:SLOP? C,MAX

[SOURce:]VOLTage:SLOPe[:IMMediate]:DC <NRf+>,[NRf+],[NRf+]

This command is used to set and query the DC voltage slope in normal mode.

<NRf+>: DC voltage slope setting value

Syntax:

[SOURce]VOLTage:SLOPe[:IMMediate]:DC <NRf+>,[NRf+],[NRf+]

Query syntax:

VOLTage:SLOPe[:IMMediate]:DC? [MAXimum|MINimum| DEFault]

Example

VOLT:SLOP:DC 300.0

VOLT:SLOP:DC?

VOLT:SLOP:DC? MAX

[SOURce:]PVOLtage:SLOPe[:IMMediate]:DC <phase>,<NRf+>

Set and query the DC voltage slope in normal mode.

Phase: phase identification.

NRf+: DC voltage slope setting value.

Syntax:

[SOURce]PVOLtage:SLOPe[:IMMediate]:DC <phase>,<NRf+>

Query syntax:

PVOLtage:SLOPe[:IMMediate]:DC? <phase>,[MAXimum|MINimum| DEFault]

*RST:

10V/ms

Example

PVOLT:SLOP:DC C,300.0

PVOLT:SLOP:DC? C

PVOLT:SLOP:DC? C,MAX

[SOURce:]FREQuency[:IMMediate] <NRf+>[,NRf+][,NRf+]

This command is used to set and query the frequency value in normal mode.

<NRf+>: frequency setting value.

Note: in multi-channel, this command must carry three parameters to set at the same time, or use PFREquency to set a separate frequency.

Syntax:

FREQuency[:IMMediate] <NRf+>[,NRf+][,NRf+]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

FREQuency[:IMMediate]? [MAXimum|MINimum| DEFault]

Returns:

MAXimum|MINimum|DEFault

*RST:

50Hz

Example

FREQuency 300.0

FREQuency?

[SOURce:]PFREquency[:IMMediate] <phase>,<NRf+>

Set and query the frequency value of a phase.

Syntax:

PFREquency[:IMMediate] <phase>,<NRf+>

Query syntax:

PFREquency[:IMMediate]? <phase>

Example

PFREQuency A,300.0 //Set the frequency of phase A to 300.0Hz

PFREQuency? A //Query the frequency value of phase A

[SOURce:]FREQuency:SLOPe[:IMMediate] <NRf+>[,NRf+][,NRf+]

This command is used to set and query the slope of the frequency in normal mode.

<NRf+>: frequency setting value.

Syntax:

FREQuency:SLOPe[:IMMediate] <NRf+>[,NRf+][,NRf+]

Query syntax:

FREQuency:SLOPe[:IMMediate]? <phase>

*RST:

1000Hz/s

Example

FREQuency:SLOPe 300.0

FREQ:SLOP 300.0

FREQ:SLOP? C

[SOURce:]PFREquency:SLOPe[:IMMediate] <phase>,<NRf+>

Set and query the frequency slope of a phase in normal mode. Invalid in DC mode.

<NRf+>: frequency setting value

Syntax:

PFREquency:SLOPe[:IMMediate] <phase>,<NRf+>

Query syntax:

PFREquency:SLOPe[:IMMediate]? <phase>

*RST:

1000Hz/S

Example

FREQ:SLOP C,300.0

FREQ:SLOP? C

[SOURce:]POWer:LIMit:HIGH <NRf+> [,CH1|CH2|CH3]

This command sets and queries the upper limit of power protection.

Note: the second parameter is optional and valid in multi-channel mode/three-phase.

Syntax:

POWer:LIMit:HIGH <NRf+> [,CH1|CH2|CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

POWer:LIMit:HIGH? [CH1|CH2|CH3][,MAXimum|MINimum|DEFault]

Example

POWer:LIMit:HIGH 20 //Set the upper limit of the power protection value to 20VA.

POWer:LIMit:HIGH 20,2 //Set the upper power limit of channel 2 to 20VA.

POWer:LIMit:HIGH? 1 //Query the power upper limit of channel 1.

[SOURce:]POWer:LIMit:LOW <NRf+> [,CH1|CH2|CH3]

This command sets and queries the lower limit of power protection.

Note: the second parameter is optional and valid in multi-channel mode/three-phase.

Syntax:

POWer:LIMit:LOW <NRf+> [,CH1|CH2|CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

POWer:LIMit:LOW? [CH1|CH2|CH3][,MAXimum|MINimum|DEFault]

Example

POWer:LIMit:LOW 10 //Set the lower limit of the power protection value to 10VA.

POWer:LIMit:LOW 10,1 //Set the lower power limit of channel 1 to 10VA.

POWer:LIMit:LOW? 1 //Query the lower power limit of channel 1.

[SOURce:] POWer:LIMit:DELay <NRf+> [,CH1|CH2|CH3]

This command sets and queries the power protection delay time, the unit is: s.

Syntax:

POWer:LIMit:DELay <NRf+> [,CH1|CH2|CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

POWer:LIMit:DELay? [CH1|CH2|CH3][,MAXimum|MINimum|DEFault]

Example

POWer:LIMit:DELay 3 //The delay time of over power protection is set to 3s

POWer:LIMit:DELay 3,2 //Set the over-power protection delay time of channel 2 to 3 seconds.

POWer:LIMit:DELay? 2 // Query the over-power protection delay time of channel 2.

[SOURce:]ANGLe:DIFFerence[:IMMediate] <type>,<NRf+>

This command is used to set and query the angle difference in Normal mode.

<type>: angle difference type information

BA: angle difference between B and A

CA: angle difference between C and A

<NRf+>: angle difference setting value

Note: meaningful in three-phase unbalanced mode.

Syntax:

ANGLe:DIFFerence[:IMMediate] <type>,<NRf+>

Query syntax:

ANGLe:DIFFerence[:IMMediate]? <type>, [MAXimum|MINimum|DEFault]

Example

ANGL:DIFFerence BA,300.0 //Set the angle difference between B and A to 300.0°.

ANGL:DIFFerence? CA //Query the angle difference between C and A.

[SOURce:]WAVE[:IMMediate] <wave>,[wave],[wave]

This command is used to set and query the waveform in Normal mode.

<wave>: waveform index value.

- Sine
- Square
- Saw
- Triangle
- Trapezoid
- Clipped-sine

Syntax:

WAVE[:IMMediate] <wave>,[wave],[wave]

Query syntax:

WAVE?

Example

WAVE SIN //Set the waveform to Sine in the present mode.

WAVE? //Query the waveform in present mode.

[SOURce:]PWAVe[:IMMediate] <phase>,<wave>

This command is used to set and query the waveform of specified phase in Normal mode.

<phase>: phase index value.

<wave>: waveform index value.

- Sine
- Square
- Saw
- Triangle
- Trapezoid

- Clipped-sine

Syntax:

PWAVe[:IMMediate] <phase>,<wave>

Query syntax:

PWAVe? <phase>]

*RST:

0-Sine

Example

```
PWAV C,"Sine"          //Set the waveform of phase C to Sine.
PWAV? C                //Query the waveform of C phase.
```

[SOURce:]DIMMing[:STATe] <CPD>[,CH1|CH2|CH3]

This command is used to set and query the dimming function switch.

Syntax:

DIMMing[:STATe] <CPD>[,CH1|CH2|CH3]

Arguments:

0|OFF|1|ON

Query syntax:

DIMMing[:STATe]? [CH1|CH2|CH3]

Returns:

0|1

Example:

```
DIMMing 1,2            //enable the dimming function of CH2.
DIMMing? 1             //Query the dimming function status of CH1
```

[SOURce:]DIMMing:EDGE <CPD>[,CH1|CH2|CH3]

This command is used to set and query the dimming edge.

Syntax:

DIMMing:EDGE <CPD>[,CH1|CH2|CH3]

Arguments:

FRONt|BACK

Query syntax:

DIMMing:EDGE? [CH1|CH2|CH3]

Returns:

FRONt|BACK

[SOURce:]DIMMing:PHASe <NRf>[,CH1|CH2|CH3]

This command is used to set and query the dimming phase.

Syntax:

DIMMing:PHASe <NRf>[,CH1|CH2|CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

DIMMing:PHASe? [CH1|CH2|CH3]

Returns:

MAXimum|MINimum|DEFault

[SOURce:]EXTern:PROGram[:STATe] <CPD>

This command is used to set and query the external control state.

Syntax:

EXTern:PROGram[:STATe] <CPD>

Arguments:

0|OFF|1|ON

Query syntax:

EXTern:PROGram[:STATe]?

Example

EXT:PROG ON

[SOURce:]EXTern:PROGram:MODE <CPD>

This command is used to set and query the external programming mode.

Syntax:

EXTern:PROGram:MODE <CPD>

Arguments:

AM|INSTant

Query syntax:

EXTern:PROGram:MODE?

Example

EXT:PROG:MODE INSTant

[SOURce:]EXTern:MONitor:PHASe <phase>

This command is used to set which phase to monitor, when three phases and multiple channels are used.

Syntax:

EXTern:MONitor:PHASe <phase>

Arguments:

A|B|C

Query syntax:

EXTern:MONitor:PHASe?

Example

EXTern:MONitor:PHASe A //monitor A phase

EXTern:MONitor:PHASe? //query the phase to monitor

[SOURce:]EXTern:PROGram:VOLTage:RANGe <range>

This command is used to set the voltage level for external simulation. Either 50V/1 or 100V/1 can be selected.

This setting is valid for both external analog and external voltage monitoring.

Syntax:

EXTern:PROGram:VOLTage:RANGe <range>

Arguments:

50V/1V|100V/1V

Query syntax:

EXTern:PROGram:VOLTage:RANGe?

Example:

EXTern:PROGram:VOLTage:RANGe 50V/1V

EXTern:PROGram:VOLTage:RANGe?

[SOURce:]CONFigure:HARMonic:THD:FORMula <CPD>

This command is used to set and query the calculation formula of harmonic distortion. Whether to use the fundamental wave (1st order) component data as

the denominator f% or all the harmonic measurement data as the denominator r%.

Syntax:

CONFigure:HARMonic:THD:FORMula <CPD>

Arguments:

FUNDamental|TOTal

Query syntax:

CONFigure:HARMonic:THD:FORMula?

Example

CONF:HARM:THD:FORM TOTal //Set the harmonic measurement formula to r%

[SOURce:]CONFigure:HARMonic:TABLe:TYPE <CPD>

This command is used to set and query the display type of the harmonic distortion table.

Syntax:

CONFigure:HARMonic:TABLe:TYPE <CPD>

Arguments:

<ALL|EVEN|ODD>

Query syntax:

CONFigure:HARMonic:TABLe:TYPE?

Example

CONF:HARM:TABL:TYPE ODD

[SOURce:]CONFigure:HARMonic:TABLe:SELect <CPD>

This command is used to set and query the data type of the harmonic distortion list, whether it is voltage harmonic or current harmonic.

Syntax:

CONFigure:HARMonic:TABLe:SELect <CPD>

Arguments:

<U|I>

Query syntax:

CONFigure:HARMonic:TABLe:SELect?

Example

```
CONF:HARM:TABL:SEL U          //Select to display voltage harmonic data.
```

Chapter6 Output Commands

OUTPut[:STATe] <CPD>

This command sets and query the output state of the power supply.

Syntax:

OUTPut[:STATe] <CPD>

Arguments:

0|OFF|1|ON

Query syntax:

OUTPut[:STATe]?

Returns:

0|OFF|1|ON

Example

OUTP ON //Turn on the power output.

OUTPut:PROTection:CLEar

This command is used to clear the protection status.

Syntax:

OUTPut:PROTection:CLEar

Example

OUTP:PROT:CLE

OUTPut:PROTection:WDOG[:STATe] <CPD>

This command enables or disables the state of the communication watchdog.

Syntax:

OUTPut:PROTection:WDOG[:STATe] <CPD>

Arguments:

0|OFF|1|ON

Returns:

0|1

Example

OUTP:PROT:WDOG ON

OUTPut:PROTection:WDOG:DELay <NRf+>

This command sets and queries the time of the communication watchdog, unit: s.

Syntax:

OUTPut:PROTection:WDOG:DELay <NRf+>

Arguments:

MINimum|MAXimum|DEFault

Query syntax:

OUTPut:PROTection:WDOG:DELay?

Returns:

MINimum|MAXimum|DEFault

Example

OUTPut:PROTection:WDOG:DELay 100

OUTPut:IMPedance[:STATe] <CPD>[,A|B|C|CH1|CH2|CH3]

This command is used to set and query the output state of the impedance.

0|OFF indicates that the function is turned off.

1|ON indicates that the function is turned on.

A|B|C|CH1|CH2|CH3 means to specify a phase or a channel in three-phase mode or multi-channel mode.

Syntax:

OUTPut:IMPedance[:STATe] <CPD>[,A|B|C|CH1|CH2|CH3]

Arguments:

0|OFF|1|ON

Query syntax:

OUTPut:IMPedance[:STATe]? [A|B|C|CH1|CH2|CH3]

Returns:

0|OFF|1|ON

Example

OUTP:IMP ON //Turn on the output impedance.

OUTP:IMP OFF,A //Turn off the output impedance function of phase A.

OUTPut:IMPedance:RESistance:LEVel <NRf+>[,A|B|C|CH1|CH2|CH3]

Set and query the resistance value of the output impedance. It is valid after the output impedance function is turned on.

