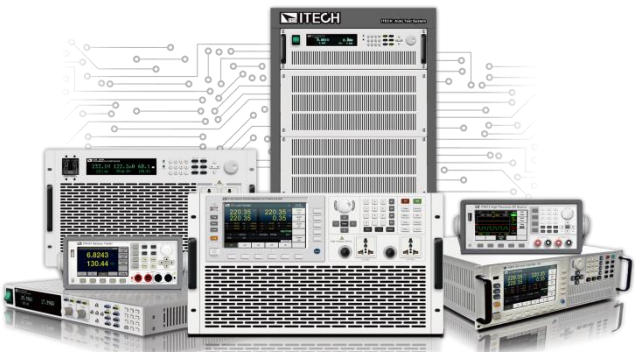
Regenerative AC/DC Electronic Load

IT8200 Series Programming Guide



Model:IT8200
Version:V1.3

Notices

Manual Part Number

IT8200

Revision

second Edition: AUG.19,

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 IT8200 electronic load 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

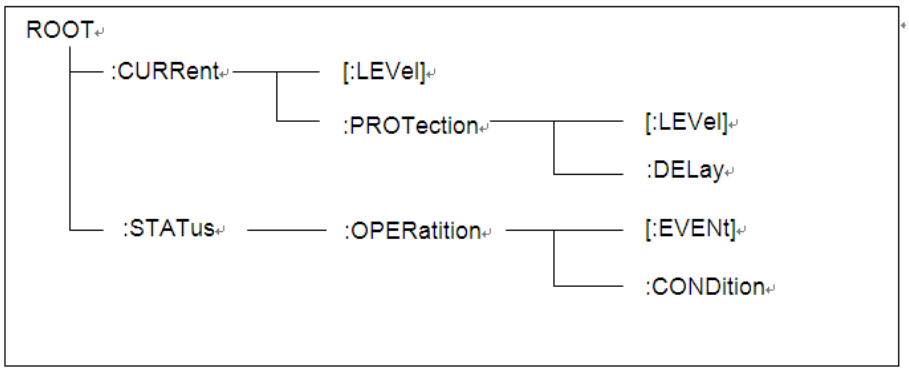
1.1 Overview

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:

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.

Case Sensitivity

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

***RST = *rst**

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:ERRor? long-form

:SYST:ERR? short form

:SYSTem:ERR? 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:Erro?** is illegal and will generate an error. The command will not be executed.

Query

Observe the following precautions with queries:

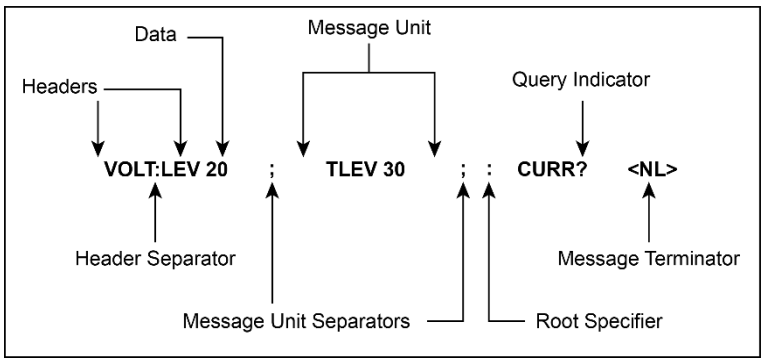
- Set up the proper number of variables for the returned data. For Example, if you are reading back a measurement array, you must dimension the array according to the number of measurements that you have placed in the measurement buffer.
- Read back all the results of a query before sending another command 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

