

# DC Programmable Electronic Load

## Series IT8500+ Programming Guide



Model: IT8511+/IT8511A+/IT8512+/IT8512A+/  
IT8512B+/IT8512C+/IT8512H+/IT8513A+/  
IT8513B+/IT8513C+/IT8514C+/IT8514B+/  
IT8516C+

Revision: V1.2

## Statement

© Itech Electronic, Co., Ltd. 2018 No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior permission and written consent from Itech Electronic, Co., Ltd. as governed by international copyright laws.

### Manual Article No.

IT8500+-402158

### Revision

Revision 1, published on

May 4th, 2018

Itech Electronic, Co., Ltd.

### Trademark Statement

Pentium is a registered trademark of Intel Corporation in the United States.

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

### Guarantee

Materials in the document are provided *talis qualis* and may be changed in future revisions without prior notice. In addition, within the maximum allowable extent of laws, ITECH is not committed to any explicit or implied guarantee for this manual and all information therein, including but not limited to the implied guarantee on marketability and availability for some special purposes. ITECH shall not be responsible for any error or incidental or indirect losses caused by the provision, use or application of this documents and information therein. If some guarantee clauses in other written agreements between ITECH and users are not consistent with clauses herein, those clauses in other written agreements shall prevail.

### Technology license

Hardware and/or software in this document cannot be provided without a license and can only be used or copied according to the license.

### Restricted permission statement

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 Statement

#### CAUTION

"Caution" signs indicate danger. It is required to pay attention to the contents of these signs during implementation of operations.

The damage to the product or loss of important data may be caused in case of improper operation steps or failure to follow operation steps. Do not continue to implement any improper operation indicated in "Caution" signs when the specified conditions are not fully understood or these conditions are not satisfied.

#### WARNING

"Warning" indicates danger. It is required to pay attention to the contents of these signs during implementation of operation steps. Personal casualties may be caused in case of improper operation steps or failure to follow these operation steps. Do not continue to implement any improper operation indicated in "Warning" signs when the specified conditions are not fully understood or these conditions are not satisfied.



#### NOTE

"Instructions" indicates operation instructions. It is required to refer to the contents of these signs during operation steps. These signs are used for providing tips or supplementary information for operators.

## Certification and Quality Assurance

IT8500+ series electronic load completely reaches nominal technical indicators in the manual.

## Warranty Service

ITECH Company will provide one-year warranty services for the product materials and manufacturing (excluding the following limitations).
















- When warranty service or repair is needed, please send the product to the service unit specified by ITECH Company.
- When the product is sent to ITECH Company for warranty service, the customer must pay the one-way freight to the maintenance department of ITECH, and ITECH will be responsible for return freight.
- If the product is sent to ITECH for warranty service from other countries, the customer will be responsible for all the freight, duties and other taxes.

## Limitation of Warranty

Warranty service does not apply to the damage caused in the following circumstances:

- Damage resulting from customer-wired circuits or customer-supplied parts or accessories;
- Product which has been modified or repaired by the customer;
- Damage caused by the circuit installed by the customer or damage caused by operation of the product in non-specified environment;
- The product model or serial number is altered, deleted, removed or made illegible by customer;
- Damage caused by accidents, including but not limited to lightning, water, fire, abuse or negligence.

## Safety Signs

	DC power		ON (with the power switched on)
	AC power		OFF (with the power supply switched off)
	Both DC and AC power supply		Power supply switching-on status
	Protective grounding terminal		Power supply switching-off status
	Grounding terminal		Reference terminal
	Danger sign		Positive terminal
	Warning sign (refer to specific "Warning" or "Caution" information in the manual)		Negative terminal
	Ground wire connection end sign	-	-

## Safety Precautions

General safety precautions below must be followed in each phase of instrument operation. In case of failure to follow these precautions or specific warnings in other parts of the manual, violation against the safety standards related to the design, manufacture and purpose of the instrument will occur. If the user does not follow these precautions, ITECH will bear no responsibility arising there from.

### WARNING

- Series IT8500+ electronic load supports 110V/220VAC input and need to switch the input voltage before operation.
- 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 electronic load 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 electronic load 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 IT8500+ series electronic load can only be used indoors or in low condensation areas. The following table shows general environmental

requirements for this instrument.




Environmental conditions	Requirement
Operating temperature	0°C - 40°C
Operating humidity	20% - 80% (non condensing)
Storage temperature	-20°C - 70 °C
Altitude	≤2,000m
Pollution	Grade 2 pollution
Installation category	II



Note

In order to ensure the accuracy of measurement, it is recommended to operate the instrument half an hour after start-up.

## Regulation Tag

	The CE tag shows that the product complies with the provisions of all relevant European laws (if the year is shown, it indicates that the year when the design is approved).
	This instrument complies with the WEEE directive (2002/96/EC) tag requirements. This attached product tag shows that the electrical/electronic product cannot be discarded in household waste.
	This symbol indicates that no danger will happen or toxic substances will not leak or cause damage in normal use within the specified period. The service life of the product is 10 years. The product can be used safely within the environmental protection period; otherwise, the product should be put into the recycling system.

## Waste Electrical and Electronic Equipment (WEEE) Directive



Waste electrical and electronic equipment (WEEE) directive, 2002/96/EC

The product complies with tag requirements of the WEEE directive (2002/96/EC). This tag indicates that the electronic equipment cannot be disposed of as ordinary household waste.

Product Category

According to the equipment classification in Annex I of the WEEE directive, this instrument belongs to the “Monitoring” product.

If you want to return the unnecessary instrument, please contact the nearest sales office of ITECH.

## 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 <sup>123</sup>

#### 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

## Contents

Certification and Quality Assurance .....	i
Warranty Service .....	i
Limitation of Warranty .....	i
Safety Signs .....	i
Safety Precautions .....	ii
Environmental Conditions .....	ii
Regulation Tag .....	iii
Waste Electrical and Electronic Equipment (WEEE) Directive .....	iii
Compliance Information .....	iv
<b>Chapter1 Remote Control .....</b>	<b>1</b>
1.1 Overview .....	1
1.2 SCPI Command Introduction .....	1
1.3 Command Type of SCPI .....	1
1.4 Message Type of SCPI .....	3
1.5 Response Data Type .....	5
1.6 Command Format .....	5
1.7 Data Type .....	7
1.8 Remote Interface Communication .....	8
1.8.1 Communication Modules Intruduction .....	8
1.8.2 Communication with PC .....	10
<b>Chapter2 SCPI Status Register .....</b>	<b>13</b>
<b>Chapter3 Essential Commands .....</b>	<b>15</b>
SYSTem:ERRor? .....	15
SYSTem:VERSion? .....	15
SYSTem:SENSe[:STATe] <bool> .....	15
SYSTem:SENSe[:STATe]? .....	16
SYSTem:LOCal .....	16
SYSTem:REMote .....	16
SYSTem:RWLock .....	17
<b>Chapter4 Status Commands .....</b>	<b>18</b>
STATus:QUEStionable[:EVENT]? .....	18
STATus:QUEStionable:CONDition? .....	18
STATus:QUEStionable:ENABle <NRf+> .....	18
STATus:QUEStionable:ENABle? .....	19
STATus:OPERation[:EVENT]? .....	19
STATus:OPERation:CONDition? .....	20
STATus:OPERation:ENABle <NRf+> .....	20
STATus:OPERation:ENABle? .....	21
<b>Chapter5 Input Settings .....</b>	<b>22</b>
[SOURce:]INPut <bool> .....	22
[SOURce:]INPut? .....	22
[SOURce:]INPut:SHORt <bool> .....	22
[SOURce:]INPut:SHORt? .....	23
[SOURce:]CURRent:RANGe <NRf+> .....	23
[SOURce:]CURRent:RANGe? .....	24
[SOURce:]VOLTage:RANGe <NRf+> .....	24
[SOURce:]VOLTage:RANGe? .....	25
[SOURce:]VOLTage:RANGe:AUTO[:STATe] .....	25
[SOURce:]CURRent:SLEW[:BOTH] <NRf+> .....	26
[SOURce:]CURRent:SLEW[:BOTH]? .....	26
[SOURce:]CURRent:SLEW:RISE <NRf+> .....	26
[SOURce:]CURRent:SLEW:RISE? .....	27
[SOURce:]CURRent:SLEW:FALL <NRf+> .....	27
[SOURce:]CURRent:SLEW:FALL? .....	28
[SOURce:]CURRent:PROTection[:LEVel] <NRf+> .....	28
[SOURce:]CURRent:PROTection[:LEVel]? .....	29
[SOURce:]POWer:PROTection[:LEVel] <NRf+> .....	29