Syntax:

OUTPut:IMPedance:RESistance:LEVel <NRf+>[,A|B|C|CH1|CH2|CH3]

Arguments:

MINimum|MAXimum|DEFault

Query syntax:

OUTPut:IMPedance:RESistance:LEVel? [A|B|C|CH1|CH2|CH3]

Returns:

MINimum|MAXimum|DEFault

Example:

OUTP:IMP:RES:LEV 10

OUTP:IMP:RES:LEV 100,A

OUTPut:IMPedance:INDuctance:LEVel <NRf>[,A|B|C|CH1|CH2|CH3]

This command is used to set and query the output inductance value.

Syntax:

OUTPut:IMPedance:INDuctance:LEVel <NRf>[,A|B|C|CH1|CH2|CH3]

Arguments:

MINimum|MAXimum|DEFault

Query syntax:

OUTPut:IMPedance:INDuctance:LEVel? [A|B|C|CH1|CH2|CH3]

Returns:

MINimum|MAXimum|DEFault

Example

OUTP:IMP:IND:LEV 10

OUTP:IMP:IND:LEV 5,A

OUTPut:OFF:MODE <CPD>

This command is used to set the output mode when the power output is OFF.

OPENz: disconnect the output relay and the output terminal is in a high-impedance state.

HIGHz: the output relay is closed and the output terminal is in a high-impedance state.

SHORt: the output terminal L and N are shorted.

Syntax:

OUTPut:OFF:MODE <CPD>

Arguments:

OPENz|HIGHz|SHORt

Example

OUTP:OFF:MODE SHORt //Set the output terminals L and N to be shorted when the machine is OFF.

OUTPut:ON:PHASe:MODE <CPD>

This command is used to set and query the phase control mode when the power output is turned on:

PHASe: when the power is turned on, the output starts at the set phase angle
IMMediate: the output starts at phase 0 when the machine is ON

Syntax:

OUTPut:ON:PHASe:MODE <CPD>

Argument:

PHASe|IMMediate

Query syntax:

OUTPut:ON:PHASe:MODE?

Returns:

PHASe|IMMediate

Example

OUTP:ON:PHAS:MODE IMM

OUTPut:ON:PHASe:LEVel <NRf+>

This command is used to set and query the starting phase angle when the power output is turned on.

Syntax:

OUTPut:ON:PHASe:LEVel <NRf+>

Arguments:

MINimum|MAXimum|DEFault

Query syntax:

OUTPut:ON:PHASe:LEVel?

Returns:

MINimum|MAXimum|DEFault

Example

OUTP:ON:PHAS:LEV 30

OUTPut:OFF:PHASe:MODE <CPD>

This command is used to set and query the phase control mode when the power output is turned off:

PHASe: when the power output is OFF, the output will be turned off at the set phase angle

IMMediate: when the power output is OFF, the output is turned off at 0 phase angle

Syntax:

OUTPut:OFF:PHASe:MODE <CPD>

Arguments:

PHASe|IMMediate

Query syntax:

OUTPut:OFF:PHASe:MODE?

Returns:

PHASe|IMMediate

Example

OUTP:OFF:PHAS:MODE PHAS

OUTPut:OFF:PHASe:LEVel <NRf+>

This command is used to set and query the phase angle when the power output is turned off.

Syntax:

OUTPut:OFF:PHASe:LEVel <NRf+>

Arguments:

MINimum|MAXimum|DEFault

Query syntax:

OUTPut:OFF:PHASe:LEVel?

Returns:

MINimum|MAXimum|DEFault

Example

OUTP:OFF:PHAS:LEV 45

OUTPut:BALance[:STATe] <CPD>

In the three-phase mode, it is used to set and query the state of the output balance mode.

Syntax:

OUTPut:BALance[:STATe] <CPD>

Arguments:

0|OFF|1|ON

Query syntax:

OUTPut:BALance[:STATe]?

Returns:

0|OFF|1|ON

Example

OUTP:BAL ON

Chapter7 SENSe Commands

SENSe[:REMote][:STATe] <CPD>

This command enables or disables the sense function.

Syntax:

SENSe[:REMote][:STATe] <CPD>

Arguments:

<OFF|ON|0|1>

Default value:

0|OFF

Query syntax:

SENSe[:REMote][:STATe]?

Returns:

<OFF|ON|0|1>

Example

SENS ON

SENSe:FILTer[:STATe] <CPD>

This command sets and queries the state of meter filter switch.

Syntax:

SENSe:FILTer[:STATe] <CPD>

Arguments:

0|OFF|1|ON

Query syntax:

SENSe:FILTer[:STATe]?

Returns:

0|OFF|1|ON

Example

SENSe:FILT OFF

SENSe:FILTer:LEVel <CPD>

This command is used to set and query the filter level of the meter.

Syntax:

SENSe:FILTer:LEVel <CPD>

Arguments:

SLOW|MEDIUM|FAST

Query syntax:

SENSe:FILTer:LEVel?

Returns:

SLOW|MEDIUM|FAST

Example

SENS:FILT:LEV FAST

SENSe:LOOP:SPEed <CPD>

This command is used to set and query the loop speed.

Syntax:

SENSe:LOOP:SPEed <speed>

Arguments:

HIGH|LOW

Default value:

HIGH

Query syntax:

SENSe:LOOP:SPEed?

Returns:

HIGH|LOW

Example

SENS:LOOP:SPE LOW

SENSe:EXTernal:FREQuency <CPD> [CH1, CH2, CH3]

This command is used to set and query the external frequency lock function.

the second parameter is optional and only valid in multi-channel mode.

Syntax:

SENSe:EXTernal:FREQuency <CPD> [CH1, CH2, CH3]

Arguments:

OFF|FREQency|PHASe

Query syntax:

SENSe:EXTernal:FREQuency?

Returns:

OFF|FREQency|PHASe

Example:

SENS:EXT:FREQ OFF

SENSe:EXTernal:FREQuency:OFFSet:HIGH <NRf>,[CH1, CH2, CH3]

Set and query the frequency upper limit value of the external frequency lock function. the second parameter is optional and only valid in multi-channel mode.

Meaning: when the external frequency lock function is turned on, and the external frequency of the machine is within the current set frequency range, the external frequency will be used, otherwise the frequency set by the user will be used for output.

Syntax:

SENSe:EXTernal:FREQuency:OFFSet:HIGH <NRf>,[CH1, CH2, CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

SENSe:EXTernal:FREQuency:OFFSet:HIGH? [CH1, CH2, CH3]

Returns:

<NRf>

Example:

SENS:EXT:FREQ:OFFS:HIGH 100

SENSe:EXTernal:FREQuency:OFFSet:LOW <NRf>,[CH1, CH2, CH3]

Set and query the frequency lower limit value of the external frequency lock function. the second parameter is optional and only valid in multi-channel mode.

Meaning: when the external frequency lock function is turned on, and the external

frequency of the machine is within the current set frequency range, the external frequency will be used, otherwise the frequency set by the user will be used for output.

Syntax:

SENSe:EXTernal:FREQuency:OFFSet:LOW <NRf>,[CH1, CH2, CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

SENSe:EXTernal:FREQuency:OFFSet:LOW? [CH1, CH2, CH3]

Returns:

<NRf>

Example:

SENS:EXT:FREQ:OFFS:LOW 50

SENSe:EXTernal:FREQuency:PHASe <NRf>,[CH1, CH2, CH3]

This command is used to set and query the angular difference between the machine and the external frequency.

Meaning: when the external frequency lock function is turned on, the output phase of the machine maintains a fixed angular difference with the external frequency.

Syntax:

SENSe:EXTernal:FREQuency:PHASe <NRf>,[CH1, CH2, CH3]

Arguments:

MAXimum|MINimum|DEFault

Query syntax:

SENSe:EXTernal:FREQuency:PHASe? [CH1, CH2, CH3]

Returns:

<NRf>

Example:

SENS:EXT:FREQ:PHAS 1

SENSe:EXTernal:FREQuency:EXCeption <CPD>,[CH1, CH2, CH3]

Set and query the working mode when the external frequency is out of lock.

Meaning: when the external frequency lock function is turned on, the processing method when the external frequency is inconsistent with the set value.

STOP: turn off the output.

SETFrequence: output at the frequency set by the user.

Syntax:

SENSe:EXTernal:FREQuency:EXCeption <CPD>,[CH1, CH2, CH3]

Arguments:

STOP|SETFrequence

Query syntax:

SENSe:EXTernal:FREQuency:EXCeption?

Returns:

STOP|SETFrequence

Example:

SENS:EXT:FREQ:EXC STOP

Chapter8 FETCh & MEASure Commands

FETCh[:SCALar]:CURRent[:AC]?

MEASure[:SCALar]:CURRent[:AC]?

This command is used to read the effective value of single-phase current.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:CURRent? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC:CURR? A

FETCh[:SCALar]:CURRent:DC?

MEASure[:SCALar]:CURRent:DC?

This command is used to read the DC component of single-phase current.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:CURRent? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC:CURR:DC?

FETCh[:SCALar]:CURRent:AMPlitude:MAXimum:POSitive?

MEASure[:SCALar]:CURRent:AMPlitude:MAXimum:POSitive?

This command is used to read the positive peak value of single-phase current.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:CURRent:AMPlitude:MAXimum:POSitive? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC:CURR:AMP:MAX:POS?

FETCh[:SCALar]:CURRent:AMPlitude:MAXimum:NEGative?

MEASure[:SCALar]:CURRent:AMPlitude:MAXimum:NEGative?

This command is used to read the negative peak value of single-phase current.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:CURRent:AMPlitude:MAXimum:NEGative? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC:CURR:AMP:MAX:NEG?

FETCh[:SCALar]:CURRent:AMPlitude:MAXimum?

MEASure[:SCALar]:CURRent:AMPlitude:MAXimum?

This command is used to read the peak value of single-phase current.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:CURRent:AMPlitude:MAXimum? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC:CURR:AMP:MAX?

FETCh[:SCALar]:CURRent:CFACtor?

MEASure [:SCALar]:CURRent:CFACtor?

This command is used to read the current crest factor values of A, B, C three-phase.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:CURRent:CFACtor? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC:CURR:CFAC?

FETCh[:SCALar]:FREQuency?

MEASure[:SCALar]:FREQuency?

This command is used to read the single-phase frequency value.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:FREQuency? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC[:SCAL]:FREQ?

FETCh[:SCALar]:POWer[:REAL]?

MEASure[:SCALar]:POWer[:REAL]?

This command is used to read the single-phase real power value.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:POWer [:REAL]? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC:POWer?

FETCh[:SCALar]:POWer:APParent?

MEASure[:SCALar]:POWer:APParent?

This command is used to read the single-phase apparent power value.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:POWer:APParent? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC:POW:APP?

FETCh[:SCALar]:POWer:REACtive?

MEASure[:SCALar]:POWer:REACtive?

This command is used to read the single-phase reactive power value.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:POWer:REACtive? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC:POW:REAC?

FETCh[:SCALar]:POWer:PFACtor?

MEASure[:SCALar]:POWer:PFACtor?

This command is used to read the single-phase power factor value.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:POWer:PFACtor? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC:POW:PFAC?

FETCh[:SCALar]:VOLTage[:AC]?

MEASure[:SCALar]:VOLTage[:AC]?

This command is used to read the effective value of single-phase voltage.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:VOLTage? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC:VOLT?

FETCh[:SCALar]:VOLTage:DC?

MEASure[:SCALar]:VOLTage:DC?

This command is used to read the DC component of single-phase voltage.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:VOLTage:DC? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf> …

Example:

FETC:VOLT:DC?

FETCh[:SCALar]:VOLTage:AMPlitude:MAXimum?

MEASure[:SCALar]:VOLTage:AMPlitude:MAXimum?

This command is used to read the peak value of single-phase voltage.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:VOLTage:AMPlitude:MAXimum? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf>…

Example:

FETC:VOLT:AMP:MAX?

FETCh[:SCALar]:VOLTage:AMPlitude:MAXimum:POSitive?

MEASure[:SCALar]:VOLTage:AMPlitude:MAXimum:POSitive?

This command is used to read the positive peak value of single-phase voltage.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:VOLTage:AMPlitude:MAXimum:POSitive?
[A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf> …

Example:

FETC:VOLT:AMP:MAX:POS?

FETCh[:SCALar]:VOLTage:AMPlitude:MAXimum:NEGative?

MEASure[:SCALar]:VOLTage:AMPlitude:MAXimum:NEGative?

This command is used to read the negative peak value of single-phase voltage.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]:VOLTage:AMPlitude:MAXimum:NEGative? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf> …

Example:

FETC:VOLT:AMP:MAX:NEG?

FETCh[:SCALar]?

MEASure[:SCALar]?

This command is used to get all METER data of single phase.

Note: this command parameter is optional. If there is no parameter, it will return the value of A, B, C three-phase by default.

Syntax:

FETCh[:SCALar]? [A|B|C|CH1|CH2|CH3]

Arguments:

[A|B|C|CH1|CH2|CH3]

Returns:

<NRf> …

Example:

FETC? A

FETCh[:SCALar]:POWer[:REAL]:TOTal?

MEASure[:SCALar]:POWer[:REAL]:TOTal?

This command is used to read the total power value.

Syntax:

FETCh[:SCALar]:POWer [:REAL]?