IT8200 series electronic load 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 IT8200 load 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 IT8200 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 instrument to remote operation mode.
*IDN?                    //Query the identification of the instrument
SYSTem:FUNCtion ONE      //Set the load mode to 1-phase
SYSTem:FUNCtion THRee    //Set the load mode to 3-phase
INPut:COUPling AC        //Set the operation mode to AC
FUNCtion CC              //Set the load run mode to CC
CURR 30                  //Set the current RMS to 30A
INPut ON                 //Enable the load input
MEASure:VOLTage?         //Read the RMS value of voltage
MEASure:CURRent?         //Read the RMS value of current
MEASure:POWer?           //Read the real power value
MEASure:FREQuency?       //Read the frequency
SYSTem:ERRor?            //Query the error information of instrument
SYSTem:CLEar             //Clear the error information
INPut:PROTection:CLEar   //Clear the protection information
```

Chapter3 SCPI status register

You can get the current status of the electronic load by reading the operation status registers. The electronic load 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	Over Voltage protection
OC_Rms	1	2	RMS Over current protection
OC_Peak	2	4	PEAK over current protection
OP_POSITIVE	3	8	Over positive power protection
sync_unlock	4	16	External Synchronization unlock
UV	5	32	Under voltage
OT	6	64	Over temperature protection
UC	7	128	Under current protection
ERR_SENSE	8	256	Sense error
SHARE	9	512	Current share error
RVS	10	1024	Input reverse
INH	11	2048	Input inhibit
PS	12	4096	protect state
OSC	13	8192	Oscillating circuit
UNR	14	16384	Unknown internal fault of instrument
OC_Dc	15	32768	Over current protection of DC mode
OV_Peak	16	65536	Over peak voltage protection
FE	17	131072	Frequency error
WDOG	18	262144	Watch dog protection
Operation Status Register			
LIST_WTG	0	1	LIST watting for trigger signal
LIST_ACTIVE	1	2	LIST is running
OFF	2	4	Input is off
CC	3	8	Constant current
CV	4	16	Constant voltage
CW	5	32	Constant power
CR	6	64	Constant resistance
CS	7	128	Constant apparent power
REMOTE_LOCK	8	256	Remote lock
LOCAL_LOCK	9	512	Local lock
RMT	10	1024	Remote mode
CAL	11	2048	Calibration

Bit name	Bit	Decimal value	Definition
Questionable Status Register			
DEV_TYPE	12--13		Device type(single phase/ three phase/reverse phase) (bit13:bit12->00b/01b/10b)
OPERATION_MODE	14--15		Operation mode(source/load/current source)
SENSE	16	65536	SENSE enabled
WORK_FUNC	17--19		Work function (NORMAL/LIST/SWEEP)
CONST_MODE	20--23		Constant mode（BIT23:BIT20） AC:CC/CR/CP/CS/NCR->01b/11b/10b/100b/110b; DC:CV/CC/CP/CR/CC+CV/CR+CV/CP+CV/CC+CR/CC+CV+CP+CR-> 00b/01b/10b/11b/100b/101b/110b/111b/1000b;
WORK_MODE	24	16777216	Work mode (AC/DC)
SURGE_SAG	25	33554432	Surge and sag
EXT_ANALOG	26	67108864	Analog
SYNC_MODE	27	134217728	External Synchronization
RECTIFIED	28	268435456	Rectified mode
UPF	29	536870912	Unit power factor
LINE_TYPE	30	1073741824	Connection type(DeLta or WYE)
PHASE_LACK	31	2147483648	Phase lost

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.

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: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?

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.

Chapter5 Channel Subsystem

CHANnel[:SELect] <CPD>

This command is used to select A B C phase.

Syntax:

CHANnel[:SELect] <CPD>

Arguments

<A|B|C|>

Returns:

IDLE|AC|ISLAnd|RLC

Example

CHANnel A //Select A phase

Query syntax:

CHANnel[:SELect]?

Chapter6 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

Chapter7 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

Chapter8 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|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

Chapter9 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

Chapter10 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?

Chapter11 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?

Chapter12 FETCh & MEASure Subsystem

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

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

This command is used to read the RMS value of 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]

MEASure[:SCALar]:CURRent[:AC]? [A|B|C]

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:CURR? A

MEAS:CURR?

FETCh[:SCALar]:CURRent:DC?

MEASure[:SCALar]:CURRent:DC?

This command is used to read the DC component of 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]

MEASure[:SCALar]:CURRent:DC? [A|B|C]

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:CURR:DC?

MEAS:CURR:DC?

FETCh[:SCALar]:CURRent[:AC][:AMPLitude]:MAXimum:POSitive?

MEASure[:SCALar]:CURRent[:AC][:AMPLitude]:MAXimum:POSitive?

This command is used to read the positive peak value of 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[:AC][:AMPLitude]:MAXimum:POSitive?

MEASure[:SCALar]:CURRent[:AC][:AMPLitude]:MAXimum:POSitive?

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:CURR:MAX:POS?

MEAS:CURR:MAX:POS? A

FETCh[:SCALar]:CURRent[:AC][:AMPLitude]:MAXimum:NEGative?

MEASure[:SCALar]:CURRent[:AC][:AMPLitude]:MAXimum:NEGative?

This command is used to read the negative peak value of 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[:AC][:AMPLitude]:MAXimum:NEGative? [A|B|C]

MEASure[:SCALar]: CURRent[:AC][:AMPLitude]:MAXimum:NEGative? [A|B|C]

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:CURR:MAX:NEG?

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

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

This command is used to read the peak value of 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[:AC][:AMPLitude]:MAXimum?

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

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:CURR:MAX?

FETCh[:SCALar]:VOLTage[:AC][:AMPLitude]:MAXimum:HOLD?

MEASure[:SCALar]:VOLTage[:AC][:AMPLitude]:MAXimum:HOLD?

This command is used to read the peak voltage hold value of ABC. It will be cleared automatically on the next power-on.

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[:AC][:AMPLitude]:MAXimum:HOLD?

MEASure[:SCALar]:VOLTage[:AC][:AMPLitude]:MAXimum:HOLD?

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:CURR:CFAC?

FETCh[:SCALar]:VOLTage[:AC][:AMPLitude]:MAXimum:HOLD:CLEar

MEASure[:SCALar]:VOLTage[:AC][:AMPLitude]:MAXimum:HOLD:CLEar

This command is used to clear the single-phase voltage peak hold 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]:VOLTage[:AC][:AMPLitude]:MAXimum:HOLD:CLEar

MEASure[:SCALar]:VOLTage[:AC][:AMPLitude]:MAXimum:HOLD:CLEar

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETCh:VOLTage:MAXimum:HOLD:CLEar

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]

MEASure[:SCALar]:CURRent:CFACtor? [A|B|C]

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 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]

MEASure[:SCALar]:FREQuency? [A|B|C]

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:FREQ?

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

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

This command is used to read the 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]

MEASure[:SCALar]:POWer[:REAL]? [A|B|C]

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:POWer?

FETCh[:SCALar]:POWer:APParent?

MEASure[:SCALar]:POWer:APParent?

This command is used to read the 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]

MEASure[:SCALar]:POWer:APParent? [A|B|C]

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:POW:APP?

FETCh[:SCALar]:POWer:REACtive?

MEASure[:SCALar]:POWer:REACtive?

This command is used to read the 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]

MEASure[:SCALar]:POWer:REACtive? [A|B|C]

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:POW:REAC?

FETCh[:SCALar]:POWer:PFACtor?

MEASure[:SCALar]:POWer:PFACtor?

This command is used to read the 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]

MEASure[:SCALar]:POWer:PFACtor? [A|B|C]

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:POW:PFAC?

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

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

This command is used to read the effective value of 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]

MEASure[:SCALar]:VOLTage[:AC]? [A|B|C]

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:VOLT?

FETCh[:SCALar]:VOLTage:DC?

MEASure[:SCALar]:VOLTage:DC?

This command is used to read the DC component of 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]

MEASure[:SCALar]:VOLTage:DC? [A|B|C]

Arguments

[A|B|C]

Returns:

<NRf> …

Example

FETC:VOLT:DC?

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

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

This command is used to read the peak value of 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[:AC][:AMPLitude]:MAXimum?

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

Arguments

[A|B|C]

Returns:

<NRf>…

Example

FETC:VOLT:AMP:MAX?

FETCh[:SCALar]:VOLTage[:AC][:AMPLitude]:MAXimum:POSitive?

MEASure[:SCALar]:VOLTage[:AC][:AMPLitude]:MAXimum:POSitive?

This command is used to read the positive peak value of 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[:AC][:AMPLitude]:MAXimum:POSitive?