[SOURce:]POWER:PROTection[:LEVel]?	30
[SOURce:]VOLTage[:LEVel:]ON <NRf+>	30
[SOURce:]Voltage[:LEVel:]ON?	30
[SOURce:]Voltage[:LEVel:]OFF <NRf+>	31
[SOURce:]Voltage[:LEVel:]OFF?	31
[SOURce:]FUNctIon <function>	32
[SOURce:]MODE <function>	32
[SOURce:]FUNctIon?	32
[SOURce:]MODE?	32
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <NRf+>	33
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?	33
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf+>	34
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]?	34
[SOURce:]POWER[:LEVel][:IMMediate][:AMPLitude] <NRf+>	35
[SOURce:]POWER[:LEVel][:IMMediate][:AMPLitude]?	35
[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude] <NRf+>	36
[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude]?	36
[SOURce:]DYNamic:HIGH[:LEVel] <NRf+>	36
[SOURce:]DYNamic:HIGH[:LEVel]?	37
[SOURce:]DYNamic:HIGH:DWELI <NRf+>	37
[SOURce:]DYNamic:HIGH:DWELI?	38
[SOURce:]DYNamic:LOW[:LEVel] <NRf+>	38
[SOURce:]DYNamic:LOW[:LEVel]?	39
[SOURce:]DYNamic:LOW:DWELI <NRf+>	39
[SOURce:]DYNamic:LOW:DWELI?	40
[SOURce:]DYNamic:SLEW <NRf+>	40
[SOURce:]DYNamic:SLEW?	41
[SOURce:]DYNamic:SLEW:RISE <NRf+>	41
[SOURce:]DYNamic:SLEW:RISE?	41
[SOURce:]DYNamic:SLEW:FALL <NRf+>	42
[SOURce:]DYNamic:SLEW:FALL?	42
[SOURce:]DYNamic:MODE <mode>	43
[SOURce:]DYNamic:MODE?	43
<b>Chapter6 Measure Commands</b>	<b>44</b>
MEASure[:SCALar]:VOLTage[:DC]?	44
MEASure[:SCALar]:VOLTage:MAXimum?	44
MEASure[:SCALar]:VOLTage:MINimum?	44
MEASure[:SCALar]:VOLTage:PTPeak?	45
MEASure[:SCALar]:CURRent[:DC]?	45
MEASure[:SCALar]:CURRent:MAXimum?	45
MEASure[:SCALar]:CURRent:MINimum?	46
MEASure[:SCALar]:CURRent:PTPeak?	46
MEASure[:SCALar]:POWer[:DC]?	46
MEAS[:SCALar]:RESistance[:DC]?	47
<b>Chapter7 OCP Testing Commands</b>	<b>48</b>
OCP[:STATe] <bool>	48
OCP[:STATe]?	48
OCP:ISart <NRf+>	48
OCP:ISart?	49
OCP:IEND <NRf+>	49
OCP:IEND?	49
OCP:STEP	50
OCP:STEP?	50
OCP:DWELI <NRf+>	50
OCP:DWEL?	51
OCP:VTRig <NRf+>	51
OCP:VTRig?	52
OCP:RESult[:OCP]?	52
OCP:RESult:PMAX?	52
<b>Chapter8 Peak Testing Commands</b>	<b>54</b>



PEAK[:STATe] <bool> .....	54
PEAK CLEAr .....	54
PEAK:VOLTage:MAXimum? .....	54
PEAK:VOLTage:MINimum? .....	54
PEAK:CURRent:MAXimum? .....	55
PEAK: CURRent:MINimum? .....	55
<b>Chapter9 Voltage Rise/Fall Time Test.....</b>	<b>56</b>
TIME:VOLTage:LOW <Voltage> .....	56
TIME:VOLTage:LOW? .....	56
TIME:VOLTage:HIGh <Voltage> .....	56
TIME:VOLTage:HIGh? .....	57
TIME:VOLTage:UP? .....	57
TIME:VOLTage:DOWN? .....	57
<b>Chapter10 Timing Test Commands.....</b>	<b>58</b>
TIMing[:STATe] .....	58
TIMing:LOAD:SETTing .....	58
TIMing:LOAD:MODE .....	59
TIMing:LOAD:VALue .....	59
TIMing:TSTart:SOURce .....	60
TIMing:TSTart:EDGE .....	60
TIMing:TSTart:LEVel .....	60
TIMing:TEND:SOURce .....	61
TIMing:TEND:EDGE .....	61
TIMing:TEND:LEVel .....	62
TIMing:RESult .....	62
<b>Chapter11 CR-LED Commands .....</b>	<b>63</b>
LED:VOLTage .....	63
LED:CURRent .....	63
LED:RCOff .....	64
<b>Chapter12 IEEE488 Commands.....</b>	<b>65</b>
*CLS - Clear Status .....	65
*ESE <NRf> - Event Enable .....	65
*ESE? .....	66
*ESR? .....	66
*IDN? .....	66
*OPC .....	67
*OPC? .....	68
*PSC .....	68
*PSC? .....	68
*RCL .....	69
*RST .....	69
*SAV .....	69
*SRE .....	70
*SRE? .....	70
*STB? .....	71
*TST? .....	71
*TRG .....	71
<b>Chapter13 Error Messages .....</b>	<b>73</b>

# Chapter1 Remote Control

## 1.1 Overview

This chapter will provide following remote configuration introductions:

- SCPI Command Introduction
- Command type
- Command format
- Data format
- Remote Operation

## 1.2 SCPI Command Introduction

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 {NORMa|CARRier}

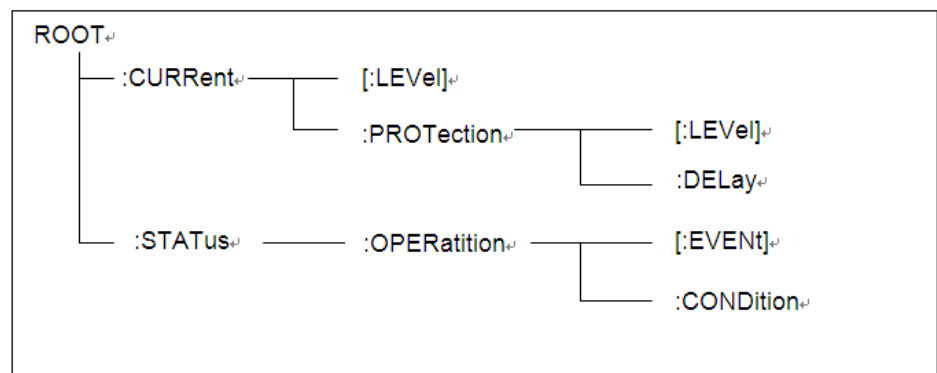
POLarity {NORMa|INVerted}

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.

## 1.3 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 electronic load functions, such as reset, status, and synchronization. All common commands consist of a three-letter mnemonic preceded by an asterisk: \*RST \*IDN? \*SRE 8.
- Subsystem commands perform specific electronic load 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 ONIT872X-3X SCPI Communication protocol 17`

## 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:

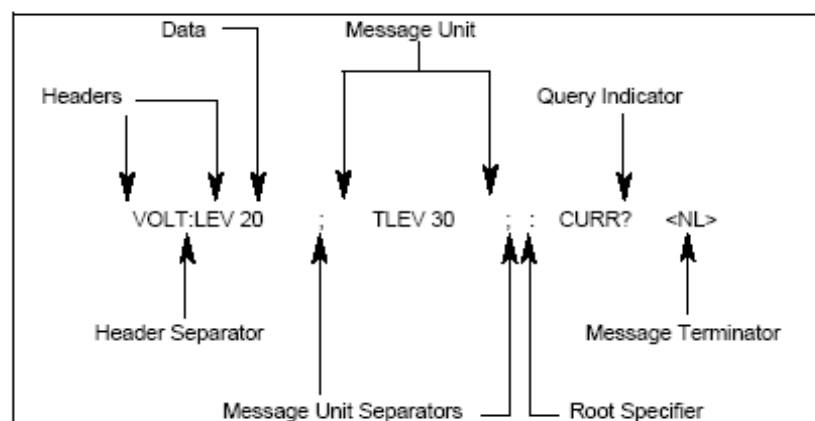
- Set up the proper number of variables for the returned data. For example, if you are reading back a measurement array, you must dimension the array according to the number of measurements that you have placed in the measurement buffer.
- Read back all the results of a query before sending another command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

## 1.4 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 electronic load. The message, which may be sent at any time, requests the electronic load to perform some action.
- response message: A response message consists of data in a specific SCPI format sent from the electronic load to the controller. The electronic load 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.

VOLTage 20<NL>

## Headers

Headers, also referred to as keywords, are instructions recognized by the electronic load. 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.

## 1.5 Response Data Type

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

<b>&lt;CRD&gt;</b>	Character Response Data. Permits the return of character strings.
<b>&lt;AARD&gt;</b>	Arbitrary ASCII Response Data. Permits the return of undelimited 7-bit ASCII. This data type has an implied message terminator.
<b>&lt;SRD&gt;</b>	String Response Data. Returns string parameters enclosed in double quotes

### 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 IT8500+ Series 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 (see the paragraph entitled, "Multiple Command Messages"), the multiple response messages for all the queries is sent to the computer when the IT8500+ Series 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 IT8500+ Series what to send to the computer.

The following two steps must always be performed to send information from the instrument to the computer:

1. Send the appropriate query command(s) in a program message.
2. Address the IT8500+ Series to talk.

**Rule 2.** The complete response message must be received by the computer before another program message can be sent to the IT8500+ Series.

## 1.6 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.

## Generic commands (\*)

Execute functions like reset, self inspection and status operation. Generic commands always start with an 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.

# 1.7 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}



- 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...'

## 1.8 Remote Interface Communication

### 1.8.1 Communication Modules Introduction

The DB9 interface connector on the rear panel of the load (IT8511+/IT8511A+/IT8512+/IT8512A+/IT8512B+/IT8512C+/IT8512H+/IT8513A+/IT8513C+) is TTL voltage level, which needs to be connected to the serial port of the PC through level conversion. The optional communication module is IT-E121/IT-E121A/IT-E122/IT-E123.

The IT8513B+/IT8514B+/IT8514C+/IT8516C+ load includes standard RS232 and USB communication interface.

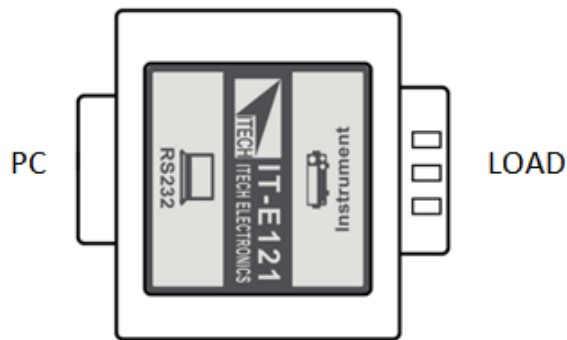
#### CAUTION

Do not connect the electronic load (IT8511+/IT8511A+/IT8512+/IT8512A+/IT8512B+/IT8512C+/IT8512H+/IT8513A+/IT8513C+) with a standard RS232 cable, which may result in damage to the instrument.

#### IT-E121 Communication Module

The DB9 interface connector on the rear panel of the DC load is TTL voltage level; you can use the communication module IT-E121 and a standard RS232 extension cable to connect the DB9 interface connector of the DC load and the RS-232 interface connector of computer for the communication.

