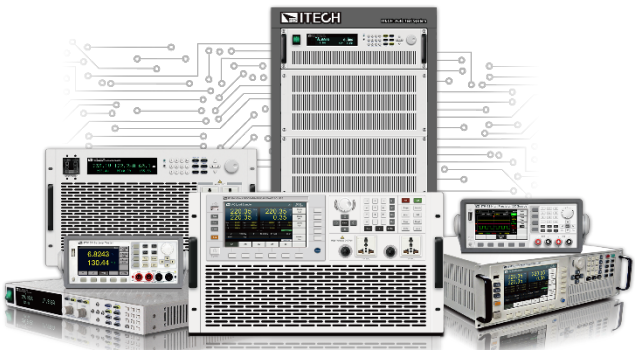
Regenerative DC Electronic Load

IT-M3300 Series Programing Guide



Model: IT-M3300 Series
Version: V2.0 / 01, 2026

Notices

Manual Part Number

IT-M3400

Revision

2nd Edition: Jan. 6, 2026

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 provides this customary commercial license in software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data – Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

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 IT-M3300 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.

Waste Electrical and Electronic Equipment (WEEE) Directive

2002/96/EC Waste Electrical and Electronic Equipment (WEEE) Directive

This product complies with the WEEE Directive (2002/96/EC) marking requirement. This affix product label indicates that you must not discard the electrical/electronic product in domestic household waste.

Product Category

With reference to the equipment classifications described in the Annex I of the WEEE Directive, this instrument is classified as a “Monitoring and Control Instrument”.

To return this unwanted instrument, contact your nearest ITECH office.

Compliance Information

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

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

Conforms with the following product standards:

EMC Standard

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

Reference Standards

CISPR 11:2009+A1:2010/ EN 55011:2009+A1:2010 (Group 1, Class A)

IEC 61000-4-2:2008/ EN 61000-4-2:2009

IEC 61000-4-3:2006+A1:2007+A2:2010/ EN 61000-4-3:2006+A1:2008+A2:2010

IEC 61000-4-4:2004+A1:2010/ EN 61000-4-4:2004+A1:2010

IEC 61000-4-5:2005/ EN 61000-4-5:2006

IEC 61000-4-6:2008/ EN 61000-4-6:2009

IEC 61000-4-11:2004/ EN 61000-4-11:2004

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

Safety Standard

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

Content

Chapter1 SCPI Introduction

1.1 Overview

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

OUTPut:

• SYNC {OFF|0|ON|1}

• SYNC:

– MODE {NORMal|CARRier}

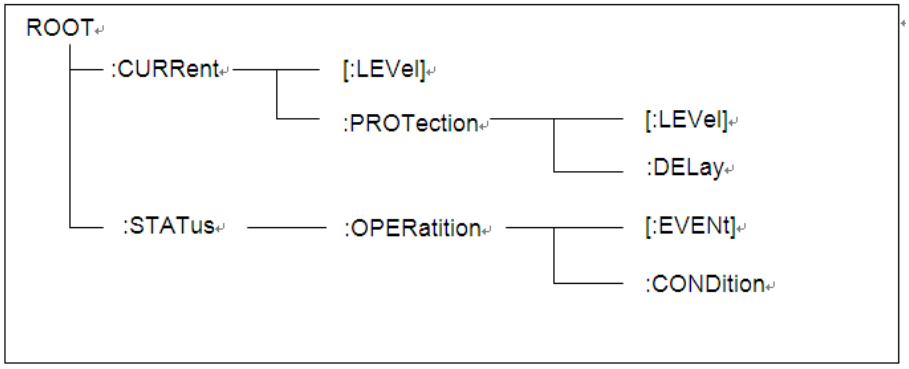
– POLarity {NORMal|INVerted}

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

1.2 Command Type of SCPI

SCPI has two types of commands, common and subsystem.

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



Multiple Commands in a Message

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

- Use a semicolon to separate commands within a message.
- Head paths influence how the instrument interprets commands.

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

CURR:LEV 3;PROT:STAT OFF

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

CURR:PROT:STAT OFF

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

Movement in the Subsystem

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

PROTection:CLEAr;:STATus:OPERation:CONDition?

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

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

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

Including Common Commands

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

VOLTage:TRIGgered 17.5;:INITialize;*TRG

OUTPut OFF;*RCL 2;OUTPut ON

Case Sensitivity

Common commands and SCPI commands are not case sensitive. You

can use upper or lower, for example:

***RST = *rst**

:DATA? = :data?

:SYSTem:PRESet = :system:preset

Long-Form and Short-Form Versions

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

:SYSTem:PRESet long-form

:SYST:PRES short form

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

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

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

Query

Observe the following precautions with queries:

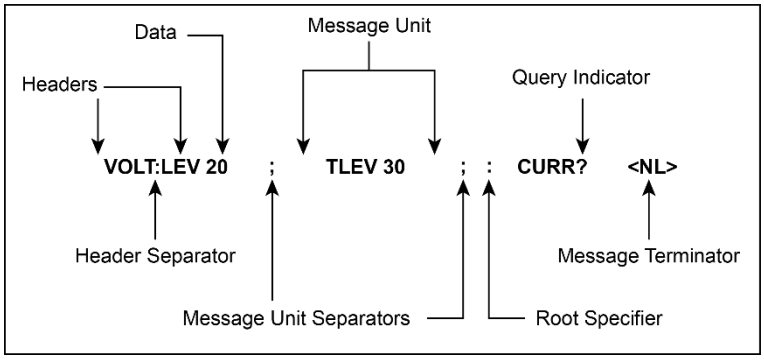
- Set up the proper number of variables for the returned data. For example, if you are reading back a measurement array, you must dimension the array according to the number of measurements that you have placed in the measurement buffer.
- Read back all the results of a query before sending another command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

1.3 Message Type of SCPI

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

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

The next figure illustrates SCPI message structure:



The message unit

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

ABORt<NL>

VOLTage 20<NL>

Headers

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

Query indicator

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

Message unit separator

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

Root specifier

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

Message terminator

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

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

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

Command execution rules

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

1.4 Response Data Type

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

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

Response messages

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

Sending a response message

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

Multiple response messages

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

0; 1; 1; 0

Response message terminator (RMT)

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

0; 1; 1; 0; <RMT>

Message exchange protocol

Two rules summarize the message exchange protocol:

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

The following two steps must always be performed to send information from the instrument other computer:
1. Send the appropriate query command(s) in a program message.
2. Address the instrument to talk.

- **Rule 2**: The complete response message must be received by the computer before another program message can be sent to the instrument.

1.5 Command Format

Formats for command display are as follows:

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

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

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

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

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

Colon (:)

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

APPL:SIN 455E3,1.15,0.0

In this example, APPLy command assigns a sine wave with frequency of 455

KHz, amplitude of 1.15 V and DC offset of 0.0 V.

Semicolon (;)

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

TRIG:SOUR EXT; COUNT 10

has the same effect as sending the following two commands:

TRIG:SOUR EXT
TRIG:COUNT 10

Question mark (?)

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

TRIG:COUN 10

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

TRIG:COUN?

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

TRIG:COUN?MIN
TRIG:COUN?MAX

Comma (,)

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

Space

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

Common commands (*)

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

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

Command terminator

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

NOTE

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

1.6 Data Type

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

- Numerical parameter

  Commands requiring numerical parameter support the notations of all common decimal notations, including optional signs, decimal points, scientific notation, etc. Special values of numerical parameter are also acceptable, such as MIN, MAX and DEF. In addition, suffixes for engineering units can also be sent together with numerical parameter (including M, k, m or u). If the command accepts only some specific values, the instrument will automatically round the input parameter to acceptable values. The following commands require numerical parameter of frequency value:
  [SOURce[1|2]:]FREQuency:CENTer {<Frequency>|MINimum|MAXimum}
  – **<NR1>**: represents an integer value, such as 273;
  – **<NR2>**: represents a real number in floating-point format, such as .273;
  – **<NR3>**: represents a real number in scientific notation, such as 2.73E+2;
  – **<Nrf>**: The extensible form includes <NR1>, <NR2> and <NR3>;
  – **<Nrf+>**: The extensible decimal form includes <Nrf>, MIN, MAX and DEF. MIN and MAX are the minimum and maximum finite number. Within the range of the parameter definition, DEF is the default of the parameter.

- Discrete parameter

  Discrete parameter are used for settings with limited number of programming values (such as IMMediate, EXTernal or BUS). They can use short and long format like key words of commands. They may be expressed in both upper and lower case. The query response always returns uppercase Parameter in short format. The following commands require discrete parameter in voltage unit:
  [SOURce[1|2]:]VOLTage:UNIT {VPP|VRMS|DBM}

- Boolean parameter

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

- ASCII string parameter

  String parameter may actually include all ASCII character sets. Character strings must start and end with paired quotation marks; and single quotation

marks or double quotation marks are both allowed. Quotation mark separators may also act as one part of a string, they can be typed twice without any character added between them. String parameter is used in the following command:

DISPlay:TEXT <quoted string>

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

DISP:TEXT "WAITING..."

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

DISP:TEXT 'WAITING...'

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

– **<CPD>**: character program data.

1.7 Remote Interface Connections

Please refer to user manual for detailed introductions of the remote interface.

Note

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.

Chapter2 Example of Common Commands

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

Note

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

Example 1: Identify the power supply

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

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

-> *IDN?

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

-> SYST:ERR?

Example 2: Common output commands

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

-> VOLT 10.00 //Set the output voltage to 10 V.