Arguments:

None

Returns:

<NRf>

Example:

FETC:POW?

FETCh[:SCALar]:POWer:APParent:TOTal?

MEASure[:SCALar]:POWer:APParent:TOTal?

This command is used to read the total apparent power value.

Syntax:

FETCh[:SCALar]:POWer:APParent?

Arguments:

None

Returns:

<NRf>

Example:

FETC:POW:APP:TOT?

FETCh[:SCALar]:POWer:REACtive:TOTal?

MEASure[:SCALar]:POWer:REACtive:TOTal?

This command is used to read the total reactive power value.

Syntax:

FETCh[:SCALar]:POWer:REACtive?

Arguments:

None

Returns:

<NRf>

Example:

FETC:POW:REAC:TOT?

FETCh[:SCALar]:LTLVoltage[:AC]?

MEASure[:SCALar]:LTLVoltage[:AC]?

This command is used to read the line voltage value between B and A, or between C and A, or between C and B.

Syntax:

FETCh[:SCALar]:LTLVoltage? <BA|CA|CB>

Arguments:

<BA|CA|CB>

Returns:

<NRf>

Example:

FETC:LTLV? CB

FETCh[:SCALar]:VOLTage:HARMonic[:AMPLitude]? <[A|B|C|CH1|CH2|CH3]>,<NR1>

MEASure[:SCALar]:VOLTage:HARMonic[:AMPLitude]? <[A|B|C|CH1|CH2|CH3]>,<NR1>

This command is used to measure the amplitude of voltage harmonics.

Syntax:

FETCh[:SCALar]:VOLTage:HARMonic[:AMPLitude]?
<[A|B|C|CH1|CH2|CH3]>,<NR1>

Arguments:

<[A|B|C|CH1|CH2|CH3]>,<NR1>

NR1: 0-50

Returns:

<NRf>

Example:

FETC:VOLT:HARM? A,4

FETCh[:SCALar]:CURRent:HARMonic[:AMPLitude]? <[A|B|C|CH1|CH2|CH3]>,<NR1>

MEASure[:SCALar]:CURRent:HARMonic[:AMPLitude]? <[A|B|C|CH1|CH2|CH3]>,<NR1>

This command is used to measure the amplitude of current harmonics.

Syntax:

FETCh[:SCALar]:CURRent:HARMonic[:AMPLitude]?
<[A|B|C|CH1|CH2|CH3]>,<NR1>

Arguments:

<[A|B|C|CH1|CH2|CH3]>,<NR1>

NR1: 0-50

Returns:

<NRf>

Example:

FETC:CURR:HARM? A,5

FETCh[:SCALar]:VOLTage:HARMonic:DISTort? <[A|B|C|CH1|CH2|CH3]>,<NR1>

MEASure[:SCALar]:VOLTage:HARMonic:DISTort? <[A|B|C|CH1|CH2|CH3]>,<NR1>

This command is used to measure voltage harmonic components.

Syntax:

FETCh[:SCALar]:VOLTage:HARMonic:DISTort? <[A|B|C|CH1|CH2|CH3]>,<NR1>

Arguments:

<[A|B|C|CH1|CH2|CH3]>,<NR1>

NR1: 0-50

Returns:

<NRf>

Example:

FETC:VOLT:HARM:DIST? A,8

FETCh[:SCALar]:CURRent:HARMonic:DISTort? <[A|B|C|CH1|CH2|CH3]>,<NR1>

MEASure[:SCALar]:CURRent:HARMonic:DISTort? <[A|B|C|CH1|CH2|CH3]>,<NR1>

This command is used to measure current harmonic components.

Syntax:

FETCh[:SCALar]:CURRent:HARMonic:DISTort?
<[A|B|C|CH1|CH2|CH3]>,<NR1>

Arguments:

<[A|B|C|CH1|CH2|CH3]>,<NR1>

NR1: 0-50

Returns:

<NRf>

Example:

FETC:CURR:HARM:DIST? 3,10

FETCh[:SCALar]:VOLTage:HARMonic:PHASe? <[A|B|C|CH1|CH2|CH3]>,<NR1>

MEASure[:SCALar]:VOLTage:HARMonic:PHASe? <[A|B|C|CH1|CH2|CH3]>,<NR1>

This command is used to measure the phase of voltage harmonics.

Syntax:

FETCh[:SCALar]:VOLTage:HARMonic:PHASe? <[A|B|C|CH1|CH2|CH3]>,<NR1>

Arguments:

<[A|B|C|CH1|CH2|CH3]>,<NR1>

NR1: 0-50

Returns:

<NRf>

Example:

FETC:VOLT:HARM:PHAS? 2,5

FETCh[:SCALar]:CURRent:HARMonic:PHASe? <[A|B|C|CH1|CH2|CH3]>,<NR1>

MEASure[:SCALar]:CURRent:HARMonic:PHASe? <[A|B|C|CH1|CH2|CH3]>,<NR1>

This command is used to measure the phase of current harmonics.

Syntax:

FETCh[:SCALar]:CURRent:HARMonic:PHASe?
<[A|B|C|CH1|CH2|CH3]>,<NR1>

Arguments:

<[A|B|C|CH1|CH2|CH3]>,<NR1>

NR1: 0-50

Returns:

<NRf>

Example:

FETC:CURR:HARM:PHAS? 2,10

FETCh[:SCALar]:VOLTage:HARMonic:THD? <[A|B|C|CH1|CH2|CH3]>

MEASure[:SCALar]:VOLTage:HARMonic:THD? <[A|B|C|CH1|CH2|CH3]>

This command is used to measure the total harmonic distortion of the voltage.

Syntax:

FETCh[:SCALar]:VOLTage:HARMonic:THD? <[A|B|C|CH1|CH2|CH3]>

Arguments:

<[A|B|C|CH1|CH2|CH3]>

Returns:

<NRf>

Example:

FETC: VOLT:HARM:THD? C

FETCh[:SCALar]:CURRent:HARMonic:THD? <[A|B|C|CH1|CH2|CH3]>

MEASure[:SCALar]:CURRent:HARMonic:THD? <[A|B|C|CH1|CH2|CH3]>

This command is used to measure the total harmonic distortion of the current.

Syntax:

FETCh[:SCALar]:CURRent:HARMonic:THD? <[A|B|C|CH1|CH2|CH3]>

Arguments:

<[A|B|C|CH1|CH2|CH3]>

Returns:

<NRf>

Example:

FETC: CURR:HARM:THD? A

FETCh[:SCALar]:ARRay:VOLTage:HARMonic[:AMPLitude]? <[A|B|C|CH1|CH2|CH3]>,<NR1>

MEASure[:SCALar]:ARRay:VOLTage:HARMonic[:AMPLitude]? <[A|B|C|CH1|CH2|CH3]>,<NR1>

This command is used to measure each harmonic amplitude of the voltage.

Meaning: the value range of NR1 is 0-50, the 0th order represents the DC component, the 1st order represents the fundamental wave, and the 2nd to 50th order the harmonic components. In this command, if NR1 is assigned a value of 10, the result will list the harmonic amplitude of the voltage from 0 to 10th.

Syntax:

FETCh[:SCALar]:ARRay:VOLTage:HARMonic[:AMPLitude]?
<[A|B|C|CH1|CH2|CH3]>,<NR1>

Arguments:

<[A|B|C|CH1|CH2|CH3]>,<NR1>

Returns:

<NRf>

Example:

FETC:ARR:VOLT:HARM? B,11

FETCh[:SCALar]:ARRay:CURRent:HARMonic[:AMPLitude]? <[A|B|C|CH1|CH2|CH3]>,<NR1>

MEASure[:SCALar]:ARRay:CURRent:HARMonic[:AMPLitude]? <[A|B|C|CH1|CH2|CH3]>,<NR1>

This command is used to measure each harmonic amplitude of the current.

Syntax:

FETCh[:SCALar]:ARRay:CURRent:HARMonic[:AMPLitude]?
<[A|B|C|CH1|CH2|CH3]>,<NR1>

Arguments:

<[A|B|C|CH1|CH2|CH3]>,<NR1>

NR1: 0-50

Returns:

<NRf>

Example:

FETC:ARR:CURR:HARM? A,6

FETCh[:SCALar]:ARRay:VOLTage:HARMonic:PHASe? <[A|B|C|CH1|CH2|CH3]>,<NR1>

MEASure[:SCALar]:ARRay:VOLTage:HARMonic:PHASe? <[A|B|C|CH1|CH2|CH3]>,<NR1>

This command is used to measure each harmonic phase of the voltage.

Syntax:

FETCh[:SCALar]:ARRay:VOLTage:HARMonic:PHASe?
<[A|B|C|CH1|CH2|CH3]>,<NR1>

Arguments:

<[A|B|C|CH1|CH2|CH3]>,<NR1>

NR1: 0-50

Returns:

<NRf>

Example:

FETC:ARR:VOLT:HARM:PHAS? 3,6

FETCh[:SCALar]:ARRay:CURRent:HARMonic:PHASe? <[A|B|C|CH1|CH2|CH3]>,<NR1>

MEASure[:SCALar]:ARRay:CURRent:HARMonic:PHASe? <[A|B|C|CH1|CH2|CH3]>,<NR1>

This command is used to measure each harmonic phase of the current.

Syntax:

FETCh[:SCALar]:ARRay:CURRent:HARMonic:PHASe?
<[A|B|C|CH1|CH2|CH3]>,<NR1>

Arguments:

<[A|B|C|CH1|CH2|CH3]>,<NR1>

NR1: 0-50

Returns:

<NRf>

Example:

FETC:ARR:CURR:HARM:PHAS? 4

FETCh[:SCALar]:ARRay:VOLTage:HARMonic:DISTort? <[A|B|C|CH1|CH2|CH3]>,<NR1>

MEASure[:SCALar]:ARRay:VOLTage:HARMonic:DISTort? <[A|B|C|CH1|CH2|CH3]>,<NR1>

This command is used to measure each harmonic distortion of the voltage.

Syntax:

FETCh[:SCALar]:ARRay:VOLTage:HARMonic:DISTort?
<[A|B|C|CH1|CH2|CH3]>,<NR1>

Arguments:

<[A|B|C|CH1|CH2|CH3]>,<NR1>

NR1: 0-50

Returns:

<NRf>

Example:

FETC:ARR:VOLT:HARM:DIST? 6

FETCh[:SCALar]:ARRay:CURRent:HARMonic:DISTort? <[A|B|C|CH1|CH2|CH3]>,<NR1>

MEASure[:SCALar]:ARRay:CURRent:HARMonic:DISTort? <[A|B|C|CH1|CH2|CH3]>,<NR1>

This command is used to measure each harmonic distortion of the current.

Syntax:

FETCh[:SCALar]:ARRay:CURRent:HARMonic:DISTort?
<[A|B|C|CH1|CH2|CH3]>,<NR1>

Arguments:

<[A|B|C|CH1|CH2|CH3]>,<NR1>

NR1: 0-50

Returns:

<NRf>

Example:

FETC:ARR:CURR:HARM:DIST? 6

VETCor:OEDer <NR1>

This command is used to set and query the order of the vector diagram.

Syntax:

VETCor:OEDer

Arguments:

<NR1>

NR1: 0-50

Returns:

<NRf>

Query syntax

VETC:OED?

Example:

VETC:OED 19

VETCor:DATA?

This command is used to query the vector data corresponding to the present order.

Syntax:

VETCor:DATA?

Arguments:

None

Returns:

<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>

Example:

VETCor:DATA?

VETCor:TYPE <CPD>

This command is used to set and query the vector type.

Syntax:

VETCor:TYPE <CPD>

Arguments:

<U|I|ALL>

Returns:

U|I|ALL

Query syntax

VETCor:TYPE??

Example:

VETC:TYPE I

Chapter9 ABORt Subsystem

ABORt:ACQuire

This command is used to stop the sweep function.

Syntax

ABORt:ACQuire

Example

ABOR:ACQ

ABORt:LIST

This command is used to discard the present measurement.

Syntax

ABORt:LIST

Example

ABOR:LIST

ABORt:SWEep

This command is used to stop the sweep function.

Syntax

ABORt:SWEep

Example

ABOR:SWE

ABORt:SURGesag

This command is used to stop the surge/trap wave function.

Syntax

ABORt:SURGesag

Example

ABORt:SURGesag

Chapter10 INITiate Subsystem

INITiate[:IMMediate]:ACQuire

This command is used to Initiates the measurement trigger system.

Syntax

INITiate[:IMMediate]:ACQuire

Example

INIT:ACQ

INITiate[:IMMediate]:LIST

This command is used to start up the LIST function.

Syntax

INITiate[:IMMediate]:LIST

Example

INIT:LIST

INITiate[:IMMediate]:SWEep

This command is used to initialize the sweep function.

Syntax

INITiate[:IMMediate]:SWEep

Example

INIT:SWE

INITiate[:IMMediate]:SURGesag

This command is used to initialize the surge/trap wave function.

Syntax

INITiate[:IMMediate]:SURGesag

Example

INIT:SURG

Chapter11 PARallel Subsystem

PARallel:ROLE <role>

This command sets and queries the power supply to single, slave or master mode in the parallel operation.

Syntax

PARallel:ROLE <role>

Argument

<CPD>

SINGle|SLAVe|MASTer

Query syntax

PARallel:ROLE?

Returns

SINGle|SLAVe|MASTer

*RST

SINGle

Example

PAR:ROLE SLAV //Set the machine to slave mode.