MEASure[:SCALar]:VOLTage[:AC][:AMPLitude]:MAXimum:POSitive?

Arguments

[A|B|C]

Returns:

<NRf> …

Example

FETC:VOLT:MAX:POS?

FETCh[:SCALar]:VOLTage[:AC][:AMPLitude]:MAXimum:NEGative?

MEASure[:SCALar]:VOLTage[:AC][:AMPLitude]:MAXimum:NEGative?

This command is used to read the negative peak value of 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[:AC][:AMPLitude]:MAXimum:NEGative?

MEASure[:SCALar]:VOLTage[:AC][:AMPLitude]:MAXimum:NEGative?

Arguments

[A|B|C]

Returns:

<NRf> …

Example

FETC:VOLT:MAX:NEG?

FETCh[:SCALar]?

MEASure[:SCALar]?

This command is used to get all METER data.

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]

MEASure[:SCALar]? [A|B|C]

Arguments

[A|B|C]

Returns:

<NRf> …

Example

FETC? A

FETCh? If set the parameter [A|B|C], the returned 19 floating point measurements are <float>,...,<float> , in order are

(Uac,Iac,P,Upp,Upn,Ipp,Ipn,Freq,CF,PF,S,Q,Udc,Idc,Uthd,Ithd,Upeak,Urms,Irms)

MEASure? If set the parameter [A|B|C], the returned 17 floating point measurements are <float>,...,<float>, in order are

(Uac,Iac,P,Upp,Upn,Ipp,Ipn,Freq,CF,PF,S,Q,Udc,Idc,Uthd,Ithd,Upeak)

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]:TOTal?

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

Arguments

无

Returns:

<NRf>

Example

FETC:POW:TOT?

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:TOTal?

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

Arguments

无

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:TOTal?

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

Arguments

无

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>

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

Arguments

<BA|CA|CB>

Returns:

<NRf>

Example

FETC:LTLV? CB

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

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

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

Syntax:

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

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

Arguments

<[A|B|C]>,<NR1>

NR1 range: 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>

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

Arguments

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

NR1 range: 0-50

Returns:

<NRf>

Example

FETC:CURR:HARM? A,5

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

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

This command is used to measure voltage harmonic components.

Syntax:

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

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

Arguments

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

NR1 range: 0-50

Returns:

<NRf>

Example

FETC:VOLT:HARM:DIST? A,8

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

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

This command is used to measure current harmonic components.

Syntax:

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

Arguments

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

NR1 range: 0-50

Returns:

<NRf>

Example

FETC:CURR:HARM:DIST? CH3,10

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

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

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

Syntax:

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

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

Arguments

<[A|B|C]>,<NR1>

NR1 range: 0-50

Returns:

<NRf>

Example

FETC:VOLT:HARM:PHAS? A,5

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

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

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

Syntax:

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

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

Arguments

<[A|B|C]>,<NR1>

NR1 range: 0-50

Returns:

<NRf>

Example

FETC:CURR:HARM:PHAS? A,10

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

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

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

Syntax:

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

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

Arguments

<[A|B|C]>

Returns:

<NRf>

Example

FETC:VOLT:HARM:THD? C

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

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

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

Syntax:

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

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

Arguments

<[A|B|C]>

Returns:

<NRf>

Example

FETC:CURR:HARM:THD? A

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

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

Arguments

<[A|B|C]>,<NR1>

NR1 range: 0-50

Returns:

<NRf>

Example

FETC:ARR:VOLT:HARM? B,11

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

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

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

Syntax:

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

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

Arguments

<[A|B|C]>,<NR1>

NR1 range: 0-50

Returns:

<NRf>

Example

FETC:ARR:CURR:HARM? A,6

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

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

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

Syntax:

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

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

Arguments

<[A|B|C]>,<NR1>

NR1: 0-50

Returns:

<NRf>

Example

FETC:ARR:VOLT:HARM:PHAS? CH3,6

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

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

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

Syntax:

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

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

Arguments

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

NR1: 0-50

Returns:

<NRf>

Example

FETC:ARR:CURR:HARM:PHAS? A,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>

MEASure[: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? A,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>

MEASure[: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? A,6

VETCor:OEDer <NR1>

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

Syntax:

VETCor:OEDer

Arguments

<NR1> range: 0-50

Returns:

<NRf>

Query syntax:

VETC:OED?

Example

VETC:OED 19

VETCor:DATA?

This command is used to query the vectior data of present order.

Syntax:

VETCor:DATA?

Arguments

无

Returns:

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

Example

VETCor:DATA?

VETCor:TYPE <CPD>

This command is used to set the vector type.

Syntax:

VETCor:TYPE <CPD>

Arguments

<U|I|ALL>

Returns:

U|I|ALL

Query syntax:

VETCor:TYPE??

Example

VETC:TYPE I

Chapter13 SENSE Subsystem

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

This command enables or disables the sense function.

Syntax:

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

Arguments

<OFF|ON|0|1>

Default

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:EXTernal:SYNC[:STATe] <boolean>

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

Syntax:

SENSe:EXTernal:SYNC[:STATe] <boolean>

Arguments

0|OFF|1|ON

Query syntax::

SENSe:EXTernal:SYNC[:STATe]?

Returns::

0|1

Example

SENS:EXT:SYNC 1

SENSe:EXTernal:SYNC:PHASe <NRf>

Set the angle difference between the instrument and the external signal frequency. When the external frequency lock function is turned on, the output phase of the instrument maintains a fixed angle difference with the external frequency.

Syntax:

SENSe:EXTernal:SYNC:PHASe <NRf>

Arguments

MAXimum|MINimum

Query syntax::

SENSe:EXTernal:SYNC:PHASe?

Returns::

<NRf>

Example

SENS:EXT:SYNC:PHAS 90

SENSe:EXTernal:SYNC:DIFFerence <CPD>,<NRf>

This command sets the instrument B to A and C to A angle difference. Editable under 3 phase only.

Syntax:

SENSe:EXTernal:SYNC:DIFFerence <CPD>,<NRf>

Arguments

CPD:BA|CA

NRf:MAXimum|MINimum

Query syntax::

SENSe:EXTernal:SYNC:DIFFerence? <CPD>

Returns::

<NRf>

Example

SENSe:EXTernal:SYNC:DIFFerence <CPD>,<NRf>

Chapter14 Load Protect Subsystem

[SOURce:]PROTection:AUTO:CLEar[:STATe] <Boolean>

This command is used to set the state of the protection auto-clear function.

Syntax:

PROTection:AUTO:CLEar[:STATe] <Boolean>

Arguments

0|1|OFF|ON

Query syntax:

PROTection:AUTO:CLEar[:STATe]?

Example:

PROT:AUTO:CLE 1

[SOURce:]CURRent:PROTection:STATe <Boolean>

This command is used to set state of over current RMS protection.

Syntax:

CURRent:PROTection:STATe <Boolean>

Arguments

0|1|OFF|ON

Query syntax:

CURRent:PROTection:STATe?

Returns:

0|1|OFF|ON

Example

CURR:PROT:STAT 1

[SOURce:]CURRent:PROTection[:LEVel] <NRf+>

The command sets the protection value of current RMS protection, unit: A.

Syntax:

CURRent:PROTection[:LEVel] <NRf+>

Arguments

0|1|OFF|ON

Query syntax:

CURRent:PROTection[:LEVel]? [MAXimum|MINimum]

Returns:

0|1|OFF|ON

Example

CURRent:PROTection 90

[SOURce:]CURRent:PROTection:DELay <NRf+>

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

Syntax:

CURRent:PROTection:DELay <NRf+>

Arguments

<MINimum-MAXimum>

Query syntax:

[SOURce:]CURRent:PROTection:DELay?

Returns:

<MINimum-MAXimum>

Example