-> CURR 3.500 //Set the output current to 3.5 A.

-> APPL 10.00,3.500 // Set the output voltage to 10V and the output current to 3.5A simultaneously.

-> FUNC:PRI VOLT|CURR //set the output CV priority or CC priority

-> INP:DEL 1.0 // set the output on delay time to 1 second.

-> INP:DEL:FALL 1.0 //set the output off delay time to 1 second.

-> INP ON // set the output on.

-> TIM ON //enable the output timing function.

-> TIM:DEL 100 //Set the power supply output to a fixed time of 100 seconds.

Example 3: List function commands

-> TRIG:LIST:SOUR KEYP //set the trigger source to manual trigge.

-> LIST:FUNC VOLT //Set the operating mode of the list to CV mode.

-> LIST:TERM NORM //Set the operational state after List execution completes to normal stop mode.

-> LIST:REP 3 //Set the list iteration count to 3 times.

-> LIST:STEP:COUN 10 //set the list file count to 10 steps

-> LIST:STEP:VOLT 1,10.00 //Set the voltage value for List Step 1 to 10V

-> LIST:STEP:CURR 1,3.500 //Set the current value for List Step 1 to 3.5A

-> LIST:STEP:SLEW 1,1.000 //Set List Step 1 slope to 1S

-> LIST:STEP:WIDT 1,1.000 //Set the duration of List Step 1 to 1 second.

-> LIST:SAV 1 // Specified list saved in List File 1

-> LIST ON 或 FUNC:MODE LIST //enable the list function

-> INP ON //turn on the output

-> INIT:LIST //running the list function

Example 4: Battery Charge Test

-> PWM OFF // Before running the battery charging test, send the following command to disable the constant current source function.

-> TRIG:BATT:SOUR KEYP //set the trigger source to manual trigge.

-> BATT:DISC:VOLT 5.0 //set the charge voltage to 5.0V

-> BATT:DISC:CURR 3.0 //set the charge current to 3.0A

-> BATT:SHUT:TIME 3.0 //set the charge time to 3s.

-> BATT:SHUT:VOLT 4.0 //set the cutting voltage of battery charge to 4V.

-> BATT:SHUT:CURR 3.0 //set the cutting current of battery charge to 3A.

-> BATT:SHUT:CAP 3.0 //Set the battery capacity for shutdown to 3AH

-> BATT ON //enable the battery function

Example 5: Trace function

-> TRAC:FEED:CONT NEVer //Disable the trace function

-> TRAC:CLE // Clear cache data

-> TRAC:FEED:CONT ALWays //Configure the cache function to operate in a cyclic storage mode.

-> TRAC:FEED:SEL BOTH //Configure the cache data source to simultaneously cache voltage and current data.

-> TRAC:POIN 1000 //Configure the cache cycle storage size to hold 1000 data entries.

-> TRAC:DEL 0.001 //Configure cache trigger delay time to 1ms

-> TRAC:TIM 1.000 // Configure the cache data sampling interval to 1 second.

-> VOLT 10 //set the output voltage to 10V

-> CURR:LIM 2.0 //set the current limit to 2A

-> INP ON //enable the output

-> *TRG //trigger the trace function

-> TRAC:POIN:ACT? //Query the number of currently cached data points

-> TRAC:DATA? // Retrieve cached data

Chapter3 SCPI status register

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

The following table describes the status signals.

Bit name	Bit	Decimal value	Definition
Questionable Status Register			
OV	0	1	Output is disabled by the over-voltage protection.
OC	1	2	Output is disabled by the over-current protection.
OP	2	4	Output is disabled by the over-power protection.
UV	3	8	Output is disabled by the under-voltage protection.
UUT_OT	4	16	Output is disabled by the UUT over-temperature protection.
UC	5	32	Output is disabled by the under-current protection.
SRvs	6	64	Sense malfunction.
LINE	7	128	Off Line.
Rvs	8	256	Output terminals reversed
BUS	9	512	Output is disabled by the external command
Wdog	10	1024	Watchdog protection.
OT	11	2048	Over-temperature protection.
FAN_FAIL	12	4096	Unknow fault internal
TEMP_SENSE_FAIL	13	8192	Temperature sensor fault
AC_LOSS	14	16384	Power down
FLDBK	15	32768	Foldback protection
Operation Status Register			
Priority	0		CV /CC mode priority
Cal	1	2	The power supply is under calibration.
List	2-3	4-8	The power supply is running the list program. 0: idle, 1: wtg, 2: run, 3: end;
CV	4	16	Output is in constant voltage.
CC	5	32	Output is in constant current.
CW	6	56	Output is in constant current.
On_Delay	8	256	The power supply is in the output-on delay.
Off_Delay	9	512	The power supply is in the output-off delay.
On	10	1024	Output is programmed on.
List Pause	11	2048	The list program pauses.

Standard Event Register			
OPC	0	1	All commands before and including *OPC have been executed.
QYE	2	4	The instrument tried to read the output buffer but it was empty, a new command line was received before a previous query has been read, or both the input and output buffers are full.
DDE	3	8	A device-specific error, including a self-test error, calibration error or other device-specific error occurred.
EXE	4	16	An execution error occurred.
CME	5	32	A command syntax error occurred.
NU	6	not used	0 is returned.
PON	7	128	Power has been cycled since the last time the event register was read or cleared.
Status Byte Register			
NU	0	not used	0 is returned.
NU	1	not used	0 is returned.
EAV	2	4	Error buffer available.
QUES	3	8	This bit is set to 1 when any one status of enabled query status register changes.
MAV	4	16	Output buffer available.
ESB	5	32	Bit ESB is set to 1 when the status of an enabled standard event status.
RQS/MSS	6	64	Register changes.
OPER	7	128	If the status of enabled operation register changes, then this bit is set to1.

Chapter4 Status Commands

STATus:OPERation[:EVENt]? [(@chanlist)]

This command can read the parameter from operation event register. After executing this order, operation event register is reset.

Syntax

STATus:OPERation[:EVENt]? [(@chanlist)]

Arguments

None

Example

STAT:OPER?

Returns

NR1

STATus:OPERation:CONDition? [(@chanlist)]

This command can read the parameter from the operation condition register. When the parameter of operation condition register changes, the bit corresponding in operation event register is 1.

Syntax

STATus:OPERation:CONDition? [(@chanlist)]

Arguments

None

Example

STAT:OPER:COND? [(@chanlist)]

Returns

NR1

STATus:OPERation:ENABle <n>[,(@chanlist)]

This command can set the parameter of operation event enable register. Setting parameter can determine which bit value of operation event register is 1 and the bit will cause OPER of status byte register to be 1.

Syntax

STATus:OPERation:ENABle <n>[,(@chanlist)]

Arguments

<0-65535>

Example

STAT:OPER:ENAB 16

Query syntax

STATus:OPERation:ENABle? [(@chanlist)]

Returns

NR1

STATus:OPERation:NTRansition <n>[,(@chanlist)]

This command is used to edit the negative transition trigger register of operation event.

Syntax

STATus:OPERation:NTRansition <n>[,(@chanlist)]

Arguments

<0-65535>

Example

STAT:OPER:NTR 16

Query syntax

STATus:OPERation:NTRansition? [(@chanlist)]

Returns

NR1

STATus:OPERation:PTRansition <n>[,(@chanlist)]

This command is used to edit the positive transition trigger register of operation event.

Syntax

STATus:OPERation:PTRansition <n>[(@chanlist)]

Arguments

<0-65535>

Example

STAT:OPER:PTR 32

Query syntax

STATus:OPERation:PTRansition? [(@chanlist)]

Returns

NR1

STATus:QUEStionable[:EVENt]? [(@chanlist)]

This command can be used to read the value in query event register.The power supply will return a decimal number corresponding to the binary weighted sum of each unit digit of register. These digits will be latched. After executing this command, the query event register will be cleared.

Syntax

STATus:QUEStionable[:EVENt]? [(@chanlist)]

Arguments

None

Example

STAT:QUES?

Returns

NR1

STATus:QUEStionable:CONDition? [(@chanlist)]

This command is used to read the value of query condition register and gets the status of power supply.

Syntax

STATus:QUEStionable:CONDition? [(@chanlist)]

Arguments

None

Example

STAT:QUES:COND?

Returns

NR1

STATus:QUEStionable:ENABle <n>[,(@chanlist)]

This command edits the enable register value of query event. In query, the power supply will return a decimal number representing the binary weighted sum of the enable register.

Syntax

STATus:QUEStionable:ENABle <n>[,(@chanlist)]

Arguments

<0-65535>

Example

STAT:QUES:ENAB 24

Query syntax

STATus:QUEStionable:ENABle? [(@chanlist)]

Returns

NR1

STATus:QUEStionable:NTRansition <n>[,(@chanlist)]

This command is used to edit the negative transition trigger register of query event.

Syntax

STATus:QUEStionable:NTRansition <n>[,(@chanlist)]

Arguments

<0-65535>

Example

STAT:QUES:NTR 64

Query syntax

STATus:QUEStionable:NTRansition? [(@chanlist)]

Returns

NR1

STATus:QUEStionable:PTRansition <n>[,(@chanlist)]

This command is used to edit the positive transition trigger register of query event.