PAR:ROLE? //Query the parallel role of the present instrument.

PARallel:NUMBer <number>

This command sets and queries the total instrument number in the parallel operation, and the setting range is 2-16.

Syntax

PARallel:NUMBer <number>

Argument

<NR1>

Query syntax

PARallel:NUMBer?

Returns

<NR1>

Example

PAR:NUMB 3 //Set the total number of parallel machines to 3.

PAR:NUMB? //Query the total number of parallel machines.

PARallel:NODE:NUMBer?

This command is used to obtain the total number of nodes after the optical fiber is paralleled.

Syntax

PARallel:NODE:NUMBer?

Argument

<NR1>

1~64

Example

PAR:NODE:NUMB?

Chapter12 TRIGger Subsystem

TRIGger:LIST:SOURce <source>

This command is used to set and query the trigger source of the LIST function.

Syntax

TRIGger:LIST:SOURce <source>

Argument

<CPD>

IMMediate|MANual|BUS|TRIG1|TRIG2

Query syntax

TRIGger:LIST:SOURce?

Returns

IMMediate|MANual|BUS|TRIG1|TRIG2

Example

TRIG:LIST:SOUR MAN //The trigger source of LIST function is selected as manual panel trigger.

TRIGger:SWEep:SOURce <CPD>

This command is used to set and query the trigger source of SWEEP function.

Syntax

TRIGger:SWEep:SOURce <CPD>

Argument

<CPD>

IMMediate|MANual|BUS|TRIG1|TRIG2

Query syntax

TRIGger:SWEep:SOURce?

Returns

IMMediate|MANual|BUS|TRIG1|TRIG2

Example

TRIG:SWE:SOUR MAN

TRIGger:SURGesag:SOURce <CPD>

This command is used to set and query the trigger source of the surge/trap wave function.

Syntax

TRIGger:SURGesag:SOURce <CPD>

Argument

<CPD>

IMMediate|MANual|BUS|TRIG1|TRIG2

Query syntax

TRIGger:SURGesag:SOURce?

Returns

IMMediate|MANual|BUS|TRIG1|TRIG2

Example

TRIG:SURG:SOUR MAN

TRIGger:SCOPe:SOURce <CPD>

This command is used to set and query the trigger source of the oscilloscope function.

Syntax

TRIGger:SCOPe:SOURce <CPD>

Argument

<CPD>

<1-6>|TRIG1|TRIG2

Query syntax

TRIGger:SCOPe:SOURce?

Example

TRIG:SCOP:SOUR TRIG1

TRIGger:SCOPe:MODE <CPD>

This command is used to set and query the mode of the oscilloscope function.

Syntax

TRIGger:SCOPe:MODE <CPD>

Argument

<CPD>

AUTO|NORMal

Query syntax

TRIGger:SCOPe:MODE?

Returns

AUTO|NORMal

Example

TRIG:SCOP:MODE AUTO

TRIGger:SCOPe:SLOPe <CPD1>

This command is used to set and query the slope of the oscilloscope function.

Syntax

TRIGger:SCOPe:SLOPe <CPD1>

Argument:

<CPD>

RISE|FALL|BOTH

Query syntax

TRIGger:SCOPe:SLOPe?

Returns

RISE|FALL|BOTH

Example

TRIG:SCOP:SLOP BOTH

TRIGger:FORCe

This command is used to trigger.

Syntax

TRIGger:FORCe

Example

TRIGger:FORCe

Chapter13 ARB Subsystem

[ARB:]LIST:STATe?

This command is used to query the running status of the list: IDLE: stop, WTG: waiting for trigger, ACTive: running.

Syntax

[ARB:]LIST:STATe?

Example

LIST:STAT?

[ARB:]LIST:REPeat <NR1>

This command is used to set and query the number of cycles that the list runs.

Syntax

[ARB:]LIST:REPeat <NR1>

Argument:

<NR1>|MINimum|MAXimum|DEFault

*RST

0

Query syntax

[ARB:]LIST:REPeat?

Returns

<NR1>

Example

LIST:REP 10

[ARB:]LIST:TERMinate <CPD>

This command is used to set and query the way to end the list operation. There are three ways to end:

NORMal: after the list runs, it will automatically return to normal.

LAST: keep the output of the last step after the list runs.

OFF: turn off the output after the list operation is over and still in the list mode.

Syntax

[ARB:]LIST:TERMinate <CPD>

Argument

<CPD>

NORMal|LAST|OFF

Query syntax

[ARB:]LIST:TERMinate?

Returns

NORMal|LAST|OFF

*RST

OFF

Example

LIST:TERM LAST

[ARB:]LIST:RUNTime:STATe?

This command is used to query the runtime information of List, in which loop the list is presently running, and at which step it is presently running.

Return value: <NR1>,<NR1> The first is the number of cycles presently running, and the second is the position of the present running step.

Syntax

[ARB:]LIST:RUNTime:STATe?

Returns

<NR1>,<NR1>

Example

LIST:RUNTime:STATe?

[ARB:]LIST:RCL <string>

This command is used to call back the saved list file and query the file name presently called.

Syntax

[ARB:]LIST:RCL <string>

Query syntax

[ARB:]LIST:RCL?

Example

LIST:RCL test.csv

[ARB:]LIST:STEP:COUNt <NR1>

This command is used to set and query the total number of steps in the list.

Syntax

[ARB:]LIST:STEP:COUNt <NR1>

Argument

<NR1>|MINimum|MAXimum|DEFault

Query syntax

[ARB:]LIST:STEP:COUNt?

Returns

<NR1>

*RST

10

Example

LIST:STEP:COUN 10

[ARB:]LIST:CLEar

This command is used to clear the list configuration.

Syntax

[ARB:]LIST:CLEar

Example

LIST:CLE

[ARB:]LIST:STEP <NR1>,<string>

This command is used to configure and query the list step parameters.

<NR1>: step index

<string>: the parameters are separated by commas, and the specific order is as follows:

- Three phase

A_Vac: AC voltage of A Phase

B_Vac: AC voltage of B Phase

C_Vac: AC voltage of C Phase

A_Vac_slew: AC voltage slope of phase A

B_Vac_slew: AC voltage slope of phase B

C_Vac_slew: AC voltage slope of phase C

A_Vdc: DC voltage of A Phase

B_Vdc : DC voltage of B Phase

C_Vdc : DC voltage of C Phase

A_Vdc_slew: DC voltage slope of phase A

B_Vdc_slew: DC voltage slope of phase B

C_Vdc_slew: DC voltage slope of phase C

Phase mode: CONTinues|PHASe

A_start: starting angle of phase A

ba_diff: phase difference of BA

ba_diff_slew: Phase differenct slew rate of BA

ca_diff : phase difference of CA

ca_diff_slew : Phase differenct slew rate of CA

Frequency: Frequency

Frequency_slew: frequency slope

A_wave: waveform type of phase A

B_wave: waveform type of phase B

C_wave: waveform type of phase C

Step Jump para1: step jump mode, TIME|TRIGger|PHASe

Step Jump para2: if the para1 is set to TIME, this parameter is time value. If the para1 is set to TRIGger, this parameter is IMMe or PHASe. And if para1 is set to PHASe, this parameter is angle value.

Step Jump para3: if para1 is set to TRIGger and para2 is set to PHASe, this parameter is angle value.

Trig_out: output a trigger signal.

- Single mode/ Reverse mode：

Vac : AC voltage

Vac_slew : AC voltage slope

Vdc : DC voltage

Vdc_slew : DC voltage slope

Phase mode: CONTinues|PHASe

Start: start angle

Frequency : frequency

Frequency_slew : frequency slope

wave : waveform type

Step Jump para1: step jump mode, TIME|TRIGger|PHASe

Step Jump para2: if the para1 is set to TIME, this parameter is time value. If the para1 is set to TRIGger, this parameter is IMMe or PHASe. And if para1 is set to PHASe, this parameter is angle value.

Step Jump para3: if para1 is set to TRIGger and para2 is set to PHASe, this parameter is angle value.

Trig_out: output a trigger signal.

Syntax

[ARB:]LIST:STEP <NR1>,<string>

Query syntax

[ARB:]LIST:STEP? <NR1>

Example

list:step
-1,"10,10,10,100,100,100,0.5,0.5,0.5,20,20,20,PHASe,356.9,30,100,40,120,100,101,Sine,Sine,Sine,TRIG,PHASE,55,OFF"

//single phase mode

list:step
-1,"10,10,10,100,100,100,0.5,0.5,0.5,20,20,20,PHASe,356.9,30,100,40,120,100,101,Sine,Sine,Sine,TRIG,PHASE,55,OFF"

//three phase mode

[ARB:]LIST:STEP:ITEM <NR1>,<NR2>,<string>

This command is used to configure and query the phase parameters of a certain step in the list.

<NR1>: Step index , [1,100]

<NR2>:

Note: The following are all the parameter IDs of the three-phase mode, those marked with "single-phase" indicate that they can be used to set and query the corresponding parameter values in the single-phase mode.

A phase AC voltage ID: 0 (single-phase)

B phase AC voltage ID: 1

C phase AC voltage ID: 2

A phase AC voltage slope ID: 3 (single-phase)

B phase AC voltage slope ID: 4

C phase AC voltage slope ID: 5

A phase DC voltage ID: 6 (single-phase)

B phase DC voltage ID: 7

C phase DC voltage ID: 8

A phase DC voltage slope ID: 9 (single-phase)

B phase DC voltage slope ID: 10

C phase DC voltage slope ID: 11

Phase mode ID: 12 (single-phase)(PHASe|CONTinues)

A Phase start angle ID: 13 (single-phase)

B Phase start angle ID: 14(Reserved)

C Phase start angle ID: 15(Reserved)

A Phase frequency ID: 16 (single-phase)

B Phase frequency ID: 17(Reserved)

C Phase frequency ID: 18(Reserved)

A Phase frequency slope ID: 19 (single-phase)

B Phase frequency slope ID: 20(Reserved)

C Phase frequency slope ID: 21(Reserved)

A Phase waveform type ID: 22 (single-phase)

B Phase waveform type ID: 23

C Phase waveform type ID: 24

Step jump 1 ID: 25 (single-phase)(TIME|TRIGger|PHASe)

Step jump 2 ID: 26 (single-phase)(When Step jump 1 ID is TIME, this parameter is the time value; when Step jump 1 ID is TRIGger, this parameter can be IMME/PHASe; when Step jump 1 ID is PHASe, this parameter is the angle value)

Trigger Pulse ID: 27 (single-phase)(OFF|ON)

BA angle ID 28

BA angle slope ID 29

CA angle ID 30

CA angle slope ID 31

Step jump 3 ID 32 (single-phase)(This parameter is meaningful only when jump Step jump 1 ID is TRIGger and Step jump 2 ID is PHASe, which is the angle value)

<string>: This parameter is the corresponding parameter value for the ID specified by NR2.

Syntax

LIST:STEP:ITEM <NR1>,<NR2>,<string>

Query syntax

LIST:STEP:ITEM? <NR1>,<item>

Example

LIST:STEP:ITEM 1,0,"100" //Set A-phase AC voltage to 100V in step 1

LIST:STEP:ITEM 1,22,"Sine" //Set the A-phase waveform type to Sine in step1

LIST:STEP:ITEM? 1,0 //Query A-phase AC voltage value

[ARB:]LIST:STEP:ITEM? <NR1>,<item>

This command is used to query the phase parameters of a certain step in the list.

Syntax

[ARB:]LIST:STEP:ITEM? <NR1>,<item>

Argument:

<NR1>,<item>

Returns:

<NRf+>

Example

LIST:STEP:ITEM? 1,6

[ARB:]LIST:NAME?

This command is used to query the path name of the presently opened list file.

Syntax

[ARB:]LIST:NAME?

Example

LIST:NAME?

[ARB:]LIST:SAVe <filename>

This command is used to save the file to the /mnt/emm_user/user_data/sys/list/ directory. If the file name already exists, the present file will be overwritten, if it does not exist, it will be saved as a new file.

Syntax

[ARB:]LIST:SAVe <filename>

Example

LIST:SAVe “123”

[ARB:]LIST:CONFigure

This command is used to make the presently configured list parameters take effect.

Syntax

[ARB:]LIST:CONFigure

Example

LIST:CONF

[ARB:]LIST:CREate

This command is used to create a list.

Syntax

[ARB:]LIST:CREate

Example

LIST:CRE

[ARB:]LIST:FILE:NUMBer?

This command is used to query the number of list files under the present device type.

Syntax

[ARB:]LIST:FILE:NUMBer?

Example

LIST:FILE:NUMB?

[ARB:]LIST:FILE:NAME? <index>

This command is used to query the name of a list file according to the index.

Syntax

[ARB:]LIST:FILE:NAME? <index>

Argument:

<NR1>

Example

LIST:FILE:NAME? 1

[ARB:]LIST:FILE:DELete <filename>

This command is used to delete the list file.

Syntax

[ARB:]LIST:FILE:DELete <filename>

Argument

<CPD>

Example

LIST:FILE:DEL "123"

[ARB:]LIST:STEP:DELete <NR1>

This command is used to delete the list step.

Syntax

[ARB:]LIST:STEP:DELete <NR1>

Argument

<NR1>

Example

LIST:STEP:DELete 1 //delete the step1

[ARB:]LIST:STEP:EDIT <NR1>,<string>

This command is used to edit the list step parameter.