IT-E121 communication cable

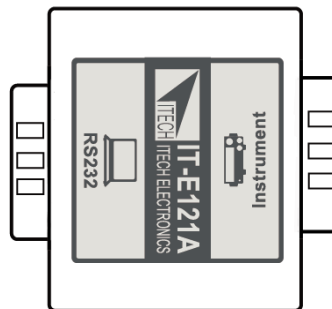


### IT-E121A Communication Module

The DB9 interface connector on the rear panel of the DC load is TTL voltage level; you can use the communication module IT-E121A and a standard RS232 extension cable to connect the DB9 interface connector of the DC load and the RS-232 interface connector of computer for the communication.

IT-E121A is derived on the basis of IT-E121, the main difference between them is that the DB9 interface connector of the RS232 changes from female to male, so that can be directly connected to the standard LAN interface.

IT-E121A communication cable



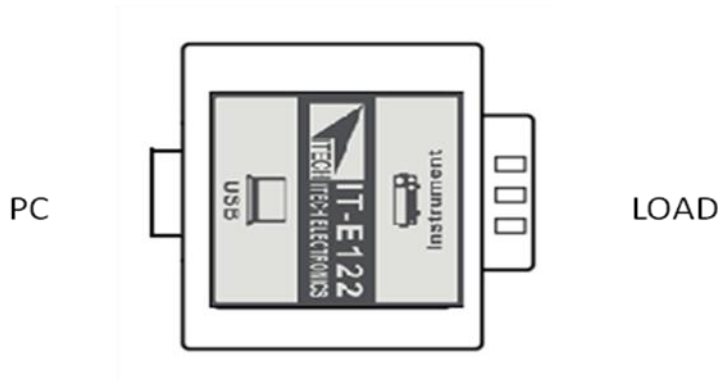
### IT-E122 Communication Module

The DB9 interface connector on the rear panel of the DC load is TTL voltage level; IT-E122 has a USB interface (male connector B) on one end, you can use IT-E122 and a standard USB extension cable (type B female connector at one end and type A at one end) to connect the DB9 interface connector of the DC load and the USB interface connector of computer for the communication.

After connecting the load and computer by USB, you need to install IT-E122 driver (download from

<http://www.itech.sh/support/Down-3.htm?model=IT-E122%E5%85%89%E9%9A%94%E7%A6%BB%E9%80%9A%E8%AE%AF%E6%8E%A5%E5%8F%A3%E9%A9%B1%E5%8A%A8> or contact ITECH agent). The device manager of PC will display 'Prolific USB-to-Serial COM Port' after installing.

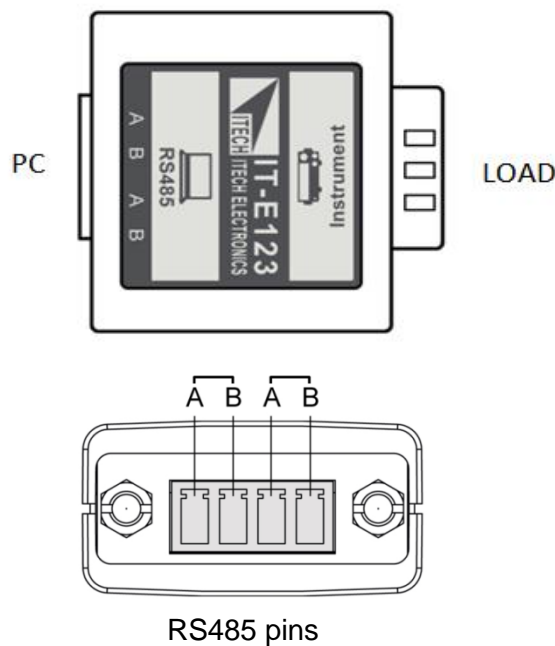
IT-E122 communication cable



### IT-E123 Communication Module

The DB9 interface connector on the rear panel of the DC load is TTL voltage level; The interface on both side port of IT-E123 are DB9 interface and RS485 interface, you can use the communication module IT-E123 and an a standard RS485-RS232 conversion cable to connect the DB9 interface connector of the DC load and the RS-232 interface connector of computer for the communication.

IT-E123 communication cable



## 1.8.2 Communication with PC

Before using the remote operation mode, please make sure that the baud rate and communication address in the DC load are the same as in the computer software, otherwise, the communication will fail, you can change the baud rate and communication address from the front panel or from computer.

### DB9 Serial Port

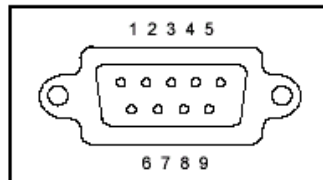
In order for the computer to communicate with the DC load, both must be set to the same RS-232 settings. These communication settings are:

1. Baud rate: 4800,9600,19200 and 38400 are selectable, default setting is

9600.

2. Data bit: 8 bit
3. Stop bit: 1
4. Parity: None
5. Address: the range is from 0 to 31, default setting is 0

Start Bit	8 Data Bits	Parity=None	Stop Bit
-----------	-------------	-------------	----------

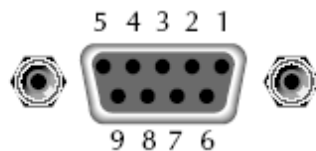


## RS-232

IT8513B+/IT8514B+/IT8514C+/IT8516C+ electronic load has a DB9 interface on rear panel. Connect E-load and computer by cable of COM ends (DB9). Composite key **[Shift] + [8]** on front board can be used to enter system menu for activation.

### ● RS-232 Interface

In RS-232 interface, all SCPI commands can be used for programming. If RS-232 interface is selected, in accordance with internal connection of data terminal equipment (DTE) and data communication equipment (DCE) as defined in EIA RS-232, the load is connected to another DTE (e.g., PC COM interface) with direct-connected Modem cable.



RS232 Pins of Plug

Base Pin Number	Description
1	No conjunction
2	TXD, data transmission
3	RXD, data receiving
4	No conjunction
5	GND, grounding
6	No conjunction
7	CTS, clear to send
8	RTS, request to send
9	No conjunction

### ● Communication Setup

Please ensure the PC and the load have the same configuration in the following items.

Baudrate: 9600(4800, 9600, 19200, 38400). You could enter the system menu to set the baudrate.

Data bit: 8

Stop bit: 1

Parity bit: (none, even, odd)

**EVEN** 8 data bits have even parity

**ODD** 8 data bits have odd parity

**NONE** 8 data bits have no parity

Native machine address: (0 ~31, factory default is 0)

Start Bit	8 Data Bits	Parity=None	Stop Bit
-----------	-------------	-------------	----------

## USB Interface

Connect the load and the computer using a cable with two USB interfaces (each end). All functions of the load can be programmed via USB.

After connecting the load and computer by USB, you need to install IT-E122 driver (download from

<http://www.itech.sh/support/Down-3.htm?model=IT-E122%E5%85%89%E9%9A%94%E7%A6%BB%E9%80%9A%E8%AE%AF%E6%8E%A5%E5%8F%A3%E9%A9%B1%E5%8A%A8> or contact ITECH agent). The device manager of PC will display 'Prolific USB-to-Serial COM Port' after installing.

Notes: Only IT8513B+/IT8514B+/IT8514C+/IT8516C+ models have the USB communication interface. After the driver is installed successfully, use USB line to connect in communication. (Please don't connect DB9 interface at the same time). Don't need to set the menu.

## Chapter2 SCPI Status Register

SCPI protocol supports the following four groups of registers.

### Questionable Status Register Group

Questionable status register group includes three 16-bit registers: status register, event register and enable register. When the corresponding status register bit is changed, the corresponding bit in the event register is set. If the corresponding bit in the enable register bit is set, it will be generated once Event (status byte register QUES is set). After executing an event register read operation, the event register will be automatically cleared. The status register is defined as follows:

Bit	Signal	Description
Bit0	VF	Either an overvoltage or a reverse voltage has occurred. This bit reflects the active state of the FLT pin on the back of the unit. The bit remains set until the condition is removed and PROT:CLE is programmed.
Bit1	OC	An over-current condition has occurred.
Bit3	OP	An overpower condition has occurred.
Bit4	OT	An over-temperature condition has occurred.
Bit8	SV	Remote measurement terminal is not connected.
Bit11	UNR	The input is unregulated, when the input is regulated the bit is cleared.
Bit13	OV	An over voltage condition has occurred. Both this bit and VF bit0 are set and the loads are turned off. Both bits remain set until the condition is removed and PROT:CLE is programmed.

### Standard Event Status Register Group

The standard event register group consists of two 16-bit registers: the event register and the enable register. When an event occurs, and if the corresponding bit in the enable register is set, an event occurs (the ESB in the Status Byte register is set). After executing an event register read operation, the event register will be automatically cleared. The event register is defined as follows:

Bit	Signal	Description
Bit0	OPC	The load has completed all pending operations. *OPC must be programmed for this bit to be set when pending operations are complete.
Bit2	QYE	The output queue was read with no data present or the data was lost. Errors in the range of 499 through 400 can set this bit.
Bit3	DDE	Device-Dependent Error. Memory was lost or self test failed. Errors in the range of 399 through 300 can set this bit.
Bit4	EXE	A command parameter was outside its legal range, inconsistent with the load's operation, or prevented from executing because of an operating condition. Errors in the range of 299 through 200 can set this bit.
Bit5	CME	A syntax or semantic error has occurred or the load received a <get> within a program message. Errors in

		the range of 199 through 100 can set this bit.
Bit7	PON	The unit has been turned off and then on since this bit was last read.

## Operation Status Register Group

The operational status register group consists of three 16-bit registers: status register, event register, and enable register. When the status register corresponding bit is changed, the corresponding event register bit will be set. If the corresponding bit in the enable register bit is set, it will be generated once Event (status byte register OPER is set). After executing an event register read operation, the event register will be automatically cleared. The status register is defined as follows:

Bit	Signal	Description
Bit0	CAL	The electronic load is calculated a new calibration constant.
Bit5	WTG	The electronic load is waiting for a trigger.