Syntax

STATus:QUEStionable:PTRansition <n>[,(@chanlist)]

Arguments

<0-65535>

Example

STAT:QUES:PTR 32

Query syntax

STATus:QUEStionable:PTRansition? [(@chanlist)]

Returns

NR1

STATus:PRESet

This command is used to clear the register status as follows:

- Enable the Questionable status, PTR register and NTR register
- Enable the operation status, PTR register and NTR register

Syntax

STATus:PRESet

Example

STAT:PRES

Chapter5 Channel Selection Commands

CHANnel <NR1>

This command selects the channel number.

Syntax

CHANnel <NR1>

Arguments

0-16

Default value

0

Example

CHAN 2

Query syntax

CHANnel?

Returns

NR1

INSTrument[:SELect] <NR1>

This command selects the channel. The function is the same as the command CHANnel <NR1>.

Syntax

INSTrument[:SELect] <NR1>

Arguments

0-16

Default value

0

Example

INST[:SEL] 2

Query syntax

INSTrument[:SELect]?

Returns

NR1

Chapter6 System Commands

SYSTem:BEEPer:IMMediate [(@chanlist)]

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

Syntax

SYSTem:BEEPer:IMMediate [(@chanlist)]

Arguments

None

Example

SYST:BEEP:IMM

Query syntax

None

Returns

None

SYSTem:BEEPer[:STATe] <bool>[,(@chanlist)]

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

Syntax

SYSTem:BEEPer[:STATe] <bool>[,(@chanlist)]

Arguments

0|OFF|1|ON

Default value

1

Example

SYST:BEEP 1

Query syntax

SYSTem:BEEPer[:STATe]? [(@chanlist)]

Returns

0|1

SYSTem:ERRor?

This command reads the error code and error information.

Syntax

SYSTem:ERRor?

Arguments

None

Example

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

Note

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

Returns

AARD

SYSTem:CLEar

This command clears the system status register.

Syntax

SYSTem:CLEar

Arguments

None

Example

SYST:CLE

Query syntax

None

Returns

None

SYSTem:REMote

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

Syntax

SYSTem:REMote

Arguments

None

Example

SYST:REM

Query syntax

None

Returns

None

SYSTem:LOCal

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

Syntax

SYSTem:LOCal

Arguments

None

Example

SYST:LOC

Query syntax

None

Returns

None

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

Example

SYST:RWL

Query syntax

None

Returns

None

SYSTem:KEY <n>[,(@chanlist)]

This command and its query form set and read the value of the Operation Enable register.

Syntax

SYSTem:KEY <n>[,(@chanlist)]

Arguments

The definition of keyboard is listed as follows:

Arguments	keys	Arguments	keys
1	Link	9	I-set
2	Shift	10	R-set
3	Source	11	Enter
4	Load	12	Left
5	Esc	13	Right
6	V-set	14	Decrease
7	Save	15	Increase
8	On/Off	-	-

Example

SYST:KEY 5

Query syntax

SYSTem:KEY? [(@chanlist)]

Returns

n

SYSTem:COMMunicate:GPIB[:SELF]:ADDRess <addr>[,(@chanlist)]

This command sets the GPIB address of the power supply.

Syntax

SYSTem:COMMunicate:GPIB[:SELF]:ADDRess <addr>[,(@chanlist)]

Arguments

NR1 <0-30>

Default value

15

Example

SYST:COMM:GPIB:ADDR 14

Query syntax

SYSTem:COMMunicate:GPIB[:SELF]:ADDRess? [(@chanlist)]

Returns

NR1 (0-30)

SYSTem:COMMunicate:LAN:CURRent:ADDRess <"addr">

This command sets the IP address of the power supply.

Syntax

SYSTem:COMMunicate:LAN:CURRent:ADDRess <"addr">

Arguments

Str "<0-255>,<0-255>,<0-255>,<0-255>"

Example

SYST:COMM:LAN:CURR:ADDR "192.168.0.211"

Query syntax

SYSTem:COMMunicate:LAN:CURRent:ADDRess?

Returns

Str

SYSTem:COMMunicate:LAN:CURRent:DGATeway <"addr">

This command sets the gateway of the power supply.

Syntax

SYSTem:COMMunicate:LAN:CURRent:DGATeway <"addr">

Arguments

Str "<0-255>,<0-255>,<0-255>,<0-255>"

Example

SYST:COMM:LAN:CURR:DGAT “192.168.0.1”

Query syntax

SYSTem:COMMunicate:LAN:CURRent:DGATeway?

Returns

Str

SYSTem:COMMunicate:LAN:CURRent:SMASk <"addr">

This command sets the subnet mask of the power supply.

Syntax

SYSTem:COMMunicate:LAN:CURRent:SMASk <"addr">

Arguments

Str "<0-255>,<0-255>,<0-255>,<0-255>"

Example

SYST:COMM:LAN:CURR:SMAS “255.255.255.0”

Query syntax

SYSTem:COMMunicate:LAN:CURRent:SMASk?

Returns

Str

SYSTem:COMMunicate:LAN:DHCP <Bool>

This command enables or disables the dynamic IP address function.

Syntax

SYSTem:COMMunicate:LAN:DHCP <Bool>

Arguments

<0|1|OFF|ON>

Example

SYST:COMM:LAN:DHCP 1

Query syntax

SYSTem:COMMunicate:LAN:DHCP?

Returns

0|1

SYSTem:COMMunicate:LAN:RAWSocket:PORT

This command sets the socket port for the LAN communication.

Syntax

SYSTem:COMMunicate:LAN:RAWSocket:PORT <NR1>

Arguments

<2000-65535>

Example

SYST:COMM:LAN:RAWS:PORT 30000

Query syntax

SYSTem:COMMunicate:LAN:RAWSocket:PORT?

Returns

NR1

SYSTem:COMMunicate:LAN:MACaddress?

This command queries the communication MAC address.

Syntax

SYSTem:COMMunicate:LAN:MACaddress?

Arguments

None

Example

- > SYST:COMM:LAN:MAC?
< - "12:34:56:79:99:AA"

Note

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

Returns

str

SYSTem:COMMunicate:SERial:BAUDrate

This command sets the baud rate of the serial port.

Syntax

SYSTem:COMMunicate:SERial:BAUDrate

Arguments

<4800|9600|19200|38400|57600|115200>

Example

SYST:COMM:SER:BAUD 9600

Query syntax

SYSTem:COMMunicate:SERial:BAUDrate?

Returns

4800|9600|19200|38400|57600|115200

SYSTem:VERSion?

This command queries the SCPI version of the instrument.

Syntax

SYSTem:VERSion?

Arguments

None

Example

- > SYST:VERS?
< - "1993.1"

Returns

AARD

SYSTem:COMMunicate:LAN:DNS1 <dns>

This command sets DNS primary address for LAN.

Syntax

SYSTem:COMMunicate:LAN:DNS1 <dns>

Arguments

SPD

Example

SYST:COMM:LAN:DNS1 "192.168.0.1"

Query syntax

SYSTem:COMMunicate:LAN:DNS1?

Returns

SPD

SYSTem:COMMunicate:LAN:DNS2 <dns>

This command sets DNS secondary address for LAN.

Syntax

SYSTem:COMMunicate:LAN:DNS2 <dns>

Arguments

SPD

Example

SYST:COMM:LAN:DNS1 "192.168.0.2"

Query syntax

SYSTem:COMMunicate:LAN:DNS2?

Returns

SPD

SYSTem:COMMunicate:LAN:RAWSocketport <port>

This command sets the socket port for the LAN communication.

Syntax

SYSTem:COMMunicate:LAN:RAWSocketport <port>

Arguments

2000-65535

Example

SYST:COMM:LAN:RAWS 30000

Query syntax

SYSTem:COMMunicate:LAN:RAWSocketport?

Returns

NR1

SYSTem:COMMunicate:LAN:MDNS <onoff>

This command is used to set the status of MDNS function.

Syntax

SYSTem:COMMunicate:LAN:MDNS <onoff>

Arguments

0|OFF|1|ON

Example

SYST:COMM:LAN:MDNS 1

Query syntax

SYSTem:COMMunicate:LAN:MDNS?

Returns

0|1

SYSTem:COMMunicate:LAN:PING <onoff>

This command is used to set the status of MDNS function.

Syntax

SYSTem:COMMunicate:LAN:PING <onoff>

Arguments

0|OFF|1|ON

Example

SYST:COMM:LAN:PING 1

Query syntax

SYSTem:COMMunicate:LAN:PING?

Returns

0|1

SYSTem:COMMunicate:LAN:TELNet <onoff>

This command is used to set the status of telnet function.

Syntax

SYSTem:COMMunicate:LAN:TELNet <onoff>

Arguments

0|OFF|1|ON

Example

SYST:COMM:LAN:TELNet 1

Query syntax

SYSTem:COMMunicate:LAN:TELNet?

Returns

0|1

SYSTem:COMMunicate:LAN:WEB <onoff>

This command is used to set the status of WEB function.

Syntax

SYSTem:COMMunicate:LAN:WEB <onoff>

Arguments

0|OFF|1|ON

Example

SYST:COMM:LAN:WEB 1

Query syntax

SYSTem:COMMunicate:LAN:WEB?

Returns

0|1

SYSTem:COMMunicate:LAN:VXI11 <onoff>

This command is used to set the status of VXI-11 function.

Syntax

SYSTem:COMMunicate:LAN:VXI11 <onoff>

Arguments

0|OFF|1|ON

Example

SYST:COMM:LAN:VXI11 1

Query syntax

SYSTem:COMMunicate:LAN:VXI11?

Returns

0|1

SYSTem:COMMunicate:LAN:RESTore

This command resets the LAN settings to default settings.