```
CURR:PROT:DEL 0.5
CURR:PROT:DEL 10        //设置通道 1 的 RMS 保护延迟为 10s
```

[SOURce:]CURRent:PEAK:PROTection:STATe <Boolean>

This command is used to set state of over current peak protection.

Syntax:

CURRent:PEAK:PROTection:STATe <Boolean>

Arguments

0|1|OFF|ON

Query syntax:

CURRent:PEAK:PROTection:STATe?

Returns:

0|1|OFF|ON

Example

CURRent:PEAK:PROTection:STATe 1 //设置峰值保护状态为开启状态.

[SOURce:]CURRent:PEAK:PROTection[:LEVel] <NRf+>

The command sets the protection value of current peak protection, unit: A.

Syntax:

CURRent:PEAK:PROTection[:LEVel] <NRf+>

Arguments

<MINimum-MAXimum>

Query syntax:

CURRent:PEAK:PROTection[:LEVel]? [MAXimum|MINimum]

Returns:

MAXimum|MINimum|DEFault

Example

CURRent:PEAK:PROT 30

[SOURce:]CURRent:PEAK:PROTection:DELay <NRf+>

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

Syntax:

CURRent:PEAK:PROTection:DELay <NRf+>

Arguments

MAXimum|MINimum|DEFault

Query syntax:

CURRent:PEAK:PROTection:DELay?

Returns:

[MAXimum|MINimum]

Example

CURRent:PEAK:PROTection:DELay 1

[SOURce:]POWer:PROTection:STATe <Boolean>

Sets the state of over power protection.

Syntax:

POWer:PROTection:STATe <Boolean>

Arguments

0|1|OFF|ON

Query syntax:

POWer:PROTection:STATe?

Returns:

0|1|OFF|ON

Example

POWer:PROTection:STATe 1

[SOURce:]POWer:PROTection[:LEVel] <NRf+>

The command sets the protection value of power protection, unit: w.

Syntax:

POWer:PROTection[:LEVel] <NRf+>

Arguments

<MINimum-MAXimum>

Query syntax:

POWer:PROTection[:LEVel]? [MAXimum|MINimum]

Returns:

MAXimum|MINimum|DEFault

Example

POWer:PROTection 30

[SOURce:]POWer:PROTection:DELay <NRf+>

Set the power protection delay time, the unit is: s

Syntax:

POWer:PROTection:DELay <NRf+>

Arguments

<MINimum-MAXimum>

Query syntax:

POWer:PROTection:DELay? [MAXimum|MINimum]

Returns:

<MINimum-MAXimum>

Example

POWer:PROTection:DELay 1

[SOURce:]VOLTage:UNDer:PROTection:STATe <Boolean>

Sets the state of the undervoltage protection. Setting is only required when in DC mode.

Syntax:

POWer:PROTection:STATe <Boolean>

Arguments

0|1|OFF|ON

Query syntax:

VOLTage:UNDer:PROTection:STATe?

Returns:

0|1|OFF|ON

Example

VOLTage:UNDer:PROTection:STATe 1

[SOURce:]VOLTage:UNDer:PROTection[:LEVel] <NRf+>

Sets the undervoltage protection value in the DC mode of the machine. The unit is V. It needs to be set only when in DC mode.

Syntax:

VOLTage:UNDer:PROTection[:LEVel] <NRf+>

Arguments

<MINimum-MAXimum>

Query syntax:

VOLTage:UNDer:PROTection[:LEVel]? [MAXimum|MINimum]

Returns:

MAXimum|MINimum|DEFault

Example

VOLTage:UNDer:PROTection 30

[SOURce:]VOLTage:UNDer:PROTection:DELay <NRf+>

This command is used to set the undervoltage protection delay time in s. It only needs to be set when in DC mode.

Syntax:

VOLTage:UNDer:PROTection:DELay <NRf+>

Arguments

<MINimum-MAXimum>

Query syntax:

VOLTage:UNDer:PROTection:DELay? [MAXimum|MINimum]

Returns:

<MINimum-MAXimum>

Example

VOLTage:UNDer:PROTection:DELay 1

[SOURce:]VOLTage:PEAK:PROTection[:LEVel] <NRf+>

Sets the upper voltage peak limit.

Syntax:

VOLTage:PEAK:PROTection[:LEVel] <NRf+>

Arguments

<MINimum-MAXimum>

Query syntax:

VOLTage:PEAK:PROTection[:LEVel]? [MAXimum|MINimum]

Returns:

<MINimum-MAXimum>

Example

VOLTage:PEAK:PROTection 350

Chapter15 SOURce Subsystem

[SOURce:]SYSTem:FUNCtion <CPD>

This command is used to set the mode of present load, including single-phase/three-phase/reverse.

Syntax:

SYSTem:FUNCtion <CPD>

Arguments

ONE|THRee|DIFFerence

Default

ONE

Query syntax:

SYSTem:FUNCtion?

Returns:

ONE|THRee|DIFFerence

Example

SYST:FUNC ONE

[SOURce:]FUNCtion <CPD1>

This command is used to set load mode.

Syntax:

FUNCtion <CPD1>

Arguments

AC mode: CC|CR|CP|CS|CC+CR|CE

DC mode: CC|CR|CP|CV|CC+CV|CR+CV|CP+CV|CR+CC|CC+CR+CP+CV

Query syntax:

FUNCtion?

Returns:

AC mode: CC|CR|CP|CS|CC+CR|CE

DC mode: CC|CR|CP|CV|CC+CV|CR+CV|CP+CV|CR+CC|CC+CR+CP+CV

Example

FUNC CC

[SOURce:]FUNCtion:CATalog?

This command is used to query the supports load mode.

Syntax:

[SOURce:]FUNCtion:CATalog?

Arguments

AC: CC|CR|CP|CS|CC+CR|CE

DC: CC|CR|CP|CV|CC+CV|CR+CV|CP+CV|CR+CC|CC+CR+CP+CV

Arguments

AC: CC|CR|CP|CS|CC+CR|CE

DC: CC|CR|CP|CV|CC+CV|CR+CV|CP+CV|CR+CC|CC+CR+CP+CV

Example

FUNC:CAT? //query the load mode catalog

[SOURce:]UPFactor[:STATe] <Boolean>

This command is used to set state of UPF (Unit Power Factor)

Syntax:

UPFactor[:STATe] <Boolean>

Arguments

<0|1|OFF|ON>

Query syntax:

UPFactor[:STATe]?

Returns:

<0|1|OFF|ON>

Example

UPFactor 1

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

Set and query the upper limit of RMS current.

Syntax:

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

Arguments

<NRf+>

MAXimum|MINimum

Query syntax:

CURRent[:LEVel][:IMMediate][:AMPLitude][:AC]? [MAXimum| MINimum]

Returns:

MAXimum|MINimum

Example

CURR 30

CURR 30,20,10

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

DC current limit value setting command in Normal mode. This command is valid only in DC and DC+AC modes.

Syntax:

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

Query syntax:

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

Returns:

<NRf+>

Example

CURRent:DC 30

[SOURce:]CURRent:SLEW[:AC] <NRf+>[,NRf+][,NRf+]

Sets the slope of load current. Unit A/ms.

Syntax:

CURRent:SLEW[:AC] <NRf+>[,NRf+][,NRf+]

Arguments

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

Query syntax:

CURRent:SLEW[:AC]? [MAXimum| MINimum]

Example

CURR:SLEW 20.0 //set the current slope to 20.0A/ms

[SOURce:]CURRent:SLEW:DC <NRf+>

Sets the slope of the current in DC mode. Unit A/ms.

Syntax:

CURRent:SLEW:DC <NRf+>

Arguments

<NRf+>

Query syntax:

CURRent:SLEW:DC? [MAXimum| MINimum]

Example

CURR:SLEW:DC 20.0

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

Sets the resistance value. Unit W.

Syntax:

RESistance[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,NRf+][,NRf+]