## Status Byte Register Group

The status byte register group consists of two 8-bit registers: event register and enable register. If the corresponding bit in the enable register bit is set, it will be generated once Event (status byte register RQS is set). The status byte register will be automatically cleared when an event register read is executed. The status register is defined as follows:

Bit	Signal	Description
Bit3	QUES	Indicates if an enabled questionable event has occurred.
Bit4	MAV	Indicates if the Output Queue contains data.
Bit5	ESB	Indicates if an enabled standard event has occurred.
Bit6	RQS	During a serial poll, RQS is returned and cleared.
Bit7	OPER	Indicates if an operation event has occurred.

---

## Chapter3 Essential Commands

---

### SYSTem:ERRor?

This command return the next error number, followed by a remote programming error message string.

Sequence is a FIFO buffer FIFO (first-in, first-out), when the error occurs, the error is stored in the cache. When it is read out, it is deleted from the sequence.

After reading all the errors, the query Returned "0, No Error". If the error accumulates too much that is more than the cache can bear, the last error of the sequence will be "-350, Too Many Errors".

#### Query Syntax

SYSTem:ERRor?

#### Parameters

None

#### Returned Parameters

<NR1>, <SRD>

#### Examples

SYSTem:ERRor[:NEXT]?

### SYSTem:VERSion?

This query returns the SCPI revision of the load used. The format is YYYY.V, where YYYY is the year and V is the revision number for that year.

#### Query Syntax

SYSTem:VERSion?

#### Parameters

None

#### Returned Parameters

<NR1>, <SRD>

#### Examples

SYSTem:VERS?

### SYSTem:SENSe[:STATe] <bool>

This command is used to control the remote compensation function on and off.

#### Command Syntax

SYSTem:SENSe[:STATe] <bool>



### Parameters

0|1|OFF|ON

### Examples

SYST:SENS ON

### Query Syntax

SYSTem:SENSe[:STATe]?

### Returned Parameters

0|1

### \*RST Value

OFF

## **SYSTem:SENSe[:STATe]?**

This command is used to query the status of remote compensation.

### Command Syntax

SYSTem:SENSe[:STATe]?

### Parameters

Bool

### Returned Parameters

<NR1>

## **SYSTem:LOCaI**

The command sets the load to local mode when in RS232 mode. Can use the buttons on front panel to operate.

### Command Syntax

SYSTem:LOCaI

### Parameters

None

### Examples

SYST:LOC

## **SYSTem:REMOte**

The command sets the load to remote mode when in RS232 mode. All the buttons except for the LOCAL and Shift buttons will lose function. In the remote state, press LOCAL key return to local mode.

## Command Syntax

SYSTem:REMOte

## Parameters

None

## Examples

SYST:REM

# SYSTem:RWLock

This command can set the load to remote mode, all the button on front panel will lose function including LOCAL button. Use SYSTem:LOCAL return to local mode.

## Command Syntax

SYSTem:RWLock

## Parameters

None

## Examples

SYST:RWL

## Related Commands

SYST:REM SYST:LOC

## Chapter4 Status Commands

Those commands configure the status registers of the electronic load.

### **STATus:QUEStionable[:EVENT]?**

This query returns the value of event register. Event register is read only register, it keeps all events sent to it. Read the quest event register will clear it.

#### Query Syntax

STATus:QUEStionable[:EVENT]?

#### Parameters

None

#### Examples

STAT:QUES:EVEN?

#### Returned Parameters

<NR1>

#### Related Commands

\*CLS

### **STATus:QUEStionable:CONDition?**

This command can read the parameter from quest condition register. It is a read only register, keep the real-time(not locked) query status of the load.

#### Query Syntax

STATus:QUEStionable:CONDition?

#### Parameters

None

#### Examples

STAT:QUES:COND?

#### ReturnedParameters

<NR1>

#### Related Commands

STAT:OPER:COND?

### **STATus:QUEStionable:ENABLE <NRf+>**

This command is used to set the value of the enable register.

## Command Syntax

STATus:QUEStionable:ENABLE <NRf+>

## Parameters

0 to 32767

## Examples

STAT:QUES:ENAB 32

## Query Syntax

STATus:QUEStionable:ENABLE?

## Returned Parameters

<NR1>

# STATus:QUEStionable:ENABLE?

This command is used to query the value of enable register.

## Command Syntax

STATus:QUEStionable:ENABLE?

## Parameters

None

## Returned Parameters

<NR1>

# STATus:OPERation[:EVENT]?

This command query the query operation event register values. The event register is read-only register, which holds (latches) all value passed by the NTR and, or PTR filter. Read channel operation event register will clear it.

## Query Syntax

STATus:OPERation[:EVENT]?

## Parameters

None

## Examples

STAT:OPER:EVEN?

## Returned Parameters

<NR1>

## Related Commands

**\*CLS**

## STATus:OPERation:CONDition?

This query returns the value of operation condition register. That is a read-only register that holds the real-time (unlatched) operational status of the electronic load.

### Query Syntax

STATus:OPERation:CONDition?

### Parameters

None

### Examples

STAT:OPER:COND?

### Returned Parameters

<NR1>

### Related Commands

STAT:QUES:COND?

## STATus:OPERation:ENABLE <NRf+>

The command is used to set the value of operations enable register. This register is a mask for enabling specific bits from the Operation Event register to set the operation summary bit (OPER) of the Status Byte register. The operation summary bit is the logical OR of all enabled Operation Event register bits.

### Command Syntax

STATus:OPERation:ENABLE <NRf+>

### Parameters

0 to 65535

### Default value

0

### Examples

STAT:OPER:ENAB 32

### Query Syntax

STATus:OPERation:ENABLE?

### Returned Parameters

<NR1>

### Related Parameters

STAT:OPER?

## **STATus:OPERation:ENABle?**

The command is used to query the value of operations enable register. This register is a mask for enabling specific bits from the Operation Event register to set the operation summary bit (OPER) of the Status Byte register. The operation summary bit is the logical OR of all enabled Operation Event register bits.

### **Command Syntax**

STATus:OPERation:ENABle <NR1>

### **Parameters**

None

### **Default value**

0

---

## Chapter5 Input Settings

---

These commands control the input of the electronic load. The INPut, CURRent, RESistance and VOLTage commands program the actual input current, resistance, and voltage.

### **[SOURce:]INPut <bool>**

This command is used to set the input status.

#### Command Syntax

[SOURce:]INPut <bool>

#### Parameters

0 | 1 | OFF | ON

#### \*RST Value

OFF

#### Examples

INP 1

#### Query Syntax

INPut[:STATe]?

#### Returned Parameters

0 | 1

### **[SOURce:]INPut?**

This command is used to query the input status.

#### Command Syntax

[SOURce:]INPut?

#### \*RST Value

OFF

#### Examples

INP?

### **[SOURce:]INPut:SHORT <bool>**

This command is used to set the status of short circuit function.

#### Command Syntax

[SOURce:]INPut:SHORT <bool>

## Parameters

0 | 1 | OFF | ON

## \*RST Value

OFF

## Examples

INP:SHOR 1

## Query Syntax

INPut:SHORt?

## Returned Parameters

0 | 1

## Related Commands

INP

# [SOURce:]INPut:SHORt?

This command is used to query the status of short circuit function.

## Command Syntax

[SOURce:]INPut:SHORt?

## \*RST Value

OFF

## Examples

INP:SHOR?

# [SOURce:]CURRent:RANGe <NRf+>

This command sets the current range of the electronic load module. When you program a range value, the load automatically selects the range that corresponds to the value that you program. If the value falls in a region where ranges overlap, the load selects the range with the highest resolution.

## Command Syntax

[SOURce:]CURRent:RANGe <NRf+>

## Parameters

MIN ~MAX | MINimum | MAXimum

## Unit

A



\*RST Value

MAXimum (high range)

Examples

CURR:RANGE MIN

Query Syntax

[SOURce:]CURRent:RANGe?

Returned Parameters

<NR2>

## **[SOURce:]CURRent:RANGe?**

This command is used to query the current range of the electronic load module.

Command Syntax

[SOURce:]CURRent:RANGe?

Parameters

None

Unit

A

\*RST Value

MAXimum (high range)

## **[SOURce:]VOLTage:RANGe <NRf+>**

This command sets the voltage range of the electronic load module. When you program a range value, the load automatically selects the range that corresponds to the value that you program. If the value falls in a region where ranges overlap, the load selects the range with the highest resolution.

Command Syntax

[SOURce:]VOLTage:RANGe <NRf+>

Parameters

MIN ~MAX | MINimum | MAXimum

Unit

V

\*RST Value

MAXimum(high range)

## Examples

SOUR:VOLT:RANG MIN

## Query Syntax

[SOURce:]VOLTage:RANGe?

## Returned Parameters

<NR2>

# [SOURce:]VOLTage:RANGe?

This command is used to query the voltage range of the electronic load module.

## Command Syntax

[SOURce:]VOLTage:RANGe?

## Parameters

None

## Unit

V

## \*RST Value

MAXimum(high range)

# [SOURce:]VOLTage:RANGe:AUTO[:STATe]

This command is used to set the auto-range state of the voltage meter of the load module.

## Command Syntax

[SOURce:]VOLTage:RANGe:AUTO[:STATe] <bool>

## Parameters

0 | 1 | ON | OFF

## Returned Parameters

1

## Examples

VOLT:RANG:AUTO 1

## Query Syntax

[SOURce:]VOLTage:RANGe:AUTO[:STATe]?

## Returned Parameters

<NR1>

## **[SOURce:]CURRent:SLEW[:BOTH] <NRf+>**

This command sets the slew rate for all programmed changes in the input current level of the electronic load. This command programs both positive and negative going slew rates.

### Command Syntax

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

### Parameters

MIN ~MAX | MAXimum | MINimum

### Unit

A / $\mu$ S

### \*RST Value

MAXimum

### Examples

CURR:SLEW 3

### Related Commands

[SOURce:]CURRent:SLEW?

### Returned Parameters

<NR2>

## **[SOURce:]CURRent:SLEW[:BOTH]?**

This command is used to query the slew rate for all programmed changes in the input current level of the electronic load.

### Command Syntax

[SOURce:]CURRent:SLEW[:BOTH]?