Syntax

SYSTem:COMMunicate:LAN:RESTore

Arguments

None

Example

SYST:COMM:LAN:REST

Query syntax

None

Returns

None

SYSTem:COMMunicate:LAN:SAVE

This command makes the LAN settings valid.

Syntax

SYSTem:COMMunicate:LAN:SAVE

Arguments

None

Example

SYST:COMM:LAN:RES

Query syntax

None

Returns

None

SYSTem:COMMunicate:LAN:STATe?

This command queries the LAN state.

Syntax

SYSTem:COMMunicate:LAN:STATe?

Arguments

None

Default value

DOWN

Example

SYST:COMM:LAN:STAT?

Returns

DOWN|UP

SYSTem:COMMunicate:LAN:HOSTname?

This command queries the host name in the LAN communication.

Syntax

SYSTem:COMMunicate:LAN:HOSTname?

Arguments

None

Example

SYST:COMM:LAN:HOST?

Returns

SPD

SYSTem:COMMunicate:LAN:DESCription?

This command queries the host name description in the LAN communication.

Syntax

SYSTem:COMMunicate:LAN:DESCription?

Arguments

None

Example

SYST:COMM:LAN:DESC?

Returns

SPD

SYSTem:COMMunicate:LAN:DOMain?

This command queries the domain in the LAN communication.

Syntax

SYSTem:COMMunicate:LAN:DOMain?

Arguments

None

Example

SYST:COMM:LAN:DOM?

Returns

SPD

[SOURce:]EXTernal[:STATe] <bool>

This command enables or disables the external analog quantity function of the power supply.

Syntax

[SOURce:]EXTernal[:STATe] <bool>

Arguments

<0|1|OFF|ON>

Default value

0

Example

EXT 1

Query syntax

[SOURce:]EXTernal[:STATe]?

Returns

0|1

ADDRess <NR1>|MINimum|MAXimum

This command is used to set instrument address which communicate through RS232 and RS485 interface.

Syntax

ADDRess <NR1>|MINimum|MAXimum

Arguments

0-127

Default value

None

Example

ADDR 1

SYSTem:BOOT:VERSion?

This command is used to query the boot version of optional accessories.

Syntax

SYSTem:BOOT:VERSion?

Arguments

None

Default value

None

Example

SYST:BOOT:VERS?

Returns

AARD

SYSTem:COMMunicate:PROTocol <DEFault|EXT1>[,(@chanlist)]

This command is used to switch the SCPI instruction set.

Syntax

SYSTem:COMMunicate:PROTocol <DEFault|EXT1>[,(@chanlist)]

Arguments

<DEFault|EXT1>

Default value

DEFault

Example

SYST:COMM:PROT DEF

Query syntax

SYSTem:COMMunicate:PROTocol? [(@chanlist)]

Returns

CPD

SYSTem:FAN:MAXimum:SPEed <NRf+>[,(@chanlist)]

This command is used to set the maximum speed limit for the fan.

Syntax

SYSTem:FAN:MAXimum:SPEed <NRf+>[,(@chanlist)]

Arguments

MINimum-MAXimum|MINimum|MAXimum

Default value

DEFault

Example

SYST:FAN:MAX:SPE MAX

Query syntax

SYSTem:FAN:MAXimum:SPEed?[MINimum|MAXimum][,(@chanlist)]

Returns

NRf+

SYSTem:READy? [(@chanlist)]

This command is used to check whether system initialization has completed.

Syntax

SYSTem:READy? [(@chanlist)]

Arguments

Null

Default value

Null

Example

SYST:READ?

Returns

0|1

Chapter7 Measure & Fetch Commands

MEASure[:SCALar]:CURRent[:DC]? [(@chanlist)]

FETCh[:SCALar]:CURRent[:DC]? [(@chanlist)]

这条命令用来返回测量的电流的平均值。

Syntax

MEASure[:SCALar]:CURRent[:DC]?

FETCh[:SCALar]:CURRent[:DC]?

Arguments

无

Default value

不适用

Example

MEAS:CURR?

FETC:CURR?

Returns

NRf

MEASure[:SCALar]:POWer[:DC]? [(@chanlist)]

FETCh[:SCALar]:POWer[:DC]? [(@chanlist)]

这条命令用来返回测量的功率的平均值。

Syntax

MEASure[:SCALar]:POWer[:DC]? [(@chanlist)]

FETCh[:SCALar]:POWer[:DC]? [(@chanlist)]

Arguments

无

Default value

不适用

Example

MEAS:POW?

FETC:POW?

Returns

NRf

MEASure[:SCALar]:VOLTage[:DC]? [(@chanlist)]

FETCh[:SCALar]:VOLTage[:DC]? [(@chanlist)]

这条命令用来返回测量的电压的平均值。

Syntax

MEASure[:SCALar]:VOLTage[:DC]? [(@chanlist)]

FETCh[:SCALar]:VOLTage[:DC]? [(@chanlist)]

Arguments

无

Default value

不适用

Example

MEAS:VOLT?

FETC:VOLT?

Returns

NRf

MEASure[:SCALar][:EXTernal]:TEMPerature? [(@chanlist)]

FETCh[:SCALar][:EXTernal]:TEMPerature? [(@chanlist)]

这条命令用来测量返回外部待测物温度。

Syntax

MEASure[:SCALar][:EXTernal]:TEMPerature? [(@chanlist)]

FETCh[:SCALar][:EXTernal]:TEMPerature? [(@chanlist)]

Arguments

无

Default value

不适用

Example

MEAS:TEMP?

FETCh:TEMP?

Returns

NRf

FETCh:AHOur? [(@chanlist)]

这条命令用来返回累积的安培-小时值。

Syntax

FETCh:AHOur? [(@chanlist)]

Arguments

无

Default value

不适用

Example

FETC:AHO?

Returns

NRf

FETCh[:SCALar]:CAPacity? [(@chanlist)]

这条命令用来返回电池充放电的容量。

Syntax

FETCh[:SCALar]:CAPacity? [(@chanlist)]

Arguments

无

Default value

不适用

Example

FETC:CAP?

Returns

NRf

FETCh[:SCALar]:ACMeter:EACTotal? [(@chanlist)]

这条命令用来返回回馈到电网上的能量。

Syntax

FETCh[:SCALar]:ACMeter:EACTotal? [(@chanlist)]

Arguments

无

Default value

不适用

Example

FETC:ACM:EACT?

Returns

NRf

FETCh:WHOur? [(@chanlist)]

这条命令用来返回累积的瓦特-小时值。

Syntax

FETCh:WHOur? [(@chanlist)]

Arguments

无

Default value

不适用

Example

FETC:WHO?

Returns

NRf

MEASure[:SCALar][:ALL]? [(@chanlist)]

FETCh[:SCALar][:ALL]? [(@chanlist)]

这条命令用来返回测量的电压,电流和功率，安时，瓦时。

Syntax

MEASure[:SCALar][:ALL]? [(@chanlist)]

FETCh[:SCALar][:ALL]? [(@chanlist)]

Arguments

无

Default value

不适用

Example

FETC?

MEAS?

Returns

NRf

Chapter8 Input Commands

INPut[:STATe][:ALL] <bool>[,(@chanlist)]

This command sets the input state of the electronic load.

Syntax

INPut[:STATe][:ALL] <bool>[,(@chanlist)]

Arguments

0|OFF|1|ON

Default value

0

Example

INP 1

Query syntax

INPut[:STATe]? [(@chanlist)]

Returns

0|1

[INPut:]PROTection:CLEar [(@chanlist)]

This command clears the protection status.

Syntax

[INPut:]PROTection:CLEar [(@chanlist)]

Arguments

None

Example

PROT:CLE

Query syntax

None

Returns

None

INPut:DELay[:RISE] <NRf+>[,(@chanlist)]

This command sets the delay time before turning the input on.

Syntax

INPut:DELay[:RISE] <NRf+>[,(@chanlist)]

Arguments

MINimum-MAXimum|MINimum|MAXimum

Default value

0

Example

INP:DEL 1.0

Query syntax

INPut:DELay[:RISE]? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

INPut:DELay:FALL <NRf+>[,(@chanlist)]

This command sets the delay time after turning the input off.

Syntax

INPut:DELay:FALL <NRf+>[,(@chanlist)]

Arguments

MINimum-MAXimum|MINimum|MAXimum

Default value

0

Example

INP:DEL:FALL 1.0

Query syntax

INPut:DELay:FALL? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

INPut:PON[:STATe] <RST|LAST|LOFF>[,(@chanlist)]

This command controls the settings and output state when the electronic load is powered on.

Syntax

INPut:PON[:STATe] <RST|LAST|LOFF>[,(@chanlist)]

Arguments

<RST|LAST|LOFF>

Default value

RST

Example

INP:PONS LAST

Query syntax

INP:PONS? [(@chanlist)]

Returns

<RST|LAST|LOFF>

INPut:SHORt[:STATe] <BOOLEAN>[,(@chanlist)]

This command sets the electronic load to short mode.

Syntax

[SOURce:]INPut:SHORt[:STATe] <BOOLEAN>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Default value

OFF

Example

INP:SHOR 1

Query syntax

[SOURce:]INPut:SHORt[:STATe]? [(@chanlist)]

Returns

0|1

INPut:REVerse[:STATe]? [(@chanlist)]

This command is used to query the connection of input terminals.

Syntax

INPut:REVerse[:STATe]? [(@chanlist)]

Arguments

None

Default value

None

Example

INP:REV?