Query syntax:

RESistance[:LEVel][:IMMediate][:AMPLitude]? [MAXimum| MINimum]

Arguments:

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

Example:

RESistance 500

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

Sets the power value. Unit W.

Syntax:

POWer[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,NRf+][,NRf+]

Query syntax:

POWer[:LEVel][:IMMediate][:AMPLitude]? [MAXimum| MINimum]

Arguments

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

Example：

POWer 500

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

This command is used to set or query the apparent power under CS mode. Unit kVA.

Syntax:

KVA[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,NRf+][,NRf+]

Query syntax:

KVA[:LEVel][:IMMediate][:AMPLitude]? [MAXimum| MINimum]

Arguments

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

Example：

KVA 10

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

This command is used to set or query the phase shift between voltage and current.

Syntax:

PSHift[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,NRf+][,NRf+]

Arguments

-90° ~ +90°

Query syntax:

PSHift[:LEVel][:IMMediate][:AMPLitude]? [MAXimum| MINimum]

Arguments

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

Example：

PSHift 90

PSHift?

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

This command is used to set or query CF.

Syntax:

CFACtor[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,NRf+][,NRf+]

Arguments

1.414 ~ 5

Query syntax:

CFACtor[:LEVel][:IMMediate][:AMPLitude]? [MAXimum| MINimum]

Arguments

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

Example：

CFACtor 5

CFACtor?

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

This command is used to set or query voltage value of DC mode.

Syntax:

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

Arguments

<NRf+>

Query syntax:

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

Arguments

MAXimum|MINimum|DEFault

Example：

VOLTage 20

[SOURce:]CE:PEAK:CURRent <NRf+>[,NRf+][,NRf+]

This command is used to set or query Ipeak value of CE mode.

Syntax:

CE:PEAK:CURRent <NRf+>[,NRf+][,NRf+]

Arguments

MAXimum|MINimum|DEFault

Query syntax:

CE:PEAK:CURRent? [MAXimum|MINimum]

Arguments

MAXimum|MINimum|DEFault

Example：

CE:PEAK:CURRent 30

[SOURce:]CE:TYPE <CPD>

This command is used to select or query CE type.

Syntax:

[SOURce:]CE:TYPE < A|B>

Arguments

A|B

A: Parallel RLC

B: Rectifier single phase RLC

Query syntax:

[SOURce:]CE:TYPE?

Example：

CE:TYPE A

[SOURce:]CE:TA:R <NRf+>[,NRf+][,NRf+]

This command is used to set or query the R set under Parallel RLC, Unit: Ω

Syntax:

[SOURce:]CE:TA:R <NRf+>[,NRf+][,NRf+]

Arguments

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

Query syntax:

[SOURce:]CE:TA:R? [MAXimum| MINimum]

Arguments

<NRf+>

Example：

CE:TA:R 100

[SOURce:]CE:TA:RL <NRf+>[,NRf+][,NRf+]

This command is used to set or query the RL under Parallel RLC. Unit: Ω

Syntax:

[SOURce:]CE:TA:RL <NRf+>[,NRf+][,NRf+]

Query syntax:

[SOURce:]CE:TA:RL? [MAXimum| MINimum]

Example

CE:TA:RL 10

[SOURce:]CE:TA:L <NRf+>[,NRf+][,NRf+]

This command is used to set or query the L value under Parallel RLC, Unit: mH

Syntax:

[SOURce:]CE:TA:L <NRf+>[,NRf+][,NRf+]

Query syntax:

[SOURce:]CE:TA:L? [MAXimum| MINimum]

Example

CE:TA:L 10

[SOURce:]CE:TA:RC <NRf+>[,NRf+][,NRf+]

This command is used to set or query the RC value under Parallel RLC, Unit: Ω

Syntax:

[SOURce:]CE:TA:RC <NRf+>[,NRf+][,NRf+]

Arguments

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

Query syntax:

[SOURce:]CE:TA:RC? [MAXimum| MINimum]

Example

CE:TA:RC 10

[SOURce:]CE:TA:C <NRf+>[,NRf+][,NRf+]

This command is used to set or query the C value under Parallel RLC, Unit: uF

Syntax:

[SOURce:]CE:TA:C <NRf+>[,NRf+][,NRf+]

Arguments

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

Query syntax:

[SOURce:]CE:TA:C? [MAXimum| MINimum]

Example

CE:TA:C 10

[SOURce:]CE:TA:L:AINitial <NRf+>[,NRf+][,NRf+]

This command is used to set or query the initial current value of inductance, Unit: A.

Syntax:

[SOURce:]CE:TA:L:AINitial <NRf+>[,NRf+][,NRf+]

Arguments

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

Query syntax:

[SOURce:]CE:TA:L:AINitial? [MAXimum| MINimum]

Example

CE:TA:L:AINitial 10

[SOURce:]CE:TB:R <NRf+>[,NRf+][,NRf+]

This command is used to set or query the R set under Rectifier single phase RLC mode, Unit: Ω

Syntax:

[SOURce:]CE:TB:R <NRf+>[,NRf+][,NRf+]

Arguments

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

Query syntax:

[SOURce:]CE:TB:R? [MAXimum| MINimum]

Arguments

<NRf+>

Example：

CE:TB:R 100

[SOURce:]CE:TB:RS <NRf+>[,NRf+][,NRf+]

This command is used to set or query the RS set under Rectifier single phase RLC mode, Unit: Ω

Syntax:

[SOURce:]CE:TB:RS <NRf+>[,NRf+][,NRf+]

Query syntax:

[SOURce:]CE:TB:RS? [MAXimum| MINimum]

Example

CE:TB:RS 10

[SOURce:]CE:TB:L <NRf+>[,NRf+][,NRf+]

This command is used to set or query the L value under Rectifier single phase RLC mode, Unit: mH

Syntax:

[SOURce:]CE:TB:L <NRf+>[,NRf+][,NRf+]

Query syntax:

[SOURce:]CE:TB:L? [MAXimum| MINimum]

Example

CE:TB:L 10

[SOURce:]CE:TB:C <NRf+>[,NRf+][,NRf+]

This command is used to set or query the C value under Rectifier single phase RLC mode, Unit: uF

Syntax:

[SOURce:]CE:TB:C <NRf+>[,NRf+][,NRf+]

Arguments

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

Query syntax:

[SOURce:]CE:TB:C? [MAXimum| MINimum]

Example

CE:TB:C 10

[SOURce:]CE:TB:C:VINitial <NRf+>[,NRf+][,NRf+]

This command is used to set or query the initial voltage value of capacitance, Unit: V.

Syntax:

[SOURce:]CE:TB:C:VINitial <NRf+>[,NRf+][,NRf+]

Arguments

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

Query syntax:

[SOURce:]CE:TB:C:VINitial? [MAXimum| MINimum]

Example

CE:TB:C:VINitial 10

[SOURce:]CE:TB:D:VOLTage <NRf+>[,NRf+][,NRf+]