Note: This command only modifies the parameters of the currently existing steps, and does not add new steps.

Syntax

[ARB:]LIST:STEP:EDIT <NR1>,<string>

Argument

<NR1>: step number

<string>: step parameters

Example

list:step:edit 1,"10,10,10,100,100,100,0.5,0.5,0.5,20,20,20,PHASe,356.9,30,100,40,120,100,101,Sine,Sine,Sine,TRIG,PHASE,55,OFF" //edit the step parameters for three phase

list:step:edit 1,"9.2,100,21,20,PHASe,359.9,100,100,Sine,TRIG,phase,123,ON"

//edit the step parameters for single phase

LIST:STEP:JUMP <NR1>

The command is used to set the step number of the loop skip, for example, when set to 2, the subsequent loop skips the previous two steps and starts from step 3.

NR1: the minimum value is 0, i.e. all steps without skipped steps are looped; the maximum value is less than the total number of steps.

Syntax

LIST:STEP:JUMP <NR1>

Argument

<NR1>: step number

Query Syntax

LIST:STEP:JUMP? [MAXimum|MINimum]

Example

LIST:STEP:JUMP 2

Chapter14 CONFigurate IO Subsystem

[CONFigurable:]IO:SELect <NR1>

This command is used to select and query IO.

Syntax

[CONFigurable:]IO:SELect <NR1>

Argument

<NR1>

[1,7]

Query syntax

[CONFigurable:]IO:SELect?

Returns

<NR1>

Example

IO:SEL 7

[CONFigurable:]IO:REVErse <NR1>,<CPD>

This command is used to set and query whether the IO port signal is reversed. The first parameter is the IO number, and the second parameter is the reverse state.

Syntax

[CONFigurable:]IO:REVErse <NR1>,<CPD>

Argument

<NR1>,<CPD>

[1,7],0|OFF|1|ON

Query syntax

[CONFigurable:]IO:REVErse? <NR1>

Returns

CPD

Example

IO:REVE 7,0

[CONFigurable:]IO:TYPE <NR1>,<CPD>

This command is used to set and query the IO type.

Note:

1: LIVing|LATCh

2: PSCLear

3: PSSTate

4: SYIN|SYOut

5: OFFState

6: TIN1|TOUT1

7: TIN2|TOUT2

Only specific IO can be configured for the above special functions. IORD|IOWR all IO can be configured.

Syntax

[CONFigurable:]IO:TYPE <NR1>,<CPD>

Argument

NR1: [1,7]

CPD:

LIVing|LATCh|PSCLear|PSSTate|SYIN|SYOut|OFFState|TIN1|TIN2|TOUT1|TOUT2|IORD|IOWR

Query syntax

[CONFigurable:]IO:TYPE? <NR1>

Returns

CPD

Example

IO:TYPE 1,LIV

[CONFigurable:]IO:TOUT:SOURce <CPD1>,<CPD2>

This command is used to set and query the trig source of IO-6 and IO-7 and whether the function is on or off.

Note: this command should be used in conjunction with IO:SELect, that is, select IO-6 or IO-7, and then set the trigger source.

Syntax

IO:TOUT:SOURce <CPD1>,<CPD2>

Argument

CPD1:AC|DC|FREQuency|LIST

CPD2:0|OFF|1|ON

Query syntax

IO:TOUT:SOURce? <CPD>

Returns

<CPD>

Example

IO:TOUT:SOURce ON

[CONFigurable:]IO:STATe <NR1>,<CPD>

When IO is set to output mode, it can output high level (False) or low level (True). This command is used to set and query the output level status, 0 means FALSE, 1 means True.

Syntax

[CONFigurable:]IO:STATe <NR1>,<CPD>

Argument

NR1:[1,7]

CPD:0|OFF|1|ON

Query syntax

[CONFigurable:]IO:STATe?

Example

IO:STATe 1,ON

Chapter15 SWEep Subsystem

SWEep:VOLTage:STARt <NRf+>[,<NRf+>,<NRf+>]

This command is used to set and query the start voltage of Sweep.

<NRf+>: initial voltage setting value, or initial voltage MINimum or MAXimum or DEFault setting

Note:

In single-phase (including AC, ACDC, DC, DCAC), there is only one parameter; in three-phase, three parameters must be taken.

Syntax

SWEep:VOLTage:STARt <NRf+>[,<NRf+>,<NRf+>]

Argument

<NRf+>[,<NRf+>,<NRf+>]

*RST

0

Query syntax

SWEep:VOLTage:STARt? [A|B|C][MAXimum|MINimum|DEFault]

Returns

<NR2>

Example

SWEep:VOLTage:STARt 100 //Set the starting voltage of sweep to 100V.

SWEep:VOLTage:STARt? //Query the setting value of sweep starting voltage.

SWEep:VOLTage:STARt? MAX //Query the maximum value of sweep initial Voltage.

SWEep:VOLTage:STOP <NRf+>[,<NRf+>,<NRf+>]

This command is used to set and query the stop voltage of sweep.

<NRf+>: stop voltage setting value, or stop voltage MINimum or MAXimum or DEFault setting.

Note:

In single-phase (including AC, ACDC, DC, DCAC), there is only one parameter; in three-phase, three parameters must be taken.

Syntax

SWEep:VOLTage:STOP <NRf+>[,<NRf+>,<NRf+>]

Argument

<NRf+>[,<NRf+>,<NRf+>]

Query syntax

SWEep:VOLTage:STOP? [A|B|C][MAXimum|MINimum|DEFault]

Returns

<NR2>

Example

SWEep:VOLTage:STOP 90 //Set the stop voltage of sweep to 90V.

SWEep:VOLTage:STOP? //Query the presently set stop voltage.

SWEep:VOLTage:STOP? MAX //Query the maximum value of stop voltage.

SWEep:VOLTage:STEP <NRf+>[,<NRf+>,<NRf+>]

This command is used to set and query the step voltage of sweep.

<NRf+>: step voltage setting value, or step MINimum or MAXimum or DEFault setting.

Note:

In single-phase (including AC, ACDC, DC, DCAC), there is only one parameter;

in three-phase, three parameters must be taken.

Syntax

SWEep:VOLTage:STEP <NRf+>[,<NRf+>,<NRf+>]

Argument

<NRf+>[,<NRf+>,<NRf+>]

*RST

0

Query syntax

SWEep:VOLTage:STEP? [A|B|C][MAXimum|MINimum|DEFault]

Returns

<NR2>

Example

SWEep:VOLTage:STEP 1 // Set the step voltage of sweep to 1V.

SWEep:VOLTage:STEP? // Query the presently set step voltage.

SWEep:VOLTage:STEP? MAX // Query the maximum value of step voltage.

SWEep:FREQ:STARt <NRf+>[,<NRf+>,<NRf+>]

This command is used to set and query the start frequency of sweep.

<NRf+>: the set value of the start frequency, or the MINimum or MAXimum or DEFault setting of start frequency.

Note:

In single-phase (including AC, ACDC, DC, DCAC), there is only one parameter; in three-phase, three parameters must be taken.

Syntax

SWEep:FREQ:STARt <NRf+>[,<NRf+>,<NRf+>]

Argument

<NRf+>[,<NRf+>,<NRf+>]

*RST

0

Query syntax

SWEep:FREQ:STARt? [A|B|C][MAXimum|MINimum|DEFault]

Returns

<NR2>

Example

SWEep:FREQ:STARt 1 //Set the start frequency of sweep to 1Hz.

SWEep:FREQ:STARt? //Query the presently set start frequency.

SWEep:FREQ:STARt? MAX //Query the maximum value of the start frequency.

SWEep:FREQ:STOP <NRf+>[,<NRf+>,<NRf+>]

This command is used to set and query the stop frequency of sweep.

<NRf+>: stop frequency setting value, or stop frequency MINimum or MAXimum or DEFault setting.

Note:

In single-phase (including AC, ACDC, DC, DCAC), there is only one parameter; in three-phase, three parameters must be taken.

Syntax

SWEep:FREQ:STOP <NRf+>[,<NRf+>,<NRf+>]

Argument

<NRf+>[,<NRf+>,<NRf+>]

*RST

0

Query syntax

SWEep:FREQ:STOP? [A|B|C][MAXimum|MINimum|DEFault]

Returns

<NR2>

Example

SWEep:FREQ:STOP 1 //Set the stop frequency of sweep to 1Hz.

SWEep:FREQ:STOP? //Query the presently set stop frequency.

SWEep:FREQ:STOP? MAX //Query the maximum value of the stop frequency.

SWEep:FREQ:STEP <NRf+>[,<NRf+>,<NRf+>]

This command is used to set and query step frequency of sweep.

<NRf+>: step frequency setting value, or step frequency MINimum or MAXimum or DEFault setting.

Note:

In single-phase (including AC, ACDC, DC, DCAC), there is only one parameter; in three-phase, three parameters must be taken.

Syntax

SWEep:FREQ:STEP <NRf+>[,<NRf+>,<NRf+>]

Argument

<NRf+>[,<NRf+>,<NRf+>]

*RST

0

Query syntax

SWEep:FREQ:STEP? [A|B|C][MAXimum|MINimum|DEFault]

Returns

< NR2>

Example

SWEep:FREQ:STEP 1 //Set the step frequency of sweep to 1Hz.

SWEep:FREQ:STEP? // Query the presently set step frequency.

SWEep:FREQ:STEP? MAX //Query the maximum value of step frequency.

SWEep:TIMe:STEP <NRf+>[,<NRf+>,<NRf+>]

This command is used to set and query the single-step execution time of sweep. It is valid when the mode is TIMe.

<NRf+>: step time setting value, or step time MINimum or MAXimum or DEFault

setting.

Note:

In single-phase (including AC, ACDC, DC, DCAC), there is only one parameter; in three-phase, three parameters must be taken.

Syntax

SWEep:TIMe:STEP <NRf+>[,<NRf+>,<NRf+>]

Argument

<NRf+>[,<NRf+>,<NRf+>]

*RST

0

Query syntax

SWEep:TIMe:STEP? [A|B|C][MAXimum|MINimum|DEFault]

Returns

<NR2>

Example

SWEep:TIMe:STEP 1 //Set the step time of sweep to 1S.

SWEep:TIMe:STEP? //Query the presently set step time.

SWEep:TIMe:STEP? MAX //Query the maximum value of step time.

SWEep:MODe <mode>

This command is used to set and query the switching mode between the single steps of the sweep function, and it can be set to TIMe or TRIG mode.

Syntax

SWEep:MODe <mode>

Argument

TIMe|TRIG

*RST

TIMe

Query syntax

SWEep:MODe?

Returns

TIMe|TRIG

Example

SWEep:MODe TIMe //Set sweep mode to TIMe.

SWEep:PRIority<priority>

This command is used to set and query the priority of sweep.

<priority>: sweep priority VOLT|FREQ|VF

Syntax

SWEep:PRIority<priority>

Argument

VOLT|FREQ|VF

*RST

VOLT

Query syntax

SWEep:PRIority?

Returns

VOLT|FREQ|VF

Example

SWEep:PRIority VOLT //Set the priority of sweep to VOLT.

SWEep:PRIority?

SWEep:WAVeform <type>

This command is used to set and query the sweep waveform.

<type> : sweep wave type SINe|SQUAre|SAW|Triangle

Syntax

SWEep:WAVeform <type>

Argument

SINe|SQUAre|SAW|Triangle

*RST

SINe

Query syntax

SWEep:WAVeform?

Returns

SINe|SQUAre|SAW|Triangle

Example

SWEep:WAVeform SINe //Set sweep waveform to sine.

SWEep:WAVeform? //Query sweep waveform type.

SWEep:FINish <mode>

This command is used to set and query the end status of sweep.

<type>: sweep wave type NORMal|LAST|OFF

Syntax

SWEep:FINish <mode>

Argument

NORMal|LAST|OFF

*RST

NORMal

Query syntax

SWEep:FINish?

Returns

NORMal|LAST|OFF

Example

SWEep:FINish NORMal //Set the end state of sweep to NORMal.

SWEep:FINish? //Query the end status of sweep.

SWEep:TRIG:SOURce <mode>

This command is used to set the trigger source of Sweep function.

Syntax

SWEep:TRIG:SOURce <mode>

Argument

MANual|BUS|TRIG1|TRIG2

*RST

MANual

Query syntax

SWEep:TRIG:SOURce?

Returns

MANual|BUS|TRIG1|TRIG2

Example

SWEep:TRIG:SOURce MANual

SWEep:STEP:REPeat <NR1>

Set the number of sweep repetitions, set 0 for infinite loop.

Syntax

SWEep:STEP:REPeat <NR1>

Argument

1-65535

Query syntax

SWEep:STEP:REPeat? [MAXimum|MINimum]

Example

SWEep:STEP:REPeat <NR1>

SWEep:SAVe <string>

Save the sweep file.

Syntax

SWEep:SAVe <string>

Argument

<string>

Example

SWEep:SAVe “sweep1.ini”

SWEep:RECall <string>

Recall the sweep file.

Syntax

SWEep:RECall <string>

Argument

<string>

Example

SWEep:RECall “sweep1.ini”

SWEep:STATe?

This command is used to query the running status of sweep.

Syntax

SWEep:STATe?

Example

SWE:STAT?

Chapter16 SURGesag Subsystem

SURGesag:MODE <mode>

This command is used to set and query the surge/trap wave generating mode.