Returns

0|1

0: indicates NOT reversed, 1: indicates reversed.

INPut:SDS[:STATe]? [(@chanlist)]

This command is used to query the SDS status.

Syntax

INPut:SDS[:STATe]? [(@chanlist)]

Arguments

None

Default value

None

Example

INP:SDS?

Returns

0|1

0: indicates not connecte SDS, 1: indicates the SDS is connected.

INPut:SDS:ENABle <0|OFF|1|ON>[,(@chanlist)]

This command is used to enable or disable the SDS module.1: indicates enable, 0: indicates disable.

Syntax

INPut:SDS:ENABle <0|OFF|1|ON>[,(@chanlist)]

Arguments

<0|OFF|1|ON>

Default value

1

Example

INP:SDS:ENAB 0

Returns

0|1

INPut:SDS:DC:RELay <0|OFF|1|ON>[,(@chanlist)]

This command is used to open or close the main circuit DC relay, 0 indicates open, 1 indicates close.

Syntax

OUTPut:SDS:DC:RELay <0|OFF|1|ON>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Default value

ON

Example

OUTP:SDS:DC:REL 0

Returns

0|1

Query syntax

OUTPut:SDS:DC:RELay? [(@chanlist)]

INPut:SDS:SENSe:RELay <0|OFF|1|ON>[,(@chanlist)]

This command is used to switch the SDS relay to local measurement or remote sense. 0 indicates local measurement. 1 indicates remote sensing.

Syntax

OUTPut:SDS:DC:RELay <0|OFF|1|ON>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Default value

ON

Example

OUTP:SDS:DC:REL 1

Returns

0|1

Query syntax

OUTPut:SDS:SENSe:RELay? [(@chanlist)]

INPut:SDS:SURGe:SUPPress [(@chanlist)]

This command is used to execute the prevent the surge function of SDS.

Syntax

OUTPut:SDS:SURGe:SUPPress [(@chanlist)]

Arguments

None

Default value

None

Example

OUTP:SDS:SURG:SUPP

Returns

0|1

0: indicates this function execute fail. 1: indicates this function execute success.

Query syntax

OUTPut:SDS:SURGe:SUPPress? [(@chanlist)]

INPut:SDS:MODE <0|OFF|1|ON>[,(@chanlist)]

这条命令用来在启用模块的时候，选择使用防打火自动模式还是手动模式。

Syntax

INPut:SDS:MODE <0|OFF|1|ON>[,(@chanlist)]

Arguments

无

Default value

无

Example

INP:SDS:MODE 0

Returns

0|1

查询命令

INPut:SDS:MODE? [(@chanlist)]

INPut:PROTection:WDOG[:STATe] <0|OFF|1|ON>[,(@chanlist)]

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

Syntax

INPut:PROTection:WDOG[:STATe] <0|OFF|1|ON>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Default value

0

Example

INP:PROT:WDOG 0

Query syntax

INPut:PROTection:WDOG[:STATe]? [(@chanlist)]

Returns

0|1

INPut:PROTection:WDOG:DELay <NRf+>[,(@chanlist)]

This command sets the time of the communication watchdog.

Syntax

INPut:PROTection:WDOG:DELay <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

2S

Example

INP:PROT:WDOG:DEL 3

Query syntax

INPut:PROTection:WDOG:DELay? [(@chanlist)]

Returns

<NRf+>

Chapter9 Trigger Commands

TRIGger[:IMMediate] [(@chanlist)]

This command generates a trigger signal when the trigger source is BUS.

Syntax

TRIGger[:IMMediate] [(@chanlist)]

Arguments

None

Example

TRIG

Query syntax

None

Returns

None

TRIGger:LIST:SOURce <KEYPad|BUS>[,(@chanlist)]

This command sets the trigger source.

Syntax

TRIGger:LIST:SOURce <KEYPad|BUS>[,(@chanlist)]

Arguments

<KEYPad|BUS>

Default value

KEYPad

Example

TRIG:LIST:SOUR KEYPad

Returns

KEYPad |BUS

INITiate[:IMMediate]:LIST [(@chanlist)]

this command is used to launch the LIST file. When the List finishes running, the interface displays END, and this command can be sent to initialize the List again to wait for triggering running state.

Syntax

INITiate[:IMMediate]:LIST [(@chanlist)]

Arguments

None

Example

INIT:LIST

Returns

None

ABORt:LIST [(@chanlist)]

This command is used to reset the list program to idle state.

Syntax

ABORt:LIST [(@chanlist)]

Arguments

None

Example

ABOR:LIST

Returns

None

Chapter10 Sense Commands

SENSe[:REMote][:STATe] <bool>[,(@chanlist)]

This command enables or disables the sense function.

Syntax

SENSe[:REMote][:STATe] <bool>[,(@chanlist)]

Arguments

0|OFF|1|ON

Default value

OFF

Example

SENS ON

Query syntax

SENSe[:REMote][:STATe]? [(@chanlist)]

Returns

0|OFF|1|ON

SENSe:FILTer:LEVel <SLOW|MEDium|FAST>[,(@chanlist)]

This command sets the sense filter level.

Syntax

SENSe:FILTer:LEVel <SLOW|MEDium|FAST>[,(@chanlist)]

Arguments

<FAST|MEDium|SLOW>

Default value

SLOW

Example

SENS:FILT:LEV MED

Query syntax

SENSe:FILTer:LEVel? [(@chanlist)]

Returns

FAST|MEDium|SLOW

SENSe:AHOur:RESet [(@chanlist)]

This command clears the Ah information.

Syntax

SENSe:AHOur:RESet [(@chanlist)]

Arguments

None

Default value

None

Example

SENS:AHO:RES

Returns

None

SENSe:AHOur:CLEar [(@chanlist)]

This command clears the Ah information. As same as sense reset command. Just for compatibility.

Syntax

SENSe:AHOur:CLEar [(@chanlist)]

Arguments

None

Default value

None

Example

SENS:AHO:CLE

Returns

None

SENSe:WHOur:RESet [(@chanlist)]

This command clears the Wh information.

Syntax

SENSe:WHOur:RESet [(@chanlist)]

Arguments

None

Default value

None

Example

SENS:WHO:RES

Returns

None

Chapter11 Load Commands

SYSTem:FUNCtion <SOURce|LOAD>[,(@chanlist)]

This command is used to switch the source mode and load mode. Please switch the instrument to load mode before sending load commands.

Syntax

SYSTem:FUNCtion <SOURce|LOAD>[,(@chanlist)]

Arguments

<SOURce|LOAD>

Default value

SOURce

Example

SYST:FUNC SOUR

Query syntax

SYSTem:FUNCtion? [(@chanlist)]

Returns

<SOURce|LOAD>

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

This command sets the current value of the electronic load.

Syntax

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MINimum

Example

CURR 2.0

Query syntax

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?
[MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <NRf+>[,(@chanlist)]

This command sets the input current value when the electronic load receives a trigger signal.

Syntax

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MINimum

Example

CURR:TRIG 2.0

Query syntax

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]?
[MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]CURRent[:OVER]:PROTection[:LEVel] <NRf+>[,(@chanlist)]

This command sets the over-current limit of the electronic load.

Syntax

[SOURce:]CURRent[:OVER]:PROTection[:LEVel] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MAXimum

Example

CURR:PROT 3.0

Query syntax

[SOURce:]CURRent[:OVER]:PROTection[:LEVel]?
[MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]CURRent[:OVER]:PROTection:DELay <NRf+>[,(@chanlist)]

This command sets the over-current delay time of the electronic load.

Syntax

[SOURce:]CURRent[:OVER]:PROTection:DELay <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MINimum

Example

CURR:PROT:DEL MAX

Query syntax

[SOURce:]CURRent[:OVER]:PROTection:DELay?
[MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]CURRent[:OVER]:PROTection:STATe <bool>[,(@chanlist)]

This command enables or disables the over-current function.

Syntax

[SOURce:]CURRent[:OVER]:PROTection:STATe <bool>[,(@chanlist)]

Arguments

0|OFF|1|ON

Default value

0

Example

CURR:PROT:STAT ON

Query syntax

[SOURce:]CURRent[:OVER]:PROTection:STATe? [(@chanlist)]

Returns

0|1

[SOURce:]CURRent:SLEW[:BOTH] <NRf+>[,(@chanlist)]

This command sets the current rising and falling slew rate.

Syntax

[SOURce:]CURRent:SLEW[:BOTH] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MAXimum

Example

CURR:SLEW 3.0

Query syntax

[SOURce:]CURRent:SLEW[:BOTH]? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]CURRent:SLEW:NEGative <NRf+>[,(@chanlist)]

This command sets the current falling slew rate.

Syntax

[SOURce:]CURRent:SLEW:NEGative <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MAXimum

Example

CURR:SLEW:NEG 2.0

Query syntax

[SOURce:]CURRent:SLEW:NEGative? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]CURRent:SLEW:POSitive <NRf+>[,(@chanlist)]

This command sets the current rising slew rate.

Syntax

[SOURce:]CURRent:SLEW:POSitive <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MAXimum

Example

CURR:SLEW:POS 4.0

Query syntax

[SOURce:]CURRent:SLEW:POSitive? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

This command sets the resistance value under CR mode.

Syntax

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MAXimum

Example

RES MAX

Query syntax

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude]? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude] <NRf+>[,(@chanlist)]

This command sets the input resistance value when the electronic load receives a trigger signal.