This command is used to set or query diode voltage under Rectifier single phase RLC mode,Unit:V

Syntax:

[SOURce:]CE:TB:D:VOLTage <NRf+>[,NRf+][,NRf+]

Arguments

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

Query syntax:

[SOURce:]CE:TB:D:VOLTage? [MAXimum| MINimum]

Example

CE:TB:D:VOLTage 10

Chapter16 Input Subsystem

INPut:COUPling <CPD>

This command is used to switch AC and DC mode.

Syntax:

INPut|OUTPut:COUPling <CPD>

Arguments

DC|AC

Query syntax:

INPut:COUPling?

Returns:

DC|AC

Example

INPut:COUPling AC

INPut:PHASe:LOSS <STATE>

This command is used to enable the phase loss state.

Syntax:

INPut:PHASe:LOSS <STATE>

Arguments

0|OFF|1|ON

Query syntax:

INPut:PHASe:LOSS?

Example

INPut:PHASe:LOSS 1

INPut:LINE:CONNection <CPD>

This command is used to set the connection status of the output. WYE is the star connection method and DELTa is the triangle connection method.

Syntax:

INPut:LINE:CONNection <CPD>

Arguments

<WYE|DELTa>

Query syntax:

INPut:LINE:CONNection?

Returns:

<WYE|DELTa>

Example

INP:LINE:CONN WYE

INPut <state>

This command is used to set the input state.

Syntax:

INPut <state>

Arguments

0|OFF|1|ON

Query syntax:

INPut[:STATe]?

Returns:

0|OFF|1|ON

Example

INP 1

INPut:RECTified[:STATe] <boolean>

This command is used to set the state of rectified function.

Syntax:

INPut:RECTified[:STATe] <boolean>

Arguments

0|OFF|1|ON

Query syntax:

INPut:RECTified[:STATe]?

Returns:

0|OFF|1|ON

Example

INPut:RECTified 1

INPut:INTegrity <CPD>

This command is used to set the integrity of the input waveform.

Syntax:

INPut:INTegrity <CPD>

Arguments

<FULL|POSitive|NEGative>

Query syntax:

INPut:INTegrity?

Returns:

<FULL|POSitive|NEGative>

Example

INPut:INTegrity FULL

INPut:PROTection:CLEar

This command is used to clear the protection status.

Syntax:

INPut:PROTection:CLEar

Example

INPut:PROTection:CLEar

INPut:PROTection:WDOG[:STATe] <state>

This command is used to enable or disable the watch dog function.

Syntax:

INPut:PROTection:WDOG[:STATe] <state>

Arguments

0|OFF|1|ON

Query syntax:

INPut|OUTPut:PROTection:WDOG[:STATe]?

Example

INPut:PROTection:WDOG 1

INPut:PROTection:WDOG:DELay <time>

This command is used to set software watchdog delay time, unit is s.

Syntax:

INPut:PROTection:WDOG:DELay <time>

Arguments

MINimum|MAXimum

Query syntax:

INPut:PROTection:WDOG:DELay?

Returns:

MINimum|MAXimum

Example

INPut:PROTection:WDOG:DELay 1

INPut:REGulation:SPEed <CPD>

This command is used to set the input regulation speed.

Syntax:

INPut:REGulation:SPEed <CPD>

Arguments

<SLOW|FAST>

Query syntax:

INPut:REGulation:SPEed?

Returns:

<SLOW|FAST>

Example

INPut:REGulation:SPEed SLOW

INPut:OFF:MODE <CPD>

This command is set the connection status when the load 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.

Syntax:

INPut:OFF:MODE <CPD>

Arguments

OPENz|HIGHz

Query syntax:

INPut|OUTPut:OFF:MODE?

Example

INPut:OFF:MODE <CPD>

INPut:ON:PHASe:MODE <CPD>

This command is used to set the mode of phase control when the load input is turned on.

PHASe: The load starts input with the set phase angle

IMMediate: the load starts input with 0 phase angle.

Syntax:

OUTPut:ON:PHASe:MODE <CPD>

Arguments

PHASe|IMMediate

Query syntax:

INPut:ON:PHASe:MODE?

Returns:

PHASe|IMMediate

Example

OUTP:ON:PHAS:MODE IMM

INPut:ON:PHASe:LEVel <NRf+>

The command sets the starting phase angle value when the instrument input is turned on.

Syntax:

INPut:ON:PHASe:LEVel <NRf+>

Arguments

MINimum|MAXimum

Query syntax:

INPut:ON:PHASe:LEVel?

Returns:

MINimum|MAXimum

Example

INPut:ON:PHASe:LEVel 60

INPut:OFF:PHASe:MODE <CPD>

The command sets the mode of phase control when the instrument input is turned off.

PHASe: The load stop input with the set phase angle

IMMediate: the load stop input with 0 phase angle.

Syntax:

INPut:OFF:PHASe:MODE <CPD>

Arguments

PHASe|IMMediate

Query syntax:

INPut:OFF:PHASe:MODE?

Returns:

PHASe|IMMediate

Example

INPut:OFF:PHASe:MODE IMM

INPut:OFF:PHASe:LEVel <NRf+>

This command is used to set the phase angle value when the instrument input is turned off.

Syntax:

INPut:OFF:PHASe:LEVel <NRf+>

Arguments

MINimum|MAXimum|DEFault

Query syntax:

INPut:OFF:PHASe:LEVel?

Returns:

MINimum|MAXimum

Example

INPut:OFF:PHASe:LEVel 90

INPut:BALance[:STATe] <Boolean>

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

This command is only supported in three-phase mode.

Syntax:

INPut:BALance[:STATe] <Boolean>

Arguments

0|OFF|1|ON

Query syntax:

INPut:BALance[:STATe]?

Returns:

0|1

Example

INPut:BALance ON

INPut:MONitor:PHASe <CPD>

This command is used to set the monitored phase of the input.

This command is only supported in three-phase mode.

Syntax:

INPut:MONitor:PHASe <CPD>

Arguments

A|B|C

Returns:

0|1

Example

INPut:MONitor:PHASe A

INPut:MONitor:VOLTage:RATio <CPD>

This command is used to set the input voltage monitoring ratio.

Syntax:

INPut:MONitor:VOLTage:RATio <CPD>

Arguments

50|100

Query syntax:

INPut:MONitor:VOLTage:RATio?

Example

INPut:MONitor:VOLTage:RATio 100

INPut:MONitor:CURRent:RATio <CPD>

This command is used to set the current monitoring ratio of the input.

Syntax:

INPut:MONitor:CURRent:RATio <CPD>

Arguments

5.0|10.0

Query syntax:

INPut:MONitor:CURRent:RATio?

Example

INPut:MONitor:CURRent:RATio 5.0

Chapter17 ARB Subsystem

LIST:STATe?

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

Syntax:

LIST:STATe?

Example

LIST:STAT?

LIST:REPeat <NR1>

This command is used to set and query the number of cycles run by list.

Syntax:

LIST:REPeat <NR1>

Arguments

<NR1>

MINimum|MAXimum|DEFault

*RST

0

Query syntax:

LIST:REPeat?

Returns:

<NR1>

Example

LIST:REP 10

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:

LIST:TERMinate <CPD>

Arguments

<CPD>

NORMal|LAST|OFF

Query syntax:

LIST:TERMinate?

Returns:

NORMal|LAST|OFF

*RST

OFF

Example

LIST:TERM LAST

LIST:RSTate?