<mode>: surge/trap wave operating mode PERiod|TRIG.

PERiod generates surge/trap wave periodically, TRIG generates surge/trap wave when triggered.

Syntax

SURGesag:MODE <mode>

Argument

PERiod|TRIG

Query syntax

SURGesag:MODE?

Returns

PERiod|TRIG

Example

SURGesag:MODe PERiod //Set surge/trap wave generating mode to periodic mode.

SURGesag:MODe? // Query the surge/trap wave generating mode.

SURGesag:PHASe:STARt <NRf+>

This command is used to set and query the initial phase of the surge/trap wave.

<NRf+>: start phase setting value, or start phase MINimum or MAXimum or DEFault setting.

Syntax

SURGesag:PHASe:STARt <NRf+>

Argument

<NRf+>

Unit

Degree (°)

*RST

0

Query syntax

SURGesag:PHASe:STARt? [MAXimum|MINimum|DEFault]

Returns

<NRf+>

Example

SURGesag:PHASe:STARt 90 //Set the initial phase of the surge/trap wave to 90°.

SURGesag:PHASe:STARt? //Query the presently set starting phase.

SURGesag:PHASe:STARt? MAX // Query the maximum value of the starting phase.

SURGesag:PHASe:WIDTh <NRf+>

This command is used to set and query the surge/trap wave angle width.

<NRf+>: angle width setting value

Syntax

SURGesag:PHASe:WIDTh <NRf+>

Argument

<NRf+>

Unit

Degree (°)

*RST

0

Query syntax

SURGesag:PHASe:WIDTh? [MAXimum| MINimum|DEFault]

Returns

<NRf+>

Example

SURGesag:PHASe:WIDTh 320.0 //Set the angular width to 320°.

SURGesag:PHASe:WIDTh? //Query the set value of angle width.

SURGesag:PHASe:WIDTh? MIN // Query the minimum value of the angle width.

SURGesag:ACTion <action>

This command is used to set and query the trigger action of the surge/trap wave.

<action>: trigger action setting mode IMMediate|PHASe.

Syntax

SURGesag:ACTion <action>

Argument

IMMediate|PHASe

*RST

IMMediate

Query syntax

SURGesag:ACTion?

Returns

IMMediate|PHASe

Example

SURGesag:ACTion IMMediate //Set the present trigger action to IMMediate.

SURGesag:ACTion IMMediate?

SURGesag:TRIG:SOURce <source>

This command is used to set the trigger source of surge/sag function.

Syntax

SURGesag:TRIG:SOURce <source>

Argument

MANual|BUS|TRIG1|TRIG2

Query syntax

SURGesag:TRIG:SOURce?

Returns

MANual|BUS|TRIG1|TRIG2

Example

SURGesag:TRIG:SOURce TRIG1

SURGesag:SYMMetry <bool>

This command is used to set and query the symmetry mode switch of the surge/trap wave.

Syntax

SURGesag:SYMMetry <bool>

Argument

0 | OFF | 1 | ON

*RST

OFF

Query syntax

SURGesag:SYMMetry?

Returns

0 | OFF | 1 | ON

Example

SURGesag:SYMMetry 1 //Turn on symmetry mode.

SURGesag:SYMMetry? //Query the switch status of symmetric mode.

SURGesag:REPeat:COUNt <NR1>

This command is used to set and query the number of continuous surge/trap waves.

<NR1+>: set value of continuously generated surge/trap waves.

Syntax

SURGesag:REPeat:COUNt <NR1>

Argument

<NR1>

*RST

1

Query syntax

SURGesag:REPeat:COUNt?

Returns

<NR1>

Example

SURGesag:REPeat:COUNt 5 //Set the number of continuous surge/trap waves to 5.

SURGesag:PERiod:COUNt <NR1>

This command is used to set and query the period interval of the surge/trap wave occurrence.

<NR1+>: the period interval setting value of the surge/trap wave occurrence.

Syntax

SURGesag:PERiod:COUNt <NR1>

Argument

<NR1>

*RST

1

Query syntax

SURGesag:PERiod:COUNt?

Returns

<NR1>

Example

SURGesag: PERiod:COUNt 5 //set the period interval of the surge/trap wave occurrence to 5.

SURGesag:VALue:SELect <mode>

This command is used to set and query the setting mode of the surge/trap wave droping value.

<mode>: the expression mode of the surge/trap wave droping value PERCent|SETTing

Syntax

SURGesag:VALue:SELect <mode>

Argument

PERCent|SETTing

*RST

PERCent

Query syntax

SURGesag:VALue:SELect <mode>

Returns

PERCent|SETTing

Example

SURGesag:VALue:SELect PERcent //Set the mode of the drop value to percent

SURGesag:VALue:SELect?

SURGesag:VALue:PERCent <NRf+>

This command is used to set and query the percentage of the drop value of the surge/trap wave.

<NRf+>: dropping percentage setting value.

Syntax

SURGesag:VALue:PERCent <NRf+>

Argument

<NRf+>

*RST

0

Query syntax

SURGesag:VALue:PERCent?

Returns

<NRf+>

Example

SURGesag:VALue:PERCent 10 //Set the drop percentage to 10%

SURGesag:VALue:PERCent? //Query the set drop percentage

SURGesag:VALue <NRf+>

This command is used to set and query the drop value of the trap wave.

<NRf+> drop set value

Syntax

SURGesag:VALue <NRf+>

Argument

<NRf+>

*RST

0

Query syntax

SURGesag:VALue?

Returns

<NRf+>

Example

SURGesag:VALue 20 //Set the drop value to 20V

SURGesag:VALue? //Query the set value of the drop value

SURGesag:VALue? MAX //Query the maximum value of the drop set value.

SURGesag:PHASe:ENABle <A|B|C|AB|AC|BC|ABC>

This command is used to set and query the phase of surge/sag, available only in three-phase mode.

Syntax

SURGesag:PHASe:ENABle <A|B|C|AB|AC|BC|ABC>

Argument

<A|B|C|AB|AC|BC|ABC>

Query syntax

SURGesag:PHASe:ENABle?

Returns

<NRf+>

Example

SURGesag:PHASe:ENABle ABC

SURGesag:ANGLe:ENABle <SYNC|SPEC>

This command is used to set and query the synchronization method of dropping between three phases, SYNC means synchronous dropping, SPEC means dropping at the respective specified phase. Only available in three-phase mode.

Syntax

SURGesag:ANGLe:ENABle < SYNC|SPEC >

Argument

<SYNC|SPEC>

Query syntax

SURGesag:ANGLe:ENABle?

Returns

<NRf+>

Example

SURGesag:ANGLe:ENABle SYNC

SURGesag:SAVe <string>

Save the surge/sag file.

Syntax

SURGesag:SAVe <string>

Argument

<string>

Example

SURGesag:SAVe “sweep1.ini”

SURGesag:RECall <string>

Recall the surge/sag file.

Syntax

SURGesag:RECall <string>

Argument

<string>

Example

SURGesag:RECall “sweep1.ini”

SURGesag:STATe?

This command is used to query the running status of the surge/trap wave.

Syntax

SURGesag:STATe?

Example

SURGesag:STATe? //Query the running status of the surge/trap wave.

Chapter17 SCOPe Subsystem

SCOPe:AUTO

This is an automatic setting command of the oscilloscope.

Syntax

SCOPe:AUTO

Example

SCOPe:AUTO

SCOPe:RUN

This is an operation command of the oscilloscope.

Syntax

SCOPe:RUN

Example

SCOPe:RUN

SCOPe:SINGle

This command is used to capture single-shot oscilloscope data.

Syntax

SCOPe:SINGle

Example

SCOPe:SINGle

SCOPe:STOP

This is a stop command of the oscilloscope.

Syntax

SCOPe:STOP

Example

SCOPe:STOP

SCOPe:TIMebase:SCALe <NRf>

This command is used to set and query the time scale of the oscilloscope, unit: s. <0.001-1.0>

Syntax

SCOPe:TIMebase:SCALe <NRf>

Argument

[MINimum|MAXimum]

Query syntax

SCOPe:TIMebase:SCALe?

Returns

<NRf>

Example

SCOPe:TIMebase:SCALe 0.5

SCOPe:VOLTage:SCALe <NRf>

This command is used to set and query the voltage scale of the oscilloscope, unit: V.

Syntax

SCOPe:VOLTage:SCALe <NRf>

Argument

[MINimum|MAXimum]

Query syntax

SCOPe:VOLTage:SCALe?

Returns

<NRf>

Example

SCOPe:VOLTage:SCALe 10

SCOPe:CURRent:SCALe <NRf>

This command is used to set and query the current scale of the oscilloscope, unit: A.

Syntax

SCOPe:CURRent:SCALe <NRf>

Argument

[MINimum|MAXimum]

Query syntax

SCOPe:CURRent:SCALe?

Returns

<NRf>

Example

SCOPe:CURRent:SCALe 30

SCOPe:TIMebase:DELay <NRf>

This command is used to set and query the trigger delay of the oscilloscope, unit: s.

Syntax

SCOPe:TIMebase:DELay <NRf>

Argument

[MINimum|MAXimum]

Query syntax

SCOPe:TIMebase:DELay?

Returns

<NRf>

Example

SCOPe:TIMebase:DELay 3

SCOPe:TRIGger:SOURce

This command is used to set the trigger source of surge/sag function.

Syntax

SCOPe:TRIGger:SOURce

Argument

MANual|BUS|TRIG1|TRIG2

Query syntax

SCOPe:TRIGger:SOURce?

Returns

MANual|BUS|TRIG1|TRIG2

Example

SCOPe:TRIGger:SOURce TRIG1

SCOPe:TRIGger:LEVel <NRf>

This command is used to set and query the trigger level of the oscilloscope.

Syntax

SCOPe:TRIGger:LEVel <NRf>

Argument

[MINimum|MAXimum]

Query syntax

SCOPe:TRIGger:LEVel?

Returns

<NRf>

Example

SCOPe:TRIGger:LEVel MIN

SCOPe:TRIGger:SLOPe <CPD>

This command is used to set and query the trigger edge of the oscilloscope.

Syntax

SCOPe:TRIGger:SLOPe <CPD>

Argument

<RiSE|FALL|BOTH>

Query syntax

SCOPe:TRIGger:SLOPe?

Returns

<RiSE|FALL|BOTH>

Example

SCOPe:TRIGger:SLOPe BOTH

SCOPe:TRIGger:MODE <CPD>

This command is used to set and query the trigger mode of the oscilloscope.

Syntax

SCOPe:TRIGger:MODE <CPD>

Argument

<AUTO|NORMal>

Query syntax

SCOPe:TRIGger:MODE?

Returns

<AUTO|NORMal>

Example

SCOPe:TRIGger:MODE AUTO

SCOPe:LINE:SELection

This command is used to set and query the curve displayed by the oscilloscope, up to 6 curves can be displayed.

Syntax

SCOPe:LINE:SELection

Argument

<1-6>,<Off|On|0|1>

Query syntax

SCOPe:LINE:SELection?

Example

SCOPe:LINE:SELection 1,ON

SCOPe:STATus?

Query the present status of the oscilloscope.

Syntax

SCOPe:STATus?

Returns

"Stop"|“Ready”|

“Roll”|“Auto”|“Trig'd”

Example

SCOPe:STATus?

SCOPe:RSTate?

Query the running status of the oscilloscope.

Syntax

SCOPe:RSTate?

Returns

“RUN”|“STOP”|

“SINGle”

Example

SCOPe:RSTate?

SCOPe:WAVeform:DATA?

This command is used to get the data of the oscilloscope.

Syntax

SCOPe:WAVeform:DATA?

Example

SCOPe:WAVeform:DATA?

SCOPe:RANGe:CATalog?

This command is used to get voltage and current scale range options.

Syntax

SCOPe:RANGe:CATalog?

Returns

2/5/10/20/50/100/200/500,0.2/0.5/1/2/5/10/20/50

Example

SCOPe:RANGe:CATalog?

SCOPe:RECord:LENGth <0.6|6|60|600>

Oscilloscope data points collected in one second, 0.6 that is 600 points, 600 that is 600000, with the time scale and sampling frequency, sampling frequency multiplied by the screen time scale to be less than or equal to the length of the record, such as, sampling frequency 6s, time scale 1s, the record length is set to 0.6kpts, then the collection of 100 points per second.

Syntax

SCOPe:RECord:LENGth <0.6|6|60|600>

Returns

<0.6|6|60|600>

Query syntax

SCOPe:RECord:LENGth?

Example

SCOPe:RECord:LENGth 6

SCOPe:SAMPle:MODE <NORMal|PEAK>

This command is used to set sampling mode of Oscilloscope.

NORMal: Acquisition of data points by equal interval mode

PEAK: Acquisition of data points by peak-to-peak mode

Syntax

SCOPe:SAMPle:MODE <NORMal|PEAK>

Argument

<NORMal|PEAK>

Query syntax

SCOPe:SAMPle:MODE?

Example

SCOPe:RECord:LENGth 6

SCOPe:DATA:TAG?

Queries the sample data identification of the oscilloscope.

Syntax

SCOPe:DATA:TAG?

Returns

Return: <NR1>,<NR1> ,<NR1> ,<NR1> ,<NR1>

The first one is the starting point of the display

The second one shows the end time point

The third one is the data start time point

The fourth is the data end time point

The fifth is the data trigger time point

Example

SCOPe:DATA:TAG?