Syntax

[SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MAXimum

Example

RES:TRIG MAX

Query syntax

[SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude]? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

This command sets the input voltage value under CV mode.

Syntax

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MINimum

Example

VOLT 10.0

Query syntax

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <NRf+>[,(@chanlist)]

This command sets the input voltage value when the electronic load receives a trigger signal.

Syntax

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MINimum

Example

VOLT:TRIG 5.0

Query syntax

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]?
[MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]VOLTage:UNDer:PROTection[:LEVel] <NRf+>[,(@chanlist)]

This command sets the under-voltage limit value.

Syntax

[SOURce:]VOLTage:UNDer:PROTection[:LEVel] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MINimum

Example

VOLT:PROT:UND MIN

Query syntax

[SOURce:]VOLTage:UNDer:PROTection[:LEVel]?
[MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]VOLTage:UNDer:PROTection:DELay <NRf+>[,(@chanlist)]

This command sets the under-voltage delay time.

Syntax

[SOURce:]VOLTage:UNDer:PROTection:DELay <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MAXimum

Example

VOLT:PROT:UND:DEL MIN

Query syntax

[SOURce:]VOLTage:UNDer:PROTection:DELay? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]VOLTage:UNDer:PROTection:STATe <bool>[,(@chanlist)]

This command enables or disables the under-voltage function.

Syntax

[SOURce:]VOLTage:UNDer:PROTection:STATe <bool>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Default value

OFF

Example

VOLT:PROT:UND:STAT OFF

Query syntax

[SOURce:]VOLTage:UNDer:PROTection:STATe? [(@chanlist)]

Returns

0|1

[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

This command sets the input power value under CP mode.

Syntax

[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MAXimum

Example

POW MAX

Query syntax

[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]?
[MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]POWer[:LEVel]:TRIGgered[:AMPLitude] <NRf+>[,(@chanlist)]

This command sets the input power value when the electronic load receives a trigger signal.

Syntax

[SOURce:]POWer[:LEVel]:TRIGgered[:AMPLitude] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MAXimum

Example

POW:TRIG MAX

Query syntax

[SOURce:]POWer[:LEVel]:TRIGgered[:AMPLitude]?
[MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]POWer:PROTection[:LEVel] <NRf+>[,(@chanlist)]

This command sets the over-power limit.

Syntax

[SOURce:]POWer:PROTection[:LEVel] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MAXimum

Example

POW:PROT MAX

Query syntax

[SOURce:]POWer:PROTection[:LEVel]? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]POWer:PROTection:DELay <NRf+>[,(@chanlist)]

This command sets the over-power delay time.

Syntax

[SOURce:]POWer:PROTection:DELay <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MINimum

Example

POW:PROT:DEL MAX

Query syntax

[SOURce:]POWer:PROTection:DELay? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]POWer:PROTection:STATe <Bool>[,(@chanlist)]

This command enables or disables the over-power function.

Syntax

[SOURce:]POWer:PROTection:STATe <Bool>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Default value

OFF

Example

POW:PROT:STAT ON

Query syntax

[SOURce:]POWer:PROTection:STATe? [(@chanlist)]

Returns

0|1

[SOURce:]FUNCtion:MODE <FIXed|LIST|BATTery>[,(@chanlist)]

This command sets the operating mode.

Syntax

[SOURce:]FUNCtion:MODE <FIXed|LIST|BATTery>[,(@chanlist)]

Arguments

FIXed|LIST|BATTery

Default value

FIXed

Example

FUNC:MODE FIX

Query syntax

[SOURce:]FUNCtion:MODE? [(@chanlist)]

Returns

FIXed|LIST|BATTery

[SOURce:]FUNCtion <VOLTage|CURRent|POWer|RESistance|CV+CC|CV+CR|CC+CR|CV+CC+CP+CR|BSIM>[,(@chanlist)]

This command sets the run mode of electronic load.

Syntax

[SOURce:]FUNCtion <VOLTage|CURRent|POWer|RESistance|CV+CC|CV+CR|CC+CR|CV+CC+CP+CR|BSIM>[,(@chanlist)]

Arguments

VOLTage|CURRent|POWer|RESistance|CV+CC|CV+CR|CC+CR|CV+CC+CP+CR|BSIM

Default value

CURRent

Example

FUNC VOLT

Query syntax

[SOURce:]FUNCtion? [(@chanlist)]

Returns

VOLTage|CURRent|POWer|RESistance|CV+CC|CV+CR|CC+CR|CV+CC+CP+CR|BSIM

[SOURce:]VOLTage:LATCh[:STATe] <bool>[,(@chanlist)]

This command sets the VON mode of electronic load to latch status.

Syntax

[SOURce:]VOLTage:LATCh[:STATe] <bool>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Default value

OFF

Example

VON:LATC OFF

Query syntax

[SOURce:]VOLTage:LATCh[:STATe]? [(@chanlist)]

Returns

0|1

[SOURce:]VOLTage:LIVing:HYSTeresis <NRf+>[,(@chanlist)]

This command is used to set the hysteresis voltage for the VON load in LIVing mode.

Syntax

[SOURce:]VOLTage:LIVing:HYSTeresis <NRf+>[,(@chanlist)]

Arguments

NRf+

Default value

0.5

Example

VOLT:LIV:HYST MAX

查询命令

[SOURce:]VOLTage:LIVing:HYSTeresis? [(@chanlist)]

Returns

NR3

[SOURce:]VOLTage[:LEVel]:ON <NRf+>[,(@chanlist)]

This command sets the VON level.

Syntax

[SOURce:]VOLTage[:LEVel]:ON <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MAXimum

Example

VOLT:ON 3.0

Query syntax

[SOURce:]VOLTage[:LEVel]:ON? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

[SOURce:]UUT:TEMPerature:PROTection:STATe <bool>[,(@chanlist)]

This command sets the UUT over temperature protection status.

Syntax

[SOURce:]UUT:TEMPerature:PROTection:STATe <bool>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Default value

0

Example

UUT:TEMP:PROT:STAT 1

Query syntax

[SOURce:]UUT:TEMPerature:PROTection:STATe? [(@chanlist)]

Returns

0|1

[SOURce:]UUT:TEMPerature:PROTection[:LEVel] <NRf+>[,(@chanlist)]

This command sets the UUT over-temperature limit.

Syntax

[SOURce:]UUT:TEMPerature:PROTection[:LEVel] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum>|MINimum|MAXimum

Default value

MAXimum

Example

UUT:TEMP:PROT MAX

Query syntax

[SOURce:]UUT:TEMPerature:PROTection[:LEVel]? [MINimum|MAXimum][,(@chanlist)]

Returns

NRf+

[SOURce:]LOOP:SPEed <HIGH|LOW>[,(@chanlist)]

This command is used to set the voltage loop speed under load.

Syntax

[SOURce:]LOOP:SPEed <HIGH|LOW>[,(@chanlist)]

Arguments

<HIGH|LOW>

Default value

HIGH

Example

LOOP:SPE HIGH

查询命令

[SOURce:]LOOP:SPEed? [(@chanlist)]

Returns

<HIGH|LOW>

Chapter12 Trace Commands

TRACe:CLEar [(@chanlist)]

This command is used to clear reading cache. If cache is not cleared, subsequent saving will overwrite the previous data.

Syntax

TRACe:CLEar [(@chanlist)]

Arguments

None

Example

TRAC:CLE

Query syntax

None

Returns

None

TRACe:POINts <NRf+>[,(@chanlist)]

This command is used to specify cache size.

Syntax

TRACe:POINts <NRf+>[,(@chanlist)]

Arguments

<2-1000>|MINimum|MAXimum

Default value

1000

Example

TRAC:POINt MAX

Query syntax

TRACe:POINts? [(@chanlist)]

Returns

NR1

TRACe:FEED:CONTrol <NEVer|NEXT|ALWays>[,(@chanlist)]

This command is used to select cache control. Select NEVer, Save to Cache is disabled; select NEXT, save process starts and will stop when the cache is filled in. Select ALWays. After cache is filled in, circulate the cache.

Syntax

TRACe:FEED:CONTrol <NEVer|NEXT|ALWays>[,(@chanlist)]

Arguments

NEVer|NEXT|ALWays

Default value

NEVer

Example

TRAC:FEED:CONT NEXT

Query syntax

TRACe:FEED:CONTrol? [(@chanlist)]

Returns

NEVer|NEXT|ALWays

TRACe:FEED[:SELected] <BOTH|VOLTage|CURRent>[,(@chanlist)]

This command selects the reading source saved in the cache. If VOLTage is selected, the voltage reading is saved in cache; if CURRent is selected, the current reading is saved in cache. If BOTH is selected, both voltage and current are saved in cache when saving is executed.

Syntax

TRACe:FEED[:SELected] <BOTH|VOLTage|CURRent>[,(@chanlist)]

Arguments

BOTH|VOLTage|CURRent

Default value

VOLTage

Example

TRAC:FEED BOTH

Query syntax

TRACe:FEED? [(@chanlist)]

Returns

BOTH|VOLTage|CURRent

TRACe:DELay <NRf+>[,(@chanlist)]

This command selects cache delay time.

Syntax

TRACe:DELay <NRf+>[,(@chanlist)]

Arguments

0 to 3600s|MINimum|MAXimum

Default value

0.000S

Example

TRAC:DEL MINimum

Query syntax

TRACe:DELay?

Returns

NR1

TRACe:TIMer <NRf+>[,(@chanlist)]

This command selects cache time interval.