### Unit

A / $\mu$ S

### \*RST Value

MAXimum

## **[SOURce:]CURRent:SLEW:RISE <NRf+>**

This command sets the slew rate of the current for positive going transitions.

### Command Syntax

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

## Parameters

MIN~MAX | MAXimum | MINimum

## Unit

A/uS

## \*RST Value

MAXimum

## Examples

CURR:SLEW RISE 3

## Query Syntax

[SOURce:]CURRent:SLEW:RISE?

## Returned Parameters

<NR2

# [SOURce:]CURRent:SLEW:RISE?

This command is used to query the slew rate of the current for positive going transitions.

## Command Syntax

[SOURce:]CURRent:SLEW:RISE?

## Unit

A/uS

## \*RST Value

MAXimum

# [SOURce:]CURRent:SLEW:FALL <NRf+>

This command sets the slew rate of the current for negative going transitions.

## Command Syntax

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

## Parameters

MIN~MAX | MAXimum | MINimum

## Unit

A/uS

## \*RST Value

MAXimum

## Examples

CURR:SLEW FALL 3

## Query Syntax

[SOURce:]CURRent:SLEW:FALL?

## Returned Parameters

<NR2>

# [SOURce:]CURRent:SLEW:FALL?

This command is used to query the slew rate of the current for negative going transitions.

## Command Syntax

[SOURce:]CURRent:SLEW:FALL?

## Parameters

None

## Unit

A/uS

## \*RST Value

MAXimum

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

This command sets the current protection level. If the input current exceeds the current protection level for the time specified by CURR:PROT:DEL, the input is turned off.

---

**NOTE:** Use CURR:PROT:DEL to prevent momentary current limit conditions caused by programmed changes from tripping the overcurrent protection.

---

## Command Syntax

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

## Parameters

MIN ~ MAX | MINimum | MAXimum

## Unit

A

## \*RST Value

MAXimum

## Examples

CURR:PROT 3

## Query Syntax

[SOURce:]CURRent:PROTection[:LEVel]?

## Returned Parameters

NR2

# [SOURce:]CURRent:PROTection[:LEVel]?

This command is used to query the current protection level.

## Command Syntax

[SOURce:]CURRent:PROTection[:LEVel]?

## Unit

A

## \*RST Value

MAXimum

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

This command sets the power protection level. If the input power exceeds the power protection level for the time specified by POW:PROT:DEL, the input is turned off.

---

**NOTE:** Use POW:PROT:DEL to prevent momentary power limit conditions caused by programmed changes from tripping the over power protection.

---

## Command Syntax

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

## Parameters

MIN ~ MAX | MINimum | MAXimum

## Unit

W

## \*RST Value

MAXimum(high range)

## Examples

POW:PROT 100

## Query Syntax

[SOURce:]POWer:PROTection[:LEVel]?

## Returned Parameters

<NR2>

**[SOURce:]POWer:PROTection[:LEVel]?**

This command is used to query the power protection level.

## Command Syntax

[SOURce:]POWer:PROTection[:LEVel]?

## Unit

W

## \*RST Value

MAXimum(high range)

**[SOURce:]VOLTage:[LEVel:]ON <NRf+>**

This command sets the voltage of sink current on.

## Command Syntax

[SOURce:]VOLTage[:LEVel]:ON <NRf+>

## Parameters

None

## Unit

V

## \*RST Value

1

## Examples

VOLT:ON 3

## Query Syntax

[SOURce:]VOLTage[:LEVel]:ON?

## Returned Parameters

<NR2>

**[SOURce:]Voltage:[LEVel:]ON?**

This command is used to query the voltage of sink current on.

## Command Syntax

[SOURce:]VOLTage[:LEVel]:ON?

## Unit

V

## \*RST Value

1

## **[SOURce:]Voltage:[LEVel:]OFF <NRf+>**

This command is used to set the load start unloading voltage value.

## Command Syntax

[SOURce:]VOLTage[:LEVel]:OFF <NRf+>

## Parameters

MIN ~ MAX | MINimum | MAXimum

## Unit

V

## \*RST Value

0.5

## Examples

VOLT:OFF 2

## Query Syntax

[SOURce:]VOLTage[:LEVel]:OFF?

## Returned Parameters

<NR2>

## **[SOURce:]Voltage:[LEVel:]OFF?**

This command is used to query the load start unloading voltage value.

## Command Syntax

[SOURce:]VOLTage[:LEVel]:OFF?

## Unit

V

## \*RST Value

0.5



## [SOURce:]FUNCTION <function>

## [SOURce:]MODE <function>

These two commands are equivalent and are used to select the input mode of the load.

### Command Syntax

[SOURce:]FUNCTION <function>

[SOURce:]MODE <function>

### Parameters

CURRent | VOLTage | POWer | RESistance | DYNamic | LED | IMPedance

- CURRent: Constant Current Mode
- VOLTage: Constant Voltage Mode
- POWer: Constant Power Mode
- RESistance: Constant Resistance Mode
- DYNamic: Dynamic Operation Mode
- LED: LED Mode
- IMPedance: Constant Impedance Mode

### \*RST Value

CURRent

### Examples

MODE RES

### Query Syntax

[SOURce:]FUNCTION?

[SOURce:]MODE?

### Returned Parameters

<CRD>

## [SOURce:]FUNCTION?

## [SOURce:]MODE?

These two commands are equivalent and are used to query the input mode of the load.

### Command Syntax

[SOURce:]FUNCTION?

[SOURce:]MODE?

### Returned Parameters

CURRent | VOLTage | POWer | RESistance | DYNamic | LED | IMPedance

- CURRent: Constant Current Mode
- VOLTage: Constant Voltage Mode
- POWer: Constant Power Mode
- RESistance: Constant Resistance Mode
- DYNamic: Dynamic Operation Mode
- LED: LED Mode
- IMPedance: Constant Impedance Mode

\*RST Value

CURRent

Examples

MODE RES

## **[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <NRf+>**

This command is used to set the current setting in CC mode.

Command Syntax

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

Parameters

MIN~MAX|MINimum|MAXimum

Unit

A

\*RST Value

MINimum

Examples

CURR 5

Query Syntax

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

Returned Parameters

<NR2>

## **[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?**

This command is used to query the current setting in CC mode.

Command Syntax

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

## Parameters

None

## Unit

A

## \*RST Value

MINimum

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

This command is used to set the voltage setting in CV mode.

## Command Syntax

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

## Parameters

MIN~MAX|MINimum|MAXimum

## Unit

V

## \*RST Value

MAXimum

## Examples

VOLT 5

## Query Syntax

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

## Returned Parameters

<NR2>

# **[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]?**

This command is used to query the voltage setting in CV mode.

## Command Syntax

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

## Parameters

None

## Unit

V

\*RST Value

MAXimum

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

This command is used to set the power setting in CP mode.

Command Syntax

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

Parameters

MIN~MAX|MINimum|MAXimum

Unit

W

\*RST Value

MINimum

Examples

POW 10

Query Syntax

[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]?

Returned Parameters

<NR2>

## **[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]?**

This command is used to query the power setting in CP mode.

Command Syntax

[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]?

Parameters

None

Unit

W

\*RST Value

MINimum

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

This command is used to set the resistance setting in CR mode.

### Command Syntax

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

### Parameters

MIN~MAX|MINimum|MAXimum

### Unit

ohm

### \*RST Value

MAXimum

### Examples

RES 5

### Query Syntax

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude]?

### Returned Parameters

<NR2>

## **[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude]?**

This command is used to query the resistance setting in CR mode.

### Command Syntax

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude]?

### Parameters

None

### Unit

ohm

### \*RST Value

MAXimum

## **[SOURce:]DYNamic:HIGH[:LEVel] <NRf+>**

This command is used to set the high-level load current in dynamic mode.

## Command Syntax

[SOURce:]DYNamic:HIGH[:LEVel] <NRf+>

## Parameters

MIN~MAX|MINimum|MAXimum

## Unit

A

## \*RST Value

0

## Examples

DYN:HIGH 10

## Query Syntax

[SOURce:]DYNamic:HIGH[:LEVel]?

## Returned Parameters

<NR2>

# **[SOURce:]DYNamic:HIGH[:LEVel]?**

This command is used to query the high-level load current in dynamic mode.

## Command Syntax

[SOURce:]DYNamic:HIGH[:LEVel]?

## Parameters

None

## Unit

A

## \*RST Value

MIN

## Examples

DYN:HIGH 10

# **[SOURce:]DYNamic:HIGH:DWELI <NRf+>**

This command is used to set the duration of high-level load current in dynamic mode.

## Command Syntax

[SOURce:]DYNamic:HIGH:DWELI <NRf+>

## Parameters

MIN~MAX|MINimum|MAXimum

## Unit

S

## \*RST Value

MIN

## Examples

DYN:HIGH:DWELI 0.01

## Query Syntax

[SOURce:]DYNamic:HIGH:DWELI?

## Returned Parameters

<NR2>

# **[SOURce:]DYNamic:HIGH:DWELI?**

This command is used to query the duration of high-level load current in dynamic mode.

## Command Syntax

[SOURce:]DYNamic:HIGH:DWELI?

## Parameters

None

## Unit

S

## \*RST Value

MIN

# **[SOURce:]DYNamic:LOW[:LEVel] <NRf+>**

This command is used to set the low-level load current in dynamic mode.

## Command Syntax

[SOURce:]DYNamic:LOW[:LEVel] <NRf+>

## Parameters

MIN~MAX|MINimum|MAXimum

## Unit

A

\*RST Value

MIN

Examples

DYN:LOW 1

Query Syntax

[SOURce:]DYNamic:LOW[:LEVel]?

Returned Parameters

<NR2>

## **[SOURce:]DYNamic:LOW[:LEVel]?**

This command is used to query the low-level load current in dynamic mode.

Command Syntax

[SOURce:]DYNamic:LOW[:LEVel]?

Parameters

None

Unit

A

\*RST Value

0

## **[SOURce:]DYNamic:LOW:DWELI <NRf+>**

This command is used to set the duration of low-level load current in dynamic mode.