Chapter18 STATus Commands

STATus:QUEStionable[:EVENt]?

This command can be used to read the value in query event register.The power supply will return a decimal number corresponding to the binary weighted sum of each unit digit of register. These digits will be latched. After executing this command, the query event register will be cleared.

Query syntax

STATus:QUEStionable[:EVENt]?

Arguments

None

Returns

<NR1>

Example

-> STAT:QUES?
<- 4140

Related syntax

STATus:QUEStionable:ENABle

The definition of query event enable register is as follows

Bit	7	6	5	4	3	2	1	0
Name	OT	FAN err	OPP	OCPpeak	OCPrms	UVPrms	OVPpeak	OVPrms
Value Bit Weight	128	64	32	16	8	4	2	1

STATus:QUEStionable:CONDition?

This command is used to read the value of query condition register.

This is a read-only register, which directly reflects those bits in the instrument's problem state that are set to 1 or 0, and will not be latched. After the command is executed, the value of the query condition register will not be cleared.

Query syntax

STATus:QUEStionable:CONDition?

Arguments

None

Returns

<NR1> This is a decimal value with a binary weighted sum.

STATus:QUEStionable:ENABle <NR1>

Sets the value of the enable register for the Questionable Status group. The enable register is a mask for enabling specific bits from the Operation Event register to set the QUES (questionable summary) bit of the Status Byte register. STATus:PRESet clears all bits in the enable register. *CLS does not clear the enable register, but does clear the event register.

Syntax

STATus:QUEStionable:ENABle <state>

Arguments

0-0x7fffffff

Default value

0|OFF

Example

STAT:QUES:ENAB 128

Query syntax

STATus:QUEStionable:ENABle?

Returns

<NR1>

STATus:QUEStionable:NTRansition <NR1>

Sets the value of the NTR (Negative-Transition) registers. These registers serve as a polarity filter between the Questionable Condition and Questionable Event registers. When a bit in the NTR register is set to 1, then a 1-to-0 transition of the corresponding bit in the Questionable Condition register causes that bit in the Questionable Event register to be set. STATus:PRESet sets all bits in the PTR registers and clears all bits in the NTR registers.

• If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Questionable Condition register sets the corresponding bit in the Questionable Event register.

• If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Questionable Condition register can set the corresponding bit in the Questionable Event register.

• The value returned is the binary-weighted sum of all enabled bits in the register.

Syntax

STATus:QUEStionable:NTRansition <NR1>

Arguments

0-0x7fffffff

Default value

0

Example

STAT:QUES:NTR 128

Query syntax

STATus:QUEStionable:NTRansition?

Returns

<NR1> This is a decimal value with a binary weighted sum.

STATus:QUEStionable:PTRansition

Sets the value of the PTR (Positive-Transition) registers. These registers serve as a polarity filter between the Questionable Condition and Questionable Event registers. When a bit in the PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Questionable Condition register causes that bit in the Questionable Event register to be set. STATus:PRESet sets all bits in the PTR registers and clears all bits in the NTR registers.

Syntax

STATus:QUEStionable:PTRansition <NR1>

Arguments

0-0x7fffffff

Default value

0

Example

STAT:QUES:PTR 128

Query syntax

STATus:QUEStionable:PTRansition?

Returns

<NR1> This is a decimal value with a binary weighted sum.

STATus:OPERation[:EVENt]?

This command can read the parameter from operation event register. After executing this order, operation event register is reset.

Query syntax

STATus:OPERation[:EVENt]?

Arguments

None

Returns

<NR1>

Related syntax

STATus:OPERation:ENABle

Bit definition of operation event register:

Bit	7	6	5	4	3	2	1	0
Name	n.u	n.u	n.u	n.u	n.u	WTG	LIST	CAL
Value Bit Weight						4	2	1

STATus:OPERation:CONDition?

This command can read the parameter from the operation condition register.

This is a read-only register, which directly reflects those bits in the operating state of the instrument are set to 1 or 0, and will not be latched. After the command is executed, the value of the query condition register will not be cleared.

Query syntax

STATus:OPERation:CONDition?

Arguments

None

Example

STATus:OPERation:CONDition?

Returns

<NR1> This is a decimal value with a binary weighted sum.

STATus:OPERation:ENABle <NR1>

Sets the value of the enable register for the Operation Status group. The enable register is a mask for enabling specific bits from the Operation Event register to set the OPER (operation summary) bit of the Status Byte register. STATus:PRESet clears all bits in the enable register. *CLS does not clear the enable register, but does clear the event register.

Syntax

STATus:OPERation:ENABle <NR1>

Arguments

0-0x7fffffff

Default value

0|OFF

Example

STATus:OPERation:ENABle 128

Query syntax

STATus:OPERation:ENABle?

Returns

<NR1>

STATus:OPERation:NTRansition<NR1>

Sets the value of the NTR (Negative-Transition) registers. These registers serve as a polarity filter between the Operation Condition and Operation Event registers. When a bit in the NTR register is set to 1, then a 1-to-0 transition of the corresponding bit in the Operation Condition register causes that bit in the Operation Event register to be set. STATus:PRESet sets all bits in the PTR registers and clears all bits in the NTR registers.

• If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Operation Condition register sets the corresponding bit in the Operation Event register.

• If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Operation Condition register can set the corresponding bit in the Operation Event register.

• The value returned is the binary-weighted sum of all enabled bits in the register.

Syntax

STATus:OPERation:NTRansition <NR1>

Arguments

0-0x7fffffff

Default value

0

Example

STATus:OPERation:NTRansition 128

Query syntax

STATus:OPERation:NTRansition?

Returns

<NR1>

STATus:OPERation:PTRansition

Sets the value of the PTR (Positive-Transition) registers. These registers serve as a polarity filter between the Operation Condition and Operation Event registers. When a bit in the PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Operation Condition register causes that bit in the Operation Event register to be set. STATus:PRESet sets all bits in the PTR registers and clears all bits in the NTR registers.

Syntax

STATus:OPERation:PTRansition <NR1>

Arguments

0-0x7fffffff

Default value

0

Example

STATus:OPERation:PTRansition 128

Query syntax

STATus:OPERation:PTRansition?

Returns

<NR1>

STATus:PRESet

This command is used to preset the value of the status register.

Syntax

STATus:PRESet

Example

STAT: PRES

Chapter19 WAVEform Subsystem

WAVEform[:IMMediate] <WAVEform>,[WAVEform],[WAVEform]

This command is used to set and query the waveform in Normal mode. For the waveform with parameters, the parameters are set to default values.

Syntax

WAVEform[:IMMediate] <WAVEform>,[WAVEform],[WAVEform]

Arguments

<"Sine"|"Square"|"Triangle"|"Saw"|"Trapezoid"|"Clipped-sine"|"DST-01(30)"|"filename">,

<"Sine"|"Square"|"Triangle"|"Saw"|"Trapezoid"|"Clipped-sine"|"DST-01(30)"|"filename">,

<"Sine"|"Square"|"Triangle"|"Saw"|"Trapezoid"|"Clipped-sine"|"DST-01(30)"|"filename">,

*RST

Sine

Query syntax

WAVEform?

Returns

<string>

Example

```
WAVE "Sine"                    // Set the waveform to Sine.
WAVE "Sine","Saw","Square"     // Set three channel waveforms separately, valid
                               under three-phase unbalanced or multi-channel.
WAVE?
```

PWAVeform[:IMMediate] <phase/chan>,<WAVEform>

This command is used to set and query the waveform of a phase/channel in Normal mode. For waveforms with parameters, the parameters are set to default values.

Syntax

PWAVeform[:IMMediate] <phase/chan>,<WAVEform>

Arguments

<A|B|C|CH1|CH2|CH3>,

<"Sine"|"Square"|"Triangle"|"Saw"|"Trapezoid"|"Clipped-sine"|"DST-01(30)"|"filen

ame">

*RST

Sine

Query syntax

PWAVeform? <phase/chan>

Returns

<string>

Example

PWAVE CH1,"Sine" // Set the channel 1 waveform of the current mode to Sine.

PWAVE A,"Sine" // Set the phase A waveform of the current mode to Sine.

PWAVE? A // Query the waveform of phase A.

PWAVE? CH1 // Query the waveform of channel 1.

WAVEform:EDIT:PARameter <phase/chan>,<percent1>

This command is used to set and query the waveform parameters of a phase/channel (for example, the percentage of clipping wave. For waveform types without parameters, this parameter is invalid).

Syntax

WAVEform:EDIT:PARameter <phase/chan>,<percent1>

Arguments

<A|B|C|CH1|CH2|CH3>,<0-0.5>|<0-0.25>

Query syntax

WAVEform:EDIT:PARameter? <phase/chan>

Returns

<NRF>

Example

WAVE:EDIT:PAR 0.25,0.25,0.25

WAVE:EDIT:PAR? // Query waveform parameters of all phases/channels.

WAVE:EDIT:PAR? A //Query the waveform parameters of phase A.

WAVE:EDIT:PAR? CH1 // Query the waveform parameters of channel 1.

WAVEform:EDIT:THD:CLEar

This command is used to clear the edited memory data of the THD wave.

Syntax

WAVEform:EDIT:THD:CLEar

Example

WAVE:EDIT:THD:CLE

WAVEform:EDIT:THD:IMPort <filename>

This command is used to import the existing THD wave file into the editing memory and query the name of the THD wave file imported from the external storage.

Syntax

WAVEform:EDIT:THD:IMPort <filename>

Arguments

"filename"

Query syntax

WAVEform:EDIT:THD:IMPort?

Returns

<string>

Example

WAVE:EDIT:THD:IMP "Thd01.csv"

WAVE:EDIT:THD:IMP?

WAVEform:EDIT:THD:FORMula <formula>

This command is used to set and query the distortion factor calculation method of the edited THD wave.

Syntax

WAVEform:EDIT:THD:FORMula <formula>

Arguments

<%F|%R>

Query syntax

WAVEform:EDIT:THD:FORMula?

Returns

<CPD>

Example

WAVE:EDIT:THD:FORM %F

WAVE:EDIT:THD:FORM?

WAVEform:EDIT:THD:DATA <order>,<string>

This command is used to set and query the THD value and phase parameter of a certain order of harmonics

Syntax

WAVEform:EDIT:THD:DATA <order>,<string>

Arguments

<2-50>,<"thd,phase">

Query syntax

WAVEform:EDIT:THD:DATA? <order>

Returns

<string>

Example

WAVE:EDIT:THD:DATA 2, "9.8,0.0"

WAVE:EDIT:THD:DATA? 2

WAVEform:EDIT:THD:SAVE:LOCal <filename>

This command is used to save the presently edited THD wave data to a local file, and specify the file name (if the file name is the same, it is regarded as modifying the present waveform, if the file name does not exist, it is regarded as creating a new file)

Syntax

WAVEform:EDIT:THD:SAVE:LOCal <filename>

Arguments

"filename"

Example

WAVE:EDIT:THD:SAVE:LOC "THD01.csv"

WAVEform:EDIT:THD:SAVE:UDISk <filename>

This command is used to save the presently edited THD wave data to a U disk file, and specify the file name (if the file name is the same, it is regarded as modifying the current waveform, if the file name does not exist, it is regarded as creating a new file)

Syntax

WAVEform:EDIT:THD:SAVE:UDISk <filename>

Arguments

"filename"

Example

WAVE:EDIT:THD:SAVE:UDIS "THD01.csv"

WAVEform:EDIT:USERdefine:CLEar

This command is used to clear all of userdefined waveform data.

Syntax

WAVEform:EDIT:USERdefine:CLEar

Example

WAVE:EDIT:USER:CLE

WAVEform:EDIT:USERdefine:IMPort <filename>

This command is used to import the data of the existing custom wave into the editing memory and query the file name of the imported userdefined wave.

Syntax

WAVEform:EDIT:USERdefine:IMPort <filename>

Arguments

"filename"

Query syntax

WAVEform:EDIT:USERdefine:IMPort?

Returns

<string>

Example

WAVE:EDIT:USER:IMP "USER01"

WAVE:EDIT:USER:IMP?

WAVEform:EDIT:USERdefine:MODE <symmtery>

This command is used to set and query the userdefined wave editing mode, where:

ASYM: The user can only edit the first 512 points, not symmetrical about the origin;

SYMM: The user can only edit the first 512 points, which are symmetrical about the origin;

FULL: users can customize and edit 1024 points in the entire cycle.

Syntax

WAVEform:EDIT:USERdefine:MODE <symmtery>

Arguments

<ASYM,SYMM,FULL>

Query syntax

WAVEform:EDIT:USERdefine:MODE?

Returns

<ASYM,SYMM,FULL>

Example

WAVE:EDIT:USER:MODE ASYM

WAVE:EDIT:USER:MODE?

WAVEform:EDIT:USERdefine:DATA <index>,<normalization>

Set and query the normalized point value of the point whose order is index.