Syntax

TRACe:TIMer <NRf+>[,(@chanlist)]

Arguments

<0.001-3600.0>
MINimum-MAXimum|MINimum|MAXimum

Default value

0.001S

Example

TRAC:TIM <NRf+>

Query syntax

TRACe:TIMer? [(@chanlist)]

Returns

NR3

TRACe:POINts:ACTual? [(@chanlist)]

This command selects number of actual readings in the cache.

Syntax

TRACe:POINts:ACTual? [(@chanlist)]

Arguments

None

Example

TRAC:POIN:ACT?

Returns

NR1

TRACe:CLEar:AUTO[:STATe] <bool>[,(@chanlist)]

This command selects whether clear the readings in the cache automatically .

Syntax

TRACe:CLEar:AUTO[:STATe] <bool>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Example

TRAC:CLE:AUTO ON

Query syntax

TRACe:CLEar:AUTO[:STATe]? [(@chanlist)]

Returns

0|1

TRACe:DATA?

This command reads all values saved in the cache.

Note

Before sending the query command TRACe:DATA?, the command TRIGger[:IMMediate] must be sent to the instrument to trigger the instrument into data storage status. And the argument of the command TRACe:FEED:CONTrol <NEXT|ALWays|NEVer> must be set to NEXT or ALWays, otherwise the system will prompt an error.

Syntax

TRACe:DATA?

Arguments

None

Example

TRACe:DATA?

Returns

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

TRACe:FILTer[:STATe] [(@chanlist)]

This command enables or disables the filter.

Syntax

TRACe:FILTer[:STATe] [(@chanlist)]

Arguments

<0|1|OFF|ON>

Default value

OFF

Example

TRAC:FILT 0

Returns

0|1

Chapter13 List Commands

LIST:STEP:COUNt <NR1>|MINimum|MAXimum[,(@chanlist)]

This command sets the total steps of the list program.

Syntax

LIST:STEP:COUNt <NR1>|MINimum|MAXimum[,(@chanlist)]

Arguments

<1-100>

Example

LIST:STEP COUN 1

Query syntax

LIST:STEP:COUNt?

Returns

NR1

LIST[:STEP]:VOLTage <NR1>|MINimum|MAXimum,<NRf+>[,(@chanlist)]

This command sets the voltage value of the nth step in the list program.

Syntax

LIST[:STEP]:VOLTage <NR1>|MINimum|MAXimum,<NRf+>[,(@chanlist)]

Arguments

<1-100>,<MINimum-MAXimum|MINimum|MAXimum>

Example

LIST:VOLT 1,100.00

Query syntax

LIST:VOLTage?

Returns

NR3

LIST[:STEP]:CURRent <NR1>|MINimum|MAXimum,<NRf+>[,(@chanlist)]

This command sets the current value of the nth step in the list program.

Syntax

LIST[:STEP]:CURRent <NR1>,<NRf+>

Arguments

<1-100>,<MINimum-MAXimum|MINimum|MAXimum>

Example

LIST:CURRent 1,3.500

Query syntax

LIST[:STEP]:CURRent?

Returns

NR3

LIST[:STEP]:POWer <NR1>|MINimum|MAXimum,<NRf+>[,(@chanlist)]

设置 LIST 第 n 步的功率。（载状态下可用）

Syntax

LIST[:STEP]:POWer <NR1>|MINimum|MAXimum,<NRf+>[,(@chanlist)]

Arguments

<1-100>,<MINimum-MAXimum|MINimum|MAXimum>

Default value

不适用

Example

LIST:POW 1, 5

查询语法

LIST[:STEP]:POWer? <NR1>|MINimum|MAXimum[,(@chanlist)]

Returns

NR3

LIST[:STEP]:RESistance <NR1>|MINimum|MAXimum,<NRf+>[,(@chanlist)]

设置 LIST 第 n 步的电阻。

Syntax

LIST[:STEP]:RESistance <NR1>|MINimum|MAXimum,<NRf+>[,(@chanlist)]

Arguments

<1-100>,<MINimum-MAXimum|MINimum|MAXimum>

Default value

不适用

Example

LIST:RES 1, 15

查询语法

LIST[:STEP]:RESistance? <NR1>|MINimum|MAXimum[,(@chanlist)]

Returns

NR3

LIST[:STEP]:SLEW <NR1>,<NRf+>[,(@chanlist)]

This command sets the slew rate value of the nth step in the list program.

Syntax

LIST[:STEP]:SLEW <NR1>,<NRf+>[,(@chanlist)]

Arguments

<1-100>,<MINimum-MAXimum|MINimum|MAXimum>

Example

LIST[:STEP]:SLEW 1,1.000

Query syntax

LIST[:STEP]:SLEW? <NR1>[,(@chanlist)]

Returns

NR3

LIST[:STEP]:WIDTh <NR1>,<NRf+>[,(@chanlist)]

This command sets the width value of the nth step in the list program.

Syntax

LIST[:STEP]:WIDTh <NR1>,<NRf+>[,(@chanlist)]

Arguments

<1-100>,<MINimum-MAXimum|MINimum|MAXimum>

Example

LIST[:STEP]:WIDT 1,1.000

Query syntax

LIST[:STEP]:WIDTh? <NR1>[,(@chanlist)]

Returns

NR3

LIST:REPeat <NRf+>[,(@chanlist)]

This command sets the number of list repetitions.

Syntax

LIST:REPeat <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

LIST:REP 3

Query syntax

LIST:REPeat? [(@chanlist)]

Returns

NR3

LIST:FUNCtion <VOLTage|CURRent>[,(@chanlist)]

This command sets the working mode of list program.

Syntax

LIST:FUNCtion <VOLTage|CURRent>[,(@chanlist)]

Arguments

<VOLTage|CURRent>

Example

LIST:FUNC VOLT

Query syntax

LIST:FUNCtion? [(@chanlist)]

Returns

VOLTage|CURRent

LIST:VOLTage:LIMit[:HIGH] <NRf+>[,(@chanlist)]

This command sets the voltage upper limit of list program.

Syntax

LIST:VOLTage:LIMit[:HIGH] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

LIST:VOLT:LIM 80

Query syntax

LIST:VOLTage:LIMit[:HIGH]? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

LIST:VOLTage:LIMit:LOW <NRf+>[,(@chanlist)]

This command sets the voltage lower limit of list program.

Syntax

LIST:VOLTage:LIMit:LOW <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

LIST:VOLT:LIM:LOW 20

Query syntax

LIST:VOLTage:LIMit:LOW? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

LIST:CURRent:LIMit[:POSitive] <NRf+>[,(@chanlist)]

This command sets the current positive limit of list program.

Syntax

LIST:CURRent:LIMit[:POSitive] <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

LIST:CURR:LIM 20

Query syntax

LIST:CURRent:LIMit[:POSitive]? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

LIST:CURRent:LIMit:NEGative <NRf+>[,(@chanlist)]

This command sets the current negative limit of list program.

Syntax

LIST:CURRent:LIMit:NEGative <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

LIST:CURR:LIM:NEG 20

Query syntax

LIST:CURRent:LIMit:NEGative? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

LIST:SAVE <NR1>[,(@chanlist)]

This command saves the present list program into the specified memory.

Syntax

LIST:SAVE <NR1>[,(@chanlist)]

Arguments

<1-10>

Example

LIST:SAV 1

Query syntax

LIST:SAVE? [(@chanlist)]

Returns

NR3

LIST:RECall <NR1>[,(@chanlist)]

This command recalls the list program you saved in the specified memory location.

Syntax

LIST:RECall <NR1>[,(@chanlist)]

Arguments

<1-10>

Example

LIST:REC 1

Query syntax

LIST:RECall? [(@chanlist)]

Returns

NR3

LIST[:STATe] <bool>[,(@chanlist)]

This command enables or disables the list function.

Syntax

LIST[:STATe] <bool>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Default value

0

Example

LIST ON

Query syntax

LIST[:STATe]? [(@chanlist)]

Returns

0|1

LIST:TERMinate <terminate>[,(@chanlist)]

This command sets the end state of the list program.

Syntax

LIST:TERMinate <terminate>[,(@chanlist)]

Arguments

NORMal|LAST

Example

LIST:TERM NORM

Query syntax

LIST:TERMinate? [(@chanlist)]

Returns

NORMal|LAST

LIST:PAUSe[:STATe] <BOOLEAN>[,(@chanlist)]

This command sets the pause state of the list program.

Syntax

LIST:PAUSe[:STATe] <BOOLEAN>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Example

LIST:PAUS 1

Query syntax

LIST:PAUSe[:STATe]? [(@chanlist)]

Returns

0|1

LIST:RESet [(@chanlist)]

This command resets the running state of the list program to waiting trigger.

Syntax

[SOURce:]LIST:RESet [(@chanlist)]

Arguments

None

Example

LIST:RES

Query syntax

None

Returns

None

LIST:RUN:STEP? [(@chanlist)]

This command queries the present step number of the running list program.

Syntax

[SOURce:]LIST:RUN:STEP? [(@chanlist)]

Arguments

None

Example

LIST:RUN:STEP? [(@chanlist)]

Returns

NR1

LIST:RUN:REPeat? [(@chanlist)]

This command queries the present repetitions of the running list program.

Syntax

[SOURce:]LIST:RUN:REPeat? [(@chanlist)]

Arguments

None

Example

LIST:RUN:REP?

Returns

NR1

Chapter14 Battery Commands

BATTery:DISCharge:VOLTage <NRf+>[,(@chanlist)]

This command is used to set the battery discharge voltage value.