Command Syntax

[SOURce:]DYNamic:LOW:DWELI <NRf+>

Parameters

MIN~MAX|MINimum|MAXimum

Unit

S

\*RST Value

MIN

Examples

DYN:LOW:DWEL 1



## Query Syntax

[SOURce:]DYNamic:LOW:DWELI?

## Returned Parameters

<NR2>

# **[SOURce:]DYNamic:LOW:DWELI?**

This command is used to query the duration of low-level load current in dynamic mode.

## Command Syntax

[SOURce:]DYNamic:LOW:DWELI?

## Parameters

None

## Unit

S

## \*RST Value

0.00002

# **[SOURce:]DYNamic:SLEW <NRf+>**

This command is used to set the current slope in dynamic mode.

## Command Syntax

[SOURce:]DYNamic:SLEW <NRf+>

## Parameters

MIN~MAX|MINimum|MAXimum

## Unit

A/uS

## \*RST Value

MAX

## Examples

DYN:SLEW 3

## Query Syntax

[SOURce:]DYNamic:SLEW?

## Returned Parameters

<NR2>

## [SOURce:]DYNamic:SLEW?

This command is used to query the current slope in dynamic mode.

### Command Syntax

[SOURce:]DYNamic:SLEW?

### Parameters

None

### Unit

A/uS

### \*RST Value

MAX

## [SOURce:]DYNamic:SLEW:RISE <NRf+>

This command is used to set the current rising slope in dynamic mode.

### Command Syntax

[SOURce:]DYNamic:SLEW:RISE <NRf+>

### Parameters

MIN~MAX|MINimum|MAXimum

### Unit

A/uS

### \*RST Value

MAX

### Examples

DYN:SLEW:RISE 3

### Query Syntax

[SOURce:]DYNamic:SLEW:RISE?

### Returned Parameters

<NR2>

## [SOURce:]DYNamic:SLEW:RISE?

This command is used to query the current rising slope in dynamic mode.

### Command Syntax

[SOURce:]DYNamic:SLEW:RISE?

## Parameters

None

## Unit

A/uS

## \*RST Value

MAX

**[SOURce:]DYNamic:SLEW:FALL <NRf+>**

This command is used to set the current falling slope in dynamic mode.

## Command Syntax

`[SOURce:]DYNamic:SLEW:FALL <NRf+>`

## Parameters

MIN~MAX|MINimum|MAXimum

## Unit

A/uS

## \*RST Value

MAX

## Examples

`DYN:SLEW:FALL 3`

## Query Syntax

`[SOURce:]DYNamic:SLEW:FALL?`

## Returned Parameters

&lt;NR2&gt;

**[SOURce:]DYNamic:SLEW:FALL?**

This command is used to query the current falling slope in dynamic mode.

## Command Syntax

`[SOURce:]DYNamic:SLEW:FALL?`

## Parameters

None

## Unit

A/uS

\*RST Value

MAX

## **[SOURce:]DYNamic:MODE <mode>**

This command is used to set the working mode in dynamic mode.

Command Syntax

[SOURce:]DYNamic:MODE <mode>

Parameters

CONTinuous|PULSe|TOGGLE

\*RST Value

CONTinuous

Examples

DYN:MODE PULS

Query Syntax

[SOURce:]DYNamic:MODE?

Returned Parameters

<CRD>

## **[SOURce:]DYNamic:MODE?**

This command is used to query the working mode in dynamic mode.

Command Syntax

[SOURce:]DYNamic:MODE?

Parameters

None

\*RST Value

CONTinuous

---

## Chapter6 Measure Commands

---

This signal measure command is used to get the read back value. You can use this command to control the measurement process.

### **MEASure[:SCALar]:VOLTage[:DC]?**

This command is used to query the average voltage of the electronic load.

#### Command Syntax

MEASure[:SCALar]:VOLTage[:DC]?

#### Parameters

None

#### Examples

MEAS:VOLT?

#### Returned Parameters

<NR2>

### **MEASure[:SCALar]:VOLTage:MAXimum?**

This command is used to query the peak voltage of the electronic load.

#### Command Syntax

MEASure[:SCALar]:VOLTage:MAXimum?

#### Parameters

None

#### Examples

MEAS:VOLT:MAX?

#### Returned Parameters

<NR2>

### **MEASure[:SCALar]:VOLTage:MINimum?**

This command is used to query the minimum voltage of the electronic load.

#### Command Syntax

MEASure[:SCALar]:VOLTage:MINimum?

#### Parameters

None

## Examples

MEAS:VOLT:MIN?

## Returned Parameters

<NR2>

## **MEASure[:SCALar]:VOLTage:PTPeak?**

This command is used to query the voltage peak-to-peak value.

## Command Syntax

MEASure[:SCALar]:VOLTage:PTPeak?

## Parameters

None

## Examples

MEAS:VOLT:PTP?

## Returned Parameters

<NR2>

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

This command is used to query the current average of the electronic load.

## Command Syntax

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

## Parameters

None

## Examples

MEAS:CURRE?

## Returned Parameters

<NR2>

## **MEASure[:SCALar]:CURRent:MAXimum?**

This command is used to query the peak current of the electronic load.

## Command Syntax

MEASure[:SCALar]:CURRent:MAXimum?

## Parameters

None

## Examples

MEAS:CURRE:MAX?

## Returned Parameters

<NR2>

## **MEASure[:SCALar]:CURRent:MINimum?**

This command is used to query the minimum current of the electronic load.

## Command Syntax

MEASure[:SCALar]:CURRent:MINimum?

## Parameters

None

## Examples

MEAS:CURRE:MIN?

## Returned Parameters

<NR2>

## **MEASure[:SCALar]:CURRent:PTPeak?**

This command is used to query the current peak-to-peak value.

## Command Syntax

MEASure[:SCALar]:CURRent:PTPeak?

## Parameters

None

## Examples

MEAS:CURRE:PTP?

## Returned Parameters

<NR2>

## **MEASure[:SCALar]:POWer[:DC]?**

This command is used to query the power average.

## Command Syntax

MEASure [:SCALar]:POWer[:DC]?

## Parameters

None

## Examples

MEAS:POWer?

## Returned Parameters

<NR2>

# **MEAS[:SCALar]:RESistance[:DC]?**

This command is used to query the equivalent impedance.

## Command Syntax

MEAS[:SCALar]:RESistance[:DC]?

## Parameters

None

## Examples

MEAS: RESistance?

## Returned Parameters

<NR2>



---

## Chapter7 OCP Testing Commands

---

### **OCP[:STATe] <bool>**

This command is used to set the OCP test status.

#### Command Syntax

OCP[:STATe] <bool>

#### Parameters

0|1|OFF|ON

#### Examples

OCP ON

#### Query Syntax

OCP[:STATe]?

#### Returned Parameters

0|1

### **OCP[:STATe]?**

This command is used to query the OCP test status. At the end of OCP test, it returns 0.

#### Command Syntax

OCP[:STATe]?

#### Parameters

0|1|OFF|ON

### **OCP:IStart <NRf+>**

This command is used to set the start current of the OCP test.

#### Command Syntax

OCP:IStart <NRf+>

#### Parameters

MIN~MAX

#### Unit

A

## Examples

OCP:IST 3

## Query Syntax

OCP:ISart?

## Returned Parameters

<NR2>

## OCP:ISart?

This command is used to query the start current of the OCP test.

## Command Syntax

OCP:ISart?

## Parameters

None

## Unit

A

## OCP:IEND <NRf+>

This command is used to set the OCP cut-off current.

## Command Syntax

OCP:IEND <NRf+>

## Parameters

MIN~MAX

## Unit

A

## Examples

OCP:IEND 6

## Query Syntax

OCP:IEND?

## Returned Parameters

<NR2>

## OCP:IEND?

This command is used to query the OCP cut-off current.

## Command Syntax

OCP:IEND?

## Parameters

None

## Unit

A

# OCP:STEP

This command is used to set the current step of OCP test.

## Command Syntax

OCP:STEP<NR1>

## Parameters

1~1000

## Examples

OCP:STEP 500

## Query Syntax

OCP:STEP?

## Returned Parameters

<NR2>

# OCP:STEP?

This command is used to query the current step of OCP test.

## Command Syntax

OCP:STEP?

## Parameters

None

# OCP:DWELI <NRf+>

This command is used to set the single-step dwell time of OCP test.

## Command Syntax

OCP:DWELI <NRf+>

## Parameters

0.00001~0.99999

**Unit**

S

**Examples**

OCP:DWEL 0.01 or OCP:DWEL 10ms

**Query Syntax**

OCP:DWEL?

**Returned Parameters**

&lt;NR2&gt;

**OCP:DWEL?**

This command is used to query the single-step dwell time of OCP test.

**Command Syntax**

OCP:DWEL?

**Parameters**

None

**Unit**

S

**OCP:VTRig <NRf+>**

This command is used to set the OCP trigger level.

**Command Syntax**

OCP:VTRig &lt;NRf+&gt;

**Parameters**

MIN~MAX

**Unit**

V

**Examples**

OCP:VTR 11.8

**Query Syntax**

OCP:VTRig?

**Returned Parameters**

&lt;NR2&gt;

## OCP:VTRig?

This command is used to query the OCP trigger level.

### Command Syntax

OCP:VTRig?

### Returned Parameters

MIN~MAX

### Unit

V

## OCP:RESult[:OCP]?

This command is used to query the current value of OCP.

### Command Syntax

OCP:RESult[:OCP]?

### Returned Parameters

<NRf+>

### Unit

A

### Examples

OCP:RES?

### Returned Parameters

4.68

## OCP:RESult:PMAX?

This command is used to query the PMAX value.

### Command Syntax

OCP:RESult:PMAX?

### Returned Parameters

<NR2>,<NR2>,<NR2>

### Unit

W V A

### Examples

OCP:RES:PMAX?

## Returned Parameters

55.34 11.8 4.69

This example shows that the maximum output power of P<sub>MAX</sub> is 55.34W, at this moment the voltage is 11.8V, the current is 4.69A.

---

## Chapter8 Peak Testing Commands

---

These commands are used to read the maximum/minimum value after startup test. Peak value recording is automatically cleared when the Peak test is started.