Syntax

WAVEform:EDIT:USERdefine:DATA <index>,<normalization>

Arguments

<0-1023>,<0.0-1.0> (Note: The range of index varies with the presently set waveform editing mode (symmetrical attribute), FULL is 0-1023, and other modes are 0-511)

Query syntax

WAVEform:EDIT:USERdefine:DATA? <index>

Returns

<NR2>

Example

WAVE:EDIT:USER:DATA 1,0.5

WAVE:EDIT:USER:DATA? 1

WAVEform:EDIT:USERdefine:SAVE:LOCal <filename>

This command is used to save the presently edited custom wave data to a file, and specify the file name (if the file name is the same, it is regarded as modifying the current waveform, if the file name does not exist, it is regarded as creating a new file)

Syntax

WAVEform:EDIT:USERdefine:SAVE:LOCal <filename>

Arguments

"filename"

Example

WAVE:EDIT:USER:SAVE:LOC "USER01.csv"

WAVEform:EDIT:USERdefine:SAVE:UDISk <filename>

This command is used to save the presently edited custom wave data to a USB disk file, and specify the file name (if the file name is the same, it is regarded as modifying the current waveform, if the file name does not exist, it is regarded as creating a new file)

Syntax

WAVEform:EDIT:USERdefine:SAVE:UDISk <filename>

Arguments

"filename"

Example

WAVE:EDIT:USER:SAVE:UDIS "USER01.csv"

Chapter20 ISLand Subsystem

ISLand:RUN:STATe?

This command is used to query whether the instrument is in the island simulation state.

Syntax

ISLand:RUN:STATe?

Arguments

无

Returns

IDLE|AC|ISLAnd|RLC

Example

ISL:RUN:STAT?

ISLand:TIME?

This command is used to query the duration of island simulation state.

Syntax

ISLand:TIME?

Arguments

None

Returns

<NRF>

Example

ISLand:TIME?

ISLand:KEY1[:STATe] <CPD>

This command is used to set the state of K1 key.

Syntax

ISLand:KEY1[:STATe] <CPD>

Arguments

None

Query syntax

ISLand:KEY1[:STATe]?

Returns

0|OFF|1|ON

Example

ISLand:KEY1 1

ISLand:KEY1?

ISLand:KEY2:STATe] <CPD>

This command is used to set the state of K2 key.

Syntax

ISLand:KEY1[:STATe] <CPD>

Arguments

None

Query syntax

ISLand:KEY2[:STATe]?

Returns

0|OFF|1|ON

Example

ISLand:KEY2 1

ISLand:KEY2?

ISLand:RESistance:LEVel <NRf+>[,A|B|C]

This command is used to set the resistance R value of RLC load.

Syntax

ISLand:RESistance:LEVel <NRf+>[,A|B|C]

Arguments

<NRf+>

Query syntax

ISLand:RESistancel:LEVel? [A|B|C]

Returns

MINimum|MAXimum|DEFault

Example

ISLand:RESistance:LEVel 100

ISLand:RESistancel:LEVel?

ISLand:INDucatance:LEVel <NRf+>[,A|B|C]

This command is used to set the inductance value of the RLC analog load.

Syntax

ISLand:INDucatance:LEVel <NRf+>[,A|B|C]

Arguments

<NRf+>

Query syntax

ISLand:INDucatance:LEVel? [A|B|C]

Returns

MINimum|MAXimum|DEFault

Example

ISLand:INDucatance:LEVel 100

ISLand:INDucatance:LEVel?

ISLand:PHASe:LEVel <NRf+>[,A|B|C]

This command is used to set the island simulation start phase Angle.

Syntax

ISLand:PHASe:LEVel <NRf+>[,A|B|C]

Arguments

<NRf+>

Query syntax

ISLand:PHASe:LEVel? [A|B|C]

Returns

MINimum|MAXimum|DEFault

Example

ISLand:PHASe:LEVel 90 //start at 90 degree

ISLand:PHASe:LEVel?

ISLand:CAP:LEVel <NRf+>[,A|B|C]

This command is used to set the capacitance value for island simulation.

Syntax

ISLand:CAP:LEVel <NRf+>[,A|B|C]

Arguments

<NRf+>

Query syntax

ISLand:CAP:LEVel? [A|B|C]

Returns

MINimum|MAXimum|DEFault

Example

ISLand:CAP:LEVel 90

ISLand:CAP:LEVel?

ISLand:POWer:LEVel <NRf+>[,A|B|C]

This command is used to set the active power value for island simulation.

Syntax

ISLand:POWer:LEVel <NRf+>[,A|B|C]

Arguments

<NRf+>

Query syntax

ISLand:POWer:LEVel? [A|B|C]

Returns

MINimum|MAXimum|DEFault

Example

ISLand:POWer:LEVel 90

ISLand:POWer:LEVel?

ISLand:QL:LEVel <NRf+>[,A|B|C]

This command is used to set the active power value of inductance load for island simulation.

Syntax

ISLand:QL:LEVel <NRf+>[,A|B|C]

Arguments

<NRf+>

Query syntax

ISLand:QL:LEVel? [A|B|C]

Returns

MINimum|MAXimum|DEFault

Example

ISLand:QL:LEVel 90

ISLand:QL:LEVel?

ISLand:QC:LEVel <NRf+>[,A|B|C]

This command is used to set the reactive power of the capacitive load for island simulation.

Syntax

ISLand:QC:LEVel <NRf+>[,A|B|C]

Arguments

<NRf+>

Query syntax

ISLand:QC:LEVel? [A|B|C]

Returns

MINimum|MAXimum|DEFault

Example

ISLand:QC:LEVel 90

ISLand:QC:LEVel?

ISLand:MODE <CPD>[,CH1|CH2|CH3]

This command is used to set the island simulation mode.

Syntax

ISLand:MODE <CPD>[,CH1|CH2|CH3]

Arguments

<RLC|P-Q>

Query syntax

ISLand:MODE? [CH1|CH2|CH3]

Returns

<RLC|P-Q>

Example

ISLand:MODE RLC

ISLand:MODE?

ISLand:INHibit[:STATe] <CPD>

This command is used to set the Island Inhibit Control function.

Syntax

ISLand:INHibit[:STATe] <CPD>

Arguments

0|OFF|1|ON

Query syntax

ISLand:INHibit[:STATe]?

Returns

0|1

Example

ISLand:INHibit 0

ISLand:INHibit?

ISLand:VOLTage:CONTrol[:STATe] <CPD>[,CH1|CH2|CH3]

This command is used to set the function of detecting the input voltage status during island simulation.

Syntax

ISLand:VOLTage:CONTrol[:STATe] <CPD>[,CH1|CH2|CH3]

Arguments

0|OFF|1|ON

Query syntax

ISLand:VOLTage:CONTrol[:STATe]? [CH1|CH2|CH3]

Returns

0|1

Example

ISLand:VOLTage:CONTrol 0,CH1

ISLand:VOLTage:CONTrol?

ISLand:VOLTage:CONTrol:LEVel <NRf+>[,A|B|C|CH1|CH2|CH3]

This command is used to set the input voltage and set the voltage threshold when setting the island simulation.

Syntax

ISLand:VOLTage:CONTrol:LEVel <NRf+>[,A|B|C|CH1|CH2|CH3]

Arguments

<NRf+>

Query syntax

ISLand:VOLTage:CONTrol:LEVel? [A|B|C|CH1|CH2|CH3]

Returns

MINimum|MAXimum|DEFault

Example

ISLand:VOLTage:CONTrol:LEVel 200

ISLand:VOLTage:CONTrol:LEVel?

ISLand:CURRent:CONTrol[:STATe] <CPD>[,CH1|CH2|CH3]

This command is used to set the function of detecting input current status during island simulation.

Syntax

ISLand:CURRent:CONTrol[:STATe] <CPD>[,CH1|CH2|CH3]

Arguments

0|OFF|1|ON

Query syntax

ISLand:CURRent:CONTrol[:STATe]? [CH1|CH2|CH3]

Returns

0|1

Example

ISLand:CURRent:CONTrol 0,CH2

ISLand:CURRent:CONTrol?

ISLand:CURRent:CONTrol:LEVel <NRf+>[,A|B|C|CH1|CH2|CH3]

This command is usedt to set the current threshold to detect the input current during island simulation.

Syntax

ISLand:CURRent:CONTrol:LEVel <NRf+>[,A|B|C|CH1|CH2|CH3]

Arguments

<NRf+>

Query syntax

ISLand:CURRent:CONTrol:LEVel? [A|B|C|CH1|CH2|CH3]

Returns

MINimum|MAXimum|DEFault

Example

ISLand:CURRent:CONTrol:LEVel 1

ISLand:CURRent:CONTrol:LEVel?

ISLand:POWer:CONTrol[:STATe] <CPD>[,CH1|CH2|CH3]

This command is used to set the function of detecting the input power status during island simulation.

Syntax

ISLand:POWer:CONTrol[:STATe] <CPD>[,CH1|CH2|CH3]

Arguments

0|OFF|1|ON

Query syntax

ISLand:POWer:CONTrol[:STATe]? [CH1|CH2|CH3]

Returns

0|1

Example

ISLand:POWer:CONTrol 0,CH2

ISLand:CURRent:CONTrol?

ISLand:POWer:CONTrol:LEVel <NRf+>[,A|B|C|CH1|CH2|CH3]

This command is usedt to set the power threshold to detect the input power during island simulation.

Syntax

ISLand:POWer:CONTrol:LEVel <NRf+>[,A|B|C|CH1|CH2|CH3]

Arguments

<NRf+>

Query syntax

ISLand:POWer:CONTrol:LEVel? [A|B|C|CH1|CH2|CH3]

Returns

MINimum|MAXimum|DEFault

Example

ISLand:POWer:CONTrol:LEVel 1000

ISLand:POWer:CONTrol:LEVel?

Chapter21 IEEE-488 Common Commands

IEEE-488 Common commands usually control all instrument functions, such as reset, status, and synchronization. All common commands consist of a three-letter mnemonic, and are preceded by an asterisk, E.g:*RST *IDN? *SRE 8.

*CLS

This command clears the error queue and the bits of the following registers:

- Standard Event Register
- Questionable Event Register
- Status byte register

Syntax:

*CLS

Arguments:

None

*ESE

This command sets the bits in the Event Status Enable Register (ESER). The ESER is an eight-bit register that determines which bits in the Standard Event Status Register (SESR) will set the Event Summary Bit (ESB) in the Status Byte Register (SBR).

Syntax

*ESE <NR1>

Arguments

0～255

Default value

0

Example

*ESE 128

Query syntax

*ESE?

Returns

<NR1>

Related syntax

*ESR? *PSC *STB?

Bit definition of standard event enable register:

Bit	7	6	5	4	3	2	1	0
Name	PON	n.u	CME	EXE	DDE	QYE	n.u	OPC
Value Bit Weight	128		32	16	8	4		1

*ESE?

This command queries the bits in the Event Status Enable Register (ESER).

Query syntax

*ESE?

Arguments

None

Returns

<NR1> This is a decimal value with a binary weighted sum.

*ESR?

This command reads the value of Standard Event Status Register (SESR). Once this command executes, the SESR is reset. The bit definition for the SESR is the same as the Standard Event Status Enable Register.

Query syntax

*ESR?

Arguments

None

Returns

<NR1> This is a decimal value with a binary weighted sum.

Related syntax

*CLS *ESE *ESE? *OPC

*IDN?

This command reads information that identifies the power supply. It returns a parameter that contains four segments divided by a comma. Example: ITECH Ltd., IT3400, 60234567890123456, 1.01-1.02-1.03.

Query syntax

*IDN?

Arguments

None

Returns

Manufacture, model, serial number, UI ver-DSP1 ver-DSP2 ver-PFC ver-Interface ver

Example

ITECH, M7722, 00000000000004, 1.01-1.00-1.0-1.1-1.2

*OPC

This command sets the Operation Complete (OPC) bit in the Standard Event Status Register to 1 when all other commands are complete.

Syntax

*OPC

Arguments

None

Query syntax

*OPC?

Returns

<NR1>

*RST

This command resets the power supply to default settings.

Syntax

*RST

Arguments

None

*SRE <NR1>

This command sets or queries the bits in the Status Byte Enable Register. Setting this parameter can determine which byte of the Status Byte Register has a value of 1. The byte sets the RQS bit of the Status Byte Register to 1. The bit definition of the Status Byte Enable Register is as the same as the Status Byte Register.

Syntax

*SRE <NR1>

Arguments

0～255

Default value

Refer to *PSC command

Example

*SRE 128

Query syntax

*SRE?

Returns

<NR1>

Related syntax

*ESE *ESR? *PSC *STB?

*STB?

This command reads the data in the Status Byte Register (SBR).

Query syntax

*STB?

Arguments

None

Returns

<NR1>

Related syntax

*CLS *ESE *ESR

Bit definition of the Status Byte Register:

Bit	7	6	5	4	3	2	1	0
Name	OPER	RQS	ESB	MAV	QUES	EAV	n.u	n.u
Value Bit Weight	128	64	32	16	8	4		

*PSC

This instruction is used to control whether the status register is cleared when the instrument is powered-on. This instruction affects the value of the status register at the next powered-on.

Syntax

*PSC <ON|1|OFF|0>

Arguments

0|1|ON|OFF

Query syntax

*PSC?

Returns

<ON|1|OFF|0>

*SAV

This command saves the present setting values of the power supply into specified memory.

Syntax

*SAV <NR1>

Arguments

0~9

*RCL

This command recalls the setups you saved in the specified memory location.

Syntax

*RCL <NR1>

Arguments

0～9

Contact US

Thank you for purchasing ITECH products. If you have any doubt about this product, please contact us as follow.

1. Visit ITECH website www.itechate.com .
2. Select the most convenient contact for further consultation.