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:

LIST:RUNTime:STATe?

Returns:

<NR1>,<NR1>

Example

LIST:RUNTime:STATe?

LIST:RECall <string>

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

Syntax:

LIST:RECall <string>

Query syntax:

LIST:RECall?

Example

LIST:RCL "test.csv"

LIST:STEP:COUNt?

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

Syntax:

LIST:STEP:COUNt?

Returns:

<NR1>

Example

LIST:STEP:COUNt?

LIST:CLEar

This command is used to clear the list configuration.

Syntax:

LIST:CLEar

Example

LIST:CLE

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:

IAC_A	0
IAC_B	1
IAC_C	2
IAC_SLEW_A	3
IAC_SLEW_B	4
IAC_SLEW_C	5
IDC_A	6
IDC_B	7
IDC_C	8
IDC_SLEW_A	9
IDC_SLEW_B	10

IDC_SLEW_C 11
PHASE_MODE 12
START_A 13
START_B 14
START_C 15
R_A 16
R_B 17
R_C 18
R_A_SLEW 19
R_B_SLEW 20
R_C_SLEW 21
THREE_WAVE_A 22
THREE_WAVE_B 23
THREE_WAVE_C 24
THREE_TIME_MODE 25
THREE_TIME 26
THREE_TRIG_OUT 27
A_PHASE_SHIFT 29
B_PHASE_SHIFT 29
C_PHASE_SHIFT 30
A_wave_PARAM 31
B_WAVE_PARAM 32
C_WAVE_PARAM 33

Single mode/ Reverse mode:

IAC 0
IAC_SLEW 1
IDC 2
IDC_SLEW 3
PHASE_MODE 4
START_A 5
R 6
R_SLEW 7
WAVE 9
TIME_MODE 9
TIME 10
TRIG_OUT 11
PHASE_START 12
WAVE_PARM 13

Syntax:

LIST:STEP <NR1>,<string>

Query syntax:

LIST:STEP? <NR1>

Example

Three mode:

LIST:STEP 1,"11,11,11,11,11,11,5,5,5,6,6,6,PHASe,90,90,90,16,16,16,2,2,2,Sine,Sine,Sine,TIME,2,OFF,10,10,10,2,2,2"

Single mode or reverse mode:

LIST:STEP 1,"1,1,10,5,PHAse,90，0,0,Sine,TIME,1,OFF,10,2"

Note: The values set in this command include all parameters of CC and CR modes, but when returned, only the value displayed in the interface is returned.

LIST:STEP:ITEM <NR1>,<NR1>,<NRf+>

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

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

<NR1>:

IAC_A 0

IAC_B 1

IAC_C 2

IAC_SLEW_A 3

IAC_SLEW_B 4

IAC_SLEW_C 5

IDC_A 6

IDC_B 7

IDC_C 8

IDC_SLEW_A 9

IDC_SLEW_B 10

_SLEW_C 11

PHASE_MODE 12

START_A 13

START_B 14

START_C 15

R_A 16

R_B 17

R_C 18

R_A_SLEW 19
R_B_SLEW 20
R_C_SLEW 21
THREE_WAVE_A 22
THREE_WAVE_B 23
THREE_WAVE_C 24
THREE_TIME_MODE 25
THREE_TIME 26
THREE_TRIG_OUT 27
A_PHASE_SHIFT 29
B_PHASE_SHIFT 29
C_PHASE_SHIFT 30
A_wave_PARAM 31
B_WAVE_PARAM 32
C_WAVE_PARAM 33
<NRf+>: setting value

Syntax:

LIST:STEP:ITEM <NR1>,<NR1>,<NRf+>

Query syntax:

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

Example

LIST:STEP:ITEM 1,0,"100"
LIST:STEP:ITEM 1,22,"Sine"

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

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

Syntax:

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

Arguments

<NR1>,<item>

Returns:

<NRf+>

Example

LIST:STEP:ITEM? 1,6

LIST:NAME?

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

Syntax:

LIST:NAME?

Example

LIST:NAME?

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:

LIST:SAVe <filename>

Example

LIST:SAV "123.csv"

LIST:CONFigure

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

Syntax:

LIST:CONFigure

Example

LIST:CONF

LIST:CREate

This command is used to create a list.

Syntax:

LIST:CREate

Example

LIST:CRE

LIST:FILE:NUMBer?

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

Syntax:

LIST:FILE:NUMBer?

Example

LIST:FILE:NUMB?

LIST:FILE:NAME? <index>

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

Syntax:

LIST:FILE:NAME? <index>

Arguments

<NR1>

Example

LIST:FILE:NAME? 1

LIST:FILE:DELete <filename>

This command is used to delete the list file.

Syntax:

LIST:FILE:DELete <filename>

Arguments

<CPD>

Example

LIST:FILE:DEL "123.csv"

LIST:STEP:JUMP <NR1>

The command sets the step number of the list loop jump, for Example, if it is set to 2, the list loop skips the previous two steps and starts from step 3.

NR1: The minimum value is 0, i.e., no skipped steps and all steps are looped; the maximum value is less than the total number of steps.

Syntax:

LIST:STEP:JUMP <NR1>

Arguments

<NR1>

Query syntax:

LIST:STEP:JUMP? [MAXimum|MINimum]

Returns:

<NRf+>

Example

LIST:STEP:JUMP 2

LIST:FUNCtion <CPD>

This command is used to set list running mode. Such as CC mode or CR mode.

Syntax:

LIST:FUNCtion <CPD>

Arguments

AC mode: CC|CR

DC mode: CC|CR|CP

Query syntax:

LIST:FUNCtion?

Example

LIST:FUNCtion CC

LIST:DCCV:STATe <Boolean>

This command is used to set CV state of DC list mode.

Syntax:

LIST:DCCV:STATe <Boolean>

Arguments

<0|1|OFF|ON>

Query syntax:

LIST:DCCV:STATe?

Example

LIST:DCCV:STAT 0

LIST:DCCV:VOLTage <NRf+>

This command is used to set Voltage level of CV in DC List mode.

Syntax:

LIST:DCCV:VOLTage <NRf+>

Arguments

<NR1>

Query syntax:

LIST:DCCV:VOLTage?

Example

LIST:DCCV:VOLTage 40

Chapter18 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>

Arguments

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+>

Arguments

<NRf+>

单位

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+>

Arguments

<NRf+>

单位

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>

Arguments

IMMediate|PHASe

Query syntax:

SURGesag:ACTion?

Returns:

IMMediate|PHASe

Example

SURGesag:ACTion IMMediate //Set the present trigger action to IMMediate.

SURGesag:ACTion?

SURGesag:SYMMetry <bool>

This command is used to set and query the symmetry mode switch of the surge/trap wave.

Syntax:

SURGesag:SYMMetry <bool>

Arguments

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>

Arguments

<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>

Arguments

<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>

Arguments

PERCent|SETTing

Query syntax:

SURGesag:VALue:SELect?

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+>

Arguments

<NRf+>

*RST

0

Query syntax:

SURGesag:VALue:PERCent?

Returns:

<NRf+>

Example

SURGesag:VALue:PERCent 10 //Set the drop percentage to 10%

SURGesag:VALue:PERCent?

SURGesag:VALue:SETTing <NRf+>

This command is used to set and query the drop value of the trap wave.

<NRf+> drop set value