### **PEAK[:STATe] <bool>**

This command is used to set the peak test status.

#### Command Syntax

PEAK[:STATe] <bool>

#### Examples

Peak:ON

### **PEAK CLEAr**

This command is used to clear the peak record.

#### Command Syntax

PEAK CLEAr

#### Examples

Peak CLE

### **PEAK:VOLTage:MAXimum?**

This command is used to query the maximum voltage.

#### Command Syntax

PEAK:VOLTage:MAXimum?

#### Examples

PEAK:VOLT:MAX?

#### Returned Parameters

<NR2>

### **PEAK:VOLTage:MINimum?**

This command is used to query the minimum voltage.

#### Command Syntax

PEAK:VOLTage:MINimum?

## Examples

PEAK:VOLT:MIN?

## Returned Parameters

<NR2>

## **PEAK:CURRent:MAXimum?**

This command is used to query the maximum current.

## Command Syntax

PEAK:CURRent:MAXimum?

## Examples

PEAK: CURR:MAX?

## Returned Parameters

<NR2>

## **PEAK: CURRent:MINimum?**

This command is used to query the minimum current.

## Command Syntax

PEAK: CURRent:MINimum?

## Examples

PEAK:CURR:MIN?

## Returned Parameters

<NR2>



---

## Chapter9 Voltage Rise/Fall Time Test

---

### **TIME:VOLTage:LOW <Voltage>**

This command is used to set the start rising voltage value of the load when measuring the rise time of voltage output.

#### Command Syntax

TIME:VOLTage:LOW <Voltage>

#### Parameters

0 through MAX|MINimum|MAXimum|DEFault

#### Unit

V

#### Returned Parameters

MINimum

#### Query Syntax

TIME:VOLTage:LOW?

### **TIME:VOLTage:LOW?**

This command is used to query the start rising voltage value.

#### Command Syntax

TIME:VOLTage:LOW?

#### Parameters

None

#### Returned Parameters

<NRf>

### **TIME:VOLTage:HIGH <Voltage>**

This command is used to set the starting drop voltage value of the load when measuring the fall time of voltage output.

#### Command Syntax

TIME:VOLTage:HIGH <Voltage>

#### Parameters

0 through MAX|MINimum|MAXimum|DEFault

Unit

V

Returned Parameters

MAXimum

Query Syntax

TIME:VOLTage:HIGH?

## **TIME:VOLTage:HIGH?**

This command is used to query the starting drop voltage value.

Command Syntax

TIME:VOLTage:HIGH?

Parameters

None

Returned Parameters

<NRf>

## **TIME:VOLTage:UP?**

This command is used to read the voltage rise time.

Command Syntax

TIME:VOLTage:UP?

Parameters

None

Returned Parameters

<NRf>

## **TIME:VOLTage:DOWN?**

This command is used to read the voltage fall time.

Command Syntax

TIME:VOLTage:DOWN?

Parameters

None

Returned Parameters

<NRf>

---

## Chapter10 Timing Test Commands

---

### **TIMing[:STATe]**

This command is used to start or stop the Timing test.

After sending the TIM 1 command, the front panel of the instrument begins to display the timing of the test. Unit: S;

After sending the TIM 0 command, the front panel timing function disappears.

#### Command Syntax

TIMing[:STATe] <bool>

#### Parameters

0|1|OFF|ON

#### Examples

TIM ON

#### Query Syntax

TIMing[:STATe]?

#### Returned Parameters

0|1

### **TIMing:LOAD:SETTing**

This command is used to determine whether to change the load setting during the Timing test.

#### Command Syntax

TIMing:LOAD:SETTing <bool>

#### Parameters

0|1|OFF|ON

When it is set to ON, after the test is started, the load is changed according to the setting of TIMing:LOAD, and after the test is ended, the load input is turned off; if it is set to OFF, the test is started/stopped without changing the load setting.

#### Examples

TIM:LOAD:SETT OFF

#### Query Syntax

TIMing:LOAD:MODE?

## Returned Parameters

<mode>

## TIMing:LOAD:MODE

This command is used to set the load mode during Timing test.

### Command Syntax

TIMing:LOAD:MODE <mode>

### Parameters

CURR|VOLT|POW|RES

### Examples

TIM:LOAD:MODE CURR

### Query Syntax

TIMing:LOAD:MODE?

## Returned Parameters

<mode>

## Related Commands

If **TIM:LOAD:SETT OFF** is executed, the setting of this command does not take effect.

## TIMing:LOAD:VALue

This command is used to set the load parameters for the Timing test.

### Command Syntax

TIMing:LOAD:VALue <Nrf+>

### Parameters

A / V / W /  $\Omega$ , depending on **TIMing:LOAD:MODE**

### Examples

TIM:LOAD:VAL 1

### Query Syntax

TIMing:LOAD:VALue?

## Returned Parameters

<NR2>

## Related Commands

If **TIM:LOAD:SETT OFF** is executed, the setting of this command does not take effect.

## **TIMing:TStart:SOURce**

This command sets the trigger source for starting the test.

### Command Syntax

TIMing:TStart:SOURce <source>

### Parameters

VOLT|CURR|EXT

### Examples

TIM:TST:SOUR VOLT

### Query Syntax

TIMing:TStart:SOURce?

### Returned Parameters

<source>

## **TIMing:TStart:EDGE**

This command is used to set the trigger edge of the start test.

### Command Syntax

TIMing:TStart:EDGE <edge>

### Parameters

RISE|FALL

### Examples

TIM:TST:EDGE RISE

### Query Syntax

TIMing:TStart:EDGE?

### Returned Parameters

<edge>

## **TIMing:TStart:LEVel**

This command is used to set the trigger level for the start test.

### Command Syntax

TIMing:TStart:LEVel <Nrf+>

### Parameters

Depends on the start trigger source, which is the setting of the **TIMing:TStart:SOURce** command.

## Examples

TIM:TST:LEV 1

## Query Syntax

TIMing:TSTart:LEVel?

## Returned Parameters

<NR2>

# **TIMing:TEND:SOURce**

This command sets the trigger source for ending the test.

## Command Syntax

TIMing:TEND:SOURce <source>

## Parameters

VOLT|CURR|EXT

## Examples

TIM:TEND:SOUR VOLT

## Query Syntax

TIMing:TEND:SOURce?

## Returned Parameters

<source>

# **TIMing:TEND:EDGE**

This command is used to set the trigger edge of the end test.

## Command Syntax

TIMing:TEND:EDGE <edge>

## Parameters

RISE|FALL

## Examples

TIM:TEND:EDGE RISE

## Query Syntax

TIMing:TEND:EDGE?

## Returned Parameters

<edge>

## **TIMing:TEND:LEVel**

This command sets the trigger level for ending the test.

### Command Syntax

TIMing:TEND:LEVel <Nrf+>

### Parameters

Depends on the end trigger source, which is the setting of the **Timing:TEND:SOURce** command.

### Examples

TIM:TEND:LEV 1

### Query Syntax

TIMing:TEND:LEVel?

### Returned Parameters

<NR2>

## **TIMing:RESult**

This command is used to query Timing test results.

### Command Syntax

TIMing:RESult?

### Unit

S

### Examples

TIM:RES?

### Returned Parameters

<NR2>

## Chapter11 CR-LED Commands

This chapter describes the LED analog function related commands. The user needs to set three parameters: **Vo**, **Io**, and **Rd Coeff.**. Where **Io** is the rated output current of the LED power supply under test; **Vo** is the forward working voltage of the LED when the forward working current is **Io**, and it can be obtained from the VI curve in the LED specification. The real load is the series connection of n LEDs, then **Vo** should be set to n times of the single-section parameter, or it can be set to any value within the LED power supply output voltage range; **Rd Coeff.** is the ratio of the series equivalent resistance (Rd) to the total equivalent resistance (Vo/Io) of the LED load, that is,  $Rd\ Coeff. = Rd/(Vo/Io)$ . In series applications, the **Rd Coeff.** is only related to the selected LED's VI curve, regardless of the number of series nodes.

### LED:VOLTage

This command is used to set **LED Vo**.

#### Command Syntax

LED:VOLTage <Nrf+>

#### Parameters

0.001~MAX

#### Examples

LED:VOLT 18

#### Query Syntax

LED:VOLT?

#### Returned Parameters

<NR2>

### LED:CURREnt

This command is used to set **LED Io**.

#### Command Syntax

LED:CURREnt <Nrf+>

#### Parameters

0~MAX

#### Examples

LED:CURR 0.35

#### Query Syntax

LED:CURR?



## Returned Parameters

<NR2>

## LED:RCOeff

This command is used to set **LED Rd Coeff.**

## Command Syntax

LED:RCOeff <Nrf+>

## Parameters

0.001~1

## Examples

LED:RCO 0.2

## Query Syntax

LED:RCO?

## Returned Parameters

<NR2>

---

## Chapter12 IEEE488 Commands

---

This chapter describes IEEE-488 common commands provided by the IT8500+ series electronic load.

### **\*CLS - Clear Status**

This command clears the registers:

- \_ Standard Event Register
- \_ Operation Event Register
- \_ Questionable Event Register
- \_ Error Queue

Command syntax

\*CLS

Parameters

None

### **\*ESE <NRf> - Event Enable**

This command programs the Standard Event Status Enable register bits. The programming determines which events of the Standard Event Status Event register (see \*ESR?) are allowed to set the ESB (Event Summary Bit) of the Status Byte register. A "1" in the bit position enables the corresponding event. All of the enabled events of the Standard Event Status Event Register are logically ORed to cause the Event Summary Bit (ESB) of the Status Byte Register to be set. See chapter "Programming the Status Registers" for descriptions of the Standard Event Status registers.

The query reads the Standard Event Status Enable register.

Command Syntax

\*ESE <NRf>

Parameters

0 to 255

Powe-on Value

see \*PSC

Examples

\*ESE 129

Query Syntax

\*ESE?

## Returned Parameters

<NR1>

## Related Commands

\*ESR? \*PSC \*STB?

## **\*ESE?**

This query reads the Standard Event Enable register.