Syntax

BATTery:DISCharge:VOLTage <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:DISC:VOLT 3.0

Query syntax

BATTery:DISCharge:VOLTage? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

BATTery:DISCharge:CURRent <NRf+>[,(@chanlist)]

This command is used to set the battery discharge current value.

Syntax

BATTery:DISCharge:CURRent <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:DISC:CURR 3.0

Query syntax

BATTery:DISCharge:CURRent? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

BATTery:STOP:VOLTage <NRf+>[,(@chanlist)]

This command is used to set the voltage value for the battery test cutoff.

Syntax

BATTery:STOP:VOLTage <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:SHUT:VOLT 4.0

Query syntax

BATTery:STOP:VOLTage? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

BATTery:STOP:CURRent <NRf+>[,(@chanlist)]

This command is used to set the current value of the battery test cutoff.

Syntax

BATTery:STOP:CURRent <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:SHUT:CURR 3.0

Query syntax

BATTery:STOP:CURRent? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

BATTery:STOP:CAPacity <NRf+>[,(@chanlist)]

This command is used to set the capacitance value of the battery test cutoff.

Syntax

BATTery:STOP:CAPacity <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:SHUT:CAP 3.0

Query syntax

BATTery:STOP:CAPacity? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

BATTery:STOP:TIME <NRf+>[,(@chanlist)]

This command is used to set the battery test cutoff time.

Syntax

BATTery:STOP:TIME <NRf+>[,(@chanlist)]

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:SHUT:TIME 3.0

Query syntax

BATTery:STOP:TIME? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

BATTery[:STATe] <BOOL>[,(@chanlist)]

This command enables or disables the battery function.

Syntax

BATTery[:STATe] <BOOL>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Example

BATT 1

Query syntax

BATTery[:STATe]? [(@chanlist)]

Returns

0|1

BATTery:SAVE <BANK>[,(@chanlist)]

This command saves the present battery test file into the specified memory.

Syntax

BATTery:SAVE <BANK>[,(@chanlist)]

Arguments

<1-10>

Example

BATT:SAV 1

Query syntax

None

Returns

<NR1>

BATTery:RECall <BANK>[,(@chanlist)]

This command recalls the battery test file you saved in the specified memory location.

Syntax

BATTery:SAVE <BANK>[,(@chanlist)]

Arguments

<1-10>

Example

BATT:REC 1

Query syntax

None

Returns

<NR1>

BATTery:RESet [(@chanlist)]

This command resets the running state of the list program to initial state.

Syntax

[SOURce:]BATTery:RESet [(@chanlist)]

Arguments

None

Example

BATT:RES

Query syntax

None

Returns

None

BATTery:TIME? [(@chanlist)]

该命令用来查询本次电池充/放电的时间。

Syntax

BATTery:TIME? [(@chanlist)]

Arguments

无

Default value

不适用

Example

BATT:TIME?

Returns

无

Chapter15 Parallel&Link Commands

PARallel:ROLE <SINGle|SLAVe|MASTer>[,(@chanlist)]

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

Syntax

PARallel:ROLE <SINGle|SLAVe|MASTer>[,(@chanlist)]

Arguments

SINGle|SLAVe|MASTer

Example

PAR:ROLE SLAV

Query syntax

PARallel:ROLE? [(@chanlist)]

Returns

SINGle|SLAVe|MASTer

PARallel:GROup <group>[,(@chanlist)]

This command specifies the parallel group.

Syntax

PARallel:GROup <group>[,(@chanlist)]

Arguments

<A-H>

Example

PAR:GRO 1

Query syntax

PARallel:GROup? [(@chanlist)]

Returns

NR1

PARallel:NUMBer <NR1>[,(@chanlist)]

This command sets the total instrument number in the parallel operation.

Syntax

PARallel:NUMBer <NR1>[,(@chanlist)]

Arguments

<1-16>

Example

PAR:NUMB 3

Query syntax

PARallel:NUMBer? [(@chanlist)]

Returns

NR1

LINK:MODE <OUTPut|TRACk>[,(@chanlist)]

This command sets the link mode.

Syntax

LINK:MODE <OUTPut|TRACk|DUPLicate>[,(@chanlist)]

Arguments

OUTPut|TRACk|DUPLicate

Example

LINK:MODE OUTP

Query syntax

LINK:MODE? [(@chanlist)]

Returns

OUTPut|TRACk|DUPLicate

LINK[:STATe] <bool>[,(@chanlist)]

This command enables or disables the link function.

Syntax

LINK[:STATe] <bool>[,(@chanlist)]

Arguments

<0|1|OFF|ON>

Example

LINK ON

Query syntax

LINK[:STATe]? [(@chanlist)]

Returns

0|1

LINK:REFerence <NRf+>[,(@chanlist)]

This command sets reference proportion of the link mode.

Syntax

LINK:REFerence <NRf+>[,(@chanlist)]

Arguments

<0.01-100.00>

Example

LINK:REF 3

Query syntax

LINK:REFerence? [MINimum|MAXimum][,(@chanlist)]

Returns

NR3

Chapter16 Common Commands

*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

Default value

Not applicable

Example

*CLS

Query command

None

Returns

Not applicable

*ESE

This command sets or queries 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 <NRf>

Arguments

0 to 255

Default value

0

Example

*ESE 16

Query command

*ESE?

Returns

NR1

See also

*ESR?
*STB?

*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.

Syntax

*ESR?

Arguments

None

Default value

Not applicable

Example

*ESR?

Returns

NR1

See also

*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. Example: ITECH Ltd., IT3300, 60234567890123456, 1.01-1.02-1.03.

Syntax

*IDN?

Arguments

None

Default value

Not applicable

Example

- > *IDN?
< - ITECH Ltd.,IT3300,60234567890123456,1.01-1.02-1.03

Note

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

Returns

AARD

*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

Default value

Not applicable

Example

*OPC

Query command

*OPC?

Returns

NR1

*PSC

This command specifies whether the Service Request Enable Register (SRER) and the Event State Enable Register (ESER) are cleared when the instrument is powered on. The query form of this command gets the state of the power-on status clear function.

Syntax

*PSC <Boolean>

Arguments

0|OFF|1|ON

Default value

0

Example

*PSC 0

Query command

*PSC?

Returns

0|1

*RCL

This command recalls the setups you saved in the specified memory location.

Syntax

*RCL <NRf>

Arguments

<1-10>

Default value

Not applicable

Example

*RCL 1

Query command

None

Returns

Not applicable

*RST

This command resets the electronic load to default settings.

Syntax

*RST

Arguments

None

Default value

Not applicable

Example

*RST

Query command

None

Returns

Not applicable

*SAV

This command saves the present setting values of the electronic load into specified memory

Syntax

*SAV <NRf>

Arguments

<1-10>

Default value

Not applicable

Example

*SAV 1

Query command

None

Returns

Not applicable

*SRE

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 <NRf>

Arguments

0 to 255

Default value

Not applicable

Example

*SRE 255

Query command

*SRE?

Returns

NR1

*STB?

This command reads the data in the Status Byte Register (SBR).

Syntax

*STB?

Arguments

None

Default value

Not applicable

Example

*STB?

Returns

NR1

*TRG

This command generates a trigger signal when the trigger source is BUS.

Syntax

*TRG

Arguments

None

Default value

Not applicable

Example

*TRG

Query command

None

Returns

None

*TST?

This command initiates a self-test and reports any errors.

Query command

*TST?

Arguments

None

Default value

Not applicable

Example

*TST?

Returns

NR1,<str>

*WAI

Pauses additional command processing until all pending operations are complete.

Query command

*WAI

Arguments

None

Default value

Not applicable

Example

*WAI

Returns

None

*PSC <Bool>

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 <Bool>

Arguments

0|OFF|1|ON

Default value

OFF

Example

*PSC 0

Query syntax

*PSC?

Returns

0|1

Chapter17 Error Messages

Error List

Sending the command SYST:ERR? can read one error code and error message from the error queue.

Error code	Error message
101	"Too many numeric suffices"
110	"No input command"
114	"Invalid Numeric suffix"
116	"Invalid value"
117	"Invalid dimensions"
120	"Parameter overflowed"
130	"Wrong units for parameter"
140	"Wrong type of parameter"
150	"Wrong number of parameter"
160	"Unmatched quotation mark"
165	"Unmatched bracket"
170	"Invalid command"
180	"No entry in list"
190	"Too many dimensions"
191	"Too many char"
-200	"Execution error"
-221	"Settings conflict"
-222	"Data out of range"
-223	"Too much data"
-224	"Illegal parameter value"
-225	"Out of memory"
-230	"Data Corrupt or Stale"
-270	"Macro error"
0	"No error"
1	"Module Initialization Lost"
2	"Mainframe Initialization Lost"
3	"Module Calibration Lost"
4	"Eeprom failure"
5	"RST checksum failed"
6	"BACKUP RAM failed"
10	"RAM selftest failed"
222	"Front panel uart parity"
223	"Front panel buffer overrun"
224	"Front panel timeout"
225	"Front Crc Check error"
226	"Front Cmd Error"

401	"CAL switch prevents"
402	"CAL password is incorrect"
403	"CAL not enabled"
404	"readback cal are incorrect"
405	"programming cal are incorrect"
406	"Incorrect sequence of cal"
602	"Commond only for rs232"
603	"FETCH of data was not acquired"
604	"Measurement overrange"
800	"Sn Same Conflict"

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.