Syntax:

SURGesag:VALue:SETTing <NRf+>

Arguments

<NRf+>

*RST

0

Query syntax:

SURGesag:VALue:SETTing? [MAXimum| MINimum]

Returns:

<NRf+>

Example

SURGesag:VALue:SETT 20 //Set the drop value to 20V

SURGesag:VALue:SETT? //Query the set value of the drop value

SURGesag:VALue:SETT? MAX //Query the maximum value of the drop set value.

SURGesag:CHAN:ENABle <A|B|C|AB|AC|BC|ABC>

This command is used to set and query the phase of the trap dip, and is only available in three-phase mode.

Syntax:

SURGesag:CHAN:ENABle <A|B|C|AB|AC|BC|ABC>

Arguments

<A|B|C|AB|AC|BC|ABC>

Query syntax:

SURGesag:CHAN:ENABle?

Returns:

<NRf+>

Example

SURGesag:PHASe:ENABle ABC

SURGesag:SAVE <string>

This command is used to save this surge/sag file.

Syntax:

SURGesag:SAVE <string>

Example

SURGesag:SAVE "filename"

SURGesag:RECall <string>

This command is used to recall the surge/sag file.

Syntax:

SURGesag:RECall <string>

Example

SURGesag:RECall "filename"

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.

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.

<wave> :

- Sine
- Square
- Saw
- Triangle
- Trapezoid
- Clipped-sine
- DST-01(30)
- Filename

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 in Normal mode. For waveforms with parameters, the parameters are set to default values.

<wave>:

- Sine
- Square
- Saw
- Triangle
- Trapezoid
- Clipped-sine
- DST-01(30)
- Filename

Syntax:

PWAVeform[:IMMediate] <phase/chan>,<WAVEform>

Arguments

<A|B|C|CH1|CH2|CH3>,

<"Sine"|"Square"|"Triangle"|"Saw"|"Trapezoid"|"Clipped-sine"|"DST-01(30)"|"filename">

*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 waveform 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 SWEep Subsystem

SWEep:FUNCtion <mode>

This command is used to set Sweep mode.

AC mode: CC|CR

DC mode: CC|CP|CR

Syntax:

SWEep:FUNCtion <mode>

Arguments

CC|CR 或 CC|CP|CR

*RST

0

Query syntax:

SWEep:FUNCtion?

Returns:

CC|CP|CR

Example

SWE:FUNC CC

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

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

<NRf+>: initial setting value, or initial 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:LEVel:STARt <NRf+>[,<NRf+>,<NRf+>]

Arguments

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

*RST

0

Query syntax:

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

Returns:

<NR2>

Example

SWEep:LEVel:STARt 100 //Set the starting value of sweep to 100.

SWEep:LEVel:STARt?

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

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

<NRf+>: stop setting value, or stop 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:LEVel:STOP <NRf+>[,<NRf+>,<NRf+>]

Arguments

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

Query syntax:

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

Returns:

<NR2>

Example

SWEep:LEVel:STOP 90 // Set the stopping value of sweep to 100.

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

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

<NRf+>: step 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:LEVel:STEP <NRf+>[,<NRf+>,<NRf+>]

Arguments

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

Query syntax:

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

Returns:

<NR2>

Example

SWEep:LEVel:STEP 1 //set the step value to 1

SWEep:LEVel:STEP?

SWEep:TIME:STEP <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+>

Arguments

<NRf+>

Query syntax:

SWEep:TIME:STEP? [MAXimum|MINimum]

Returns:

<NR1>

Example

SWEep:TIME:STEP 1

SWEep:TIME:STEP?

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.

<mode> TIME|TRIG|TIME_LOOP|TRIG_LOOP

TIME_LOOP: Time loop sweep

TRIG_LOOP: trigger loop sweep

Syntax:

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

Arguments

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

Query syntax:

SWEep:MODE?

Returns:

<NR1>

Example

SWEep:MODE TIME

SWEep:WAVeform <name>

Select Sweep waveform.

<name>

<"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">,

Syntax:

SWEep:WAVeform <name>

Arguments

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

Query syntax:

SWEep:WAVeform?

Example

SWEep:WAVeform "Sine"

SWEep:WAVeform:PARameter <NRf+>

This command is used to set and query the waveform parameters (for Example, the percentage of clipping wave. For waveform types without parameters, this parameter is invalid).

Syntax:

SWEep:WAVeform:PARameter <NRf+>

Arguments

<NRf+>

Query syntax:

SWEep:WAVeform:PARameter?

Returns:

<NRf+>

Example

SWEep:WAVeform:PAR 1.414

SWEep:PSHift <NRf+>

This command is used to set the phase shift between voltage and current.

Syntax:

SWEep:PSHift <NRf+>

Arguments

TIMe|TRIG

*RST

TIMe

Query syntax:

SWEep:PSHift?

Returns:

TIMe|TRIG

Example

SWEep:MODe TIMe

SWEep:TERMinate <mode>

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

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

Syntax:

SWEep:TERMinate <mode>

Arguments

NORMal|LAST|OFF

Query syntax:

SWEep:TERMinate?

Returns:

NORMal|LAST|OFF

Example

SWEep:TERMinate OFF

SWEep:TRIG:SOURce <mode>

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

Syntax:

SWEep:TRIG:SOURce <mode>

Arguments

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>

Arguments

1-65535

Query syntax:

SWEep:STEP:REPeat? [MAXimum|MINimum]

Example

SWEep:STEP:REPeat <NR1>

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 electronic load. It returns a parameter that contains four segments divided by a comma.

Query syntax

*IDN?

Arguments

None

Returns

Manufacture, model, serial number, UI ver-DSP1 ver-DSP2 ver-PFC ver-Interface ver

Example

Itech\sElectronics,IT8215-350-90,804947011767740001,

000.001.038,83.R,13.R\n

*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 electronic load 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 electronic load 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.