## Command Syntax

\*ESE <NRf>

## Returned Parameters

0 to 255

## Default value

see \*PSC

## **\*ESR?**

This query reads the Standard Event Status Event register. Reading the register clears it. The bit configuration of this register is the same as the Standard Event Status Enable register (see \*ESE). See chapter “Programming the Status Registers” for a detailed explanation of this register.

## Query Syntax

\*ESR?

## Parameters

None

## Returned Parameters

<NR1> (register value)

## Related Commands

\*CLS \*ESE \*ESE? \*OPC

## **\*IDN?**

This query requests the electronic load to identify itself. It returns the data in four fields separated by commas.

## Query Syntax

\*IDN?

## Parameters

None

## Examples

ITECH Ltd, IT85XX, XXXXXXXXXXXXXXXXXXXX, 1.21-1.28

## Returned Parameters

<AARD>	Field	Information
--------	-------	-------------

## Technologies Manufacturer

ITECH Ltd

## Model

IT85XX

## Serial Number

XXXX.....

## Firmware Revision

1.21-1.28

## \*OPC

This command causes the interface to set the OPC bit (bit 0) of the Standard Event Status register when the electronic load has completed all pending operations. (See \*ESE for the bit configuration of the Standard Event Status registers.) Pending operations are complete when:

- All commands sent before \*OPC have been executed. This includes overlapped commands. Most commands are sequential and are completed before the next command is executed. Overlapped commands are executed in parallel with other commands. Commands that affect trigger actions are overlapped with subsequent commands sent to the electronic load. The \*OPC command provides notification that all overlapped commands have been completed.
- All triggered actions are completed and the trigger system returns to the Idle state.

\*OPC does not prevent processing of subsequent commands but Bit 0 will not be set until all pending operations are completed. The query causes the interface to place an ASCII "1" in the Output Queue when all pending operations are completed.

## Command Syntax

\*OPC

## Parameters

None

## Query Syntax

\*OPC?

## Returned Parameters

<NR1>

## Related Commands

\*TRIG \*WAI

## \*OPC?

This command is used to query whether the command execution is completed or not.

## Command Syntax

\*OPC?

## Parameters

None

## \*PSC

This command is used to control whether the electronic load will generate a service request when power on again.

1 OR ON: When the load power on, status byte enable register, operator event enable register, query event enable register and standard event enable register will be cleared.

0 OR OFF: The value of status byte enable register, operator event enable register, query event enable register and standard event enable register will be stored in the none-volatile storage, which will be recalled when power on.

## Command Syntax

\*PSC <bool>

## Parameters

0|1|ON|OFF

## Query Syntax

\*PSC?

## Returned Parameters

0|1

## \*PSC?

This command is used to query the status byte enable register.

## Command Syntax

\*PSC?

## Returned Parameters

0|1|ON|OFF

## **\*RCL**

This command restores the electronic load to a state that was previously stored in memory with a \*SAV command to the specified location. All states are recalled with the following exceptions:

CAL:STATe is set to OFF

The trigger system is set to the Idle state by an implied ABORt command (this cancels any uncompleted trigger actions)

NOTE: The device state stored in location 0 is automatically recalled at power turn-on.

### Command Syntax

\*RCL <NRf>

### Parameters

0 to 9

### Examples

\*RCL 3

### Related Commands

\*PSC   \*RST   \*SAV

## **\*RST**

This command reset the electronic load to the factory-defined states.

### Command Syntax

\*RST

### Parameters

None

## **\*SAV**

This command stores the present state of the electronic load to a specified location in memory. Up to 100 states can be stored. If a particular state is desired at power-on, it should be stored in location 0. It then will be recalled at power-on if the power-on state is set to RCL0. Use \*RCL to retrieve instrument states.

### Command Syntax

\*SAV <NRf>

### Parameters

0 - 99

### Examples

\*SAV 3

## Related Commands

\*PSC   \*RST   \*RCL

## **\*SRE**

This command sets the condition of the Service Request Enable Register. This register determines which bits from the Status Byte Register (see \*STB for its bit configuration) are allowed to set the Master Status Summary (MSS) bit and the Request for Service (RQS) summary bit. A 1 in any Service Request Enable Register bit position enables the corresponding Status Byte Register bit and all such enabled bits then are logically ORed to cause Bit 6 of the Status Byte Register to be set.

When the controller conducts a serial poll in response to SRQ, the RQS bit is cleared, but the MSS bit is not. When \*SRE is cleared (by programming it with 0), the electronic load cannot generate an SRQ to the controller. The query returns the current state of \*SRE.

## Command Syntax

\*SRE <NRf>

## Parameters

0 to 255

## Default value

see \*PSC

## Examples

\*SRE 128

## Query Syntax

\*SRE?

## Returned Parameters

<NR1> (register binary value)

## Related Commands

\*ESE   \*ESR   \*PSC

## **\*SRE?**

This command is used to query status byte group enable register.

## Command Syntax

\*SRE?

## Returned Parameters

0 - 255

**Default value**

see \*PSC

**\*STB?**

This query reads the Status Byte register, which contains the status summary bits and the Output Queue MAV bit. Reading the Status Byte register does not clear it. The input summary bits are cleared when the appropriate event registers are read (see chapter “Programming the Status Registers” for more information). A serial poll also returns the value of the Status Byte register, except that bit 6 returns Request for Service (RQS) instead of Master Status Summary (MSS). A serial poll clears RQS, but not MSS. When MSS is set, it indicates that the electronic load has one or more reasons for requesting service.

**Query Syntax**

\*STB?

**Parameters**

None

**Returned Parameters**

&lt;NR1&gt; (register value)

**Related Commands**

\*SRE   \*ESR   \*ESE

**\*TST?**

This command causes the electronic load to do a self-test and report any error. The reference information for the return value is as follows:

0: indicates no error

1: indicates that the module failed to initialize.

3: indicates that the module calibration data is lost.

4 or 5: indicates EEPROM error

**Command Syntax**

TST?

**Parameters**

None

**Returned Parameters**

&lt;NR1&gt; 0 indicates the electronic load has passed selftest.

Non-zero indicates an error code.

**\*TRG**

This command generates a trigger to any system that has BUS selected as its



source.

## Chapter13 Error Messages

### Error Number List

This chapter gives the error numbers and descriptions that are returned by the electronic load. Error numbers are returned in two ways:

- Error numbers are displayed on the front panel
- Error numbers and messages are read back with the `SYSTem:ERRor?` query. `SYSTem:ERRor?` returns the error number into a variable and returns two parameters, an NR1 and a string.

The following table lists the errors that are associated with SCPI syntax errors and interface problems. It also lists the device dependent errors. Information inside the brackets is not part of the standard error message, but is included for clarification. When errors occur, the Standard Event Status register records them in bit 2, 3, 4, or 5:

### Command Errors 100 through 199 (sets Standard Event Status Register bit #5 CME)

Error	Error String
101	device independed error Too many numeric suffices in Command Spec
110	No Input Command to parse
114	Numeric suffix is invalid value
116	Invalid value in numeric or channel list, e.g. out of range
117	Invalid number of dimensions in a channel list
120	Parameter of type Numeric Value overflowed its storage
130	Wrong units for parameter
140	Wrong type of parameter(s)
150	Wrong number of parameters
160	Unmatched quotation mark (single/double) in parameters
165	Unmatched bracket
170	Command keywords were not recognized
180	No entry in list to retrieve (number list or channel list)
190	Too many dimensions in entry to be returned in parameters
191	Too many char

### Execution Errors –200 through –299 (sets Standard Event Status Register bit #4 EXE)

Error	Error String
–200	Execution error [generic]
–221	Settings conflict [check current device state]
–222	Data out of range [e.g., too large for this device]
–223	Too much data [out of memory; block, string, or expression too long]
–224	Illegal parameter value [device-specific]
–225	Out of memory

-270	Macro error
-272	Macro execution error
-273	Illegal macro label
-276	Macro recursion error
-277	Macro redefinition not allowed

System Errors -300 through -399 (sets Standard Event Status Register bit #3 DDE)

Error	Error String
-310	System error [generic]
-350	Too many errors [errors beyond 9 lost due to queue overflow]

Query Errors "C400 through "C499 (sets Standard Event Status Register bit #2)

Error	Error String
-400	Query error [generic]
-410	Query INTERRUPTED [query followed by DAB or GET before response complete]
-420	Query UNTERMINATED [addressed to talk, incomplete programming message received]
-430	Query DEADLOCKED [too many queries in command string]
-440	Query UNTERMINATED [after indefinite response]

Selftest Errors 0 through 99 (sets Standard Event Status Register bit #3)

Error	Error String
0	No error
1	Module Initialization Lost
2	Mainframe Initialization Lost
3	Module Calibration Lost
4	Non-volatile RAM STATE section checksum failed
5	Non-volatile RAM RST section checksum failed
10	RAM selftest
11	CVDAC selftest 1
12	CVDAC selftest 2
13	CCDAC selftest 1
14	CCDAC selftest 2
15	CRDAC selftest 1
16	CRDAC selftest 2
20	Input Down
40	Flash write failed
41	Flash erase failed
80	Digital I/O selftest error

Device-Dependent Errors 100 through 32767 (sets Standard Event Status Register bit #3)

Error	Error String
213	RS-232 buffer overrun error

216	RS-232 receiver framing error
217	RS-232 receiver parity error
218	RS-232 receiver overrun error
220	Front panel uart overrun
221	Front panel uart framing
222	Front panel uart parity
223	Front panel buffer overrun
224	Front panel timeout
225	Front Crc Check error
226	Front Cmd Error
401	CAL switch prevents calibration
402	CAL password is incorrect
403	CAL not enabled
404	Computed readback cal constants are incorrect
405	Computed programming cal constants are incorrect
406	Incorrect sequence of calibration commands
407	CV or CC status is incorrect for this command
603	FETCH of data that was not acquired
604	Measurement overrange

## **Contact Us**

Thanks for purchasing ITECH products. In case of any doubts, please contact us as follows:

1. Visit ITECH website: [www.itechate.com](http://www.itechate.com).
2. Select the most convenient contact method for further information.