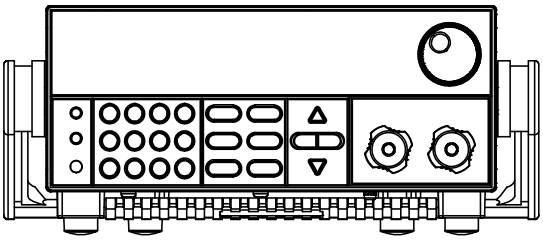
Programmable DC Electronic Load

Series IT8800 Programming Guide



Model: IT8811/IT8812/IT8812B/IT8812C
IT8813/IT8813B/IT8813C/IT8814/IT8814B/
IT8814C/IT8816/IT8816B/IT8816C/IT8817/
IT8817B/IT8817C/IT8818/IT8818B/IT8818C/
IT8818D/IT8830/IT8830B/IT8830H/IT8831/
IT8831B/IT8831H/IT8832/IT8832B/IT8832H/
IT8833/IT8833B/IT8833H/IT8834B/IT8834H/
IT8835B/IT8835H/IT8836B/IT8836H/IT8837B/
IT8837H/IT8838B/IT8838H/IT8839B/IT8839H

Revision: V2.1

Statement

Manual Article No.

IT8800-402159

Revision

Revision 2, published on

August. 1, 2022

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

IT8800 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

- Working voltage of IT8800 series includes 110V/220V. To prevent burnout, be sure to check whether the AC voltage setting of electronic load is consistent to the supply voltage before start-up!
- 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 electronic load is well grounded.
- Use electric wires of appropriate load. All loading wires should be capable of bearing maximum short-circuit of electronic load without overheating.
- 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.
- To prevent burnout, please pay special attention to positive and negative polarities of electronic load during connection!
- Do not use damaged equipment. Please check the housing before using the equipment. Check whether the instrument is subject to cracking or is lack of plastic. Do not operate the instrument in the environment with explosive gas, steam or dust.
- Observe all tags on the equipment before connection.
- Do not install alternative parts on the instrument or perform any unauthorized modification.
- Do not use the equipment when the removable cover is dismantled or loose.
- Please use the power adapter supplied by the manufacturer to avoid accidental injury.
- Do not use the equipment on the life support system or other equipment with safety requirements.

CAUTION

- If the equipment is not used in the manner specified by the manufacturer, its protection may be damaged.
- Always use dry cloth to clean the equipment housing. Do not clean the inside of the instrument.
- Do not block the air vent of the equipment.

Environmental conditions

The IT8800 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 0°C - 40°C
Operating humidity	20% - 80% (non condensing)
Storage temperature	-20°C - 70 °C

Altitude	≤2,000m
Pollution	Grade 2 pollution

Note

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

Regulation tag

CE	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 [1][2][3]

Reference Standards

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

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

Safety Standard

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

Contents

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 {NORMal|CARRier}

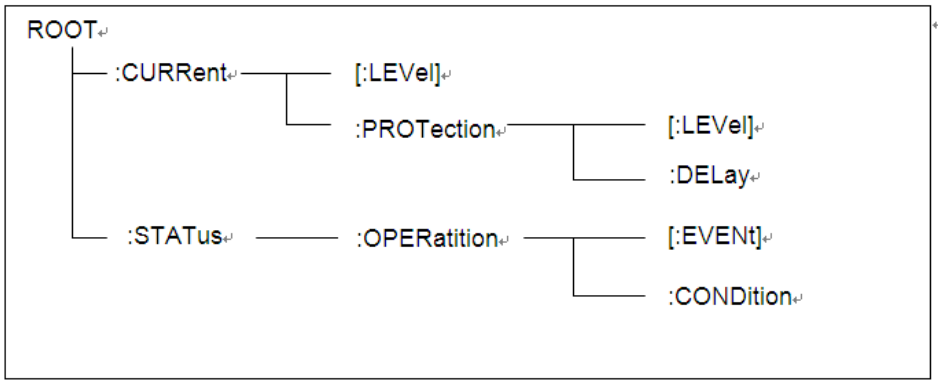
POLarity {NORMal|INVerted}

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

1.3 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 overallelectronic load functions, such as reset, status, and synchronization. All commoncommands consist of a three-letter mnemonic preceded by an asterisk: *RST *IDN? *SRE 8.
- Subsystem commands perform specific 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 subsyste 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. The long-form version are listed in the following chapters. 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 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 a asterisk (*) and occupy 3 character sizes, including one or more Parameter. Key words of a command and the first parameter are separated by a space. Semicolon (;) can separate several commands as follows:

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

Command terminator

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

Note

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

1.5 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...'

1.6 Response Data Type

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

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

Response messages

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

Sending a response message

Afte sending a query command, the response message is placed in the Output Queue. When the IT8800 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 IT8800 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 IT8800 Series what to send to the computer.

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

1. Send the appropriate query command(s) in a program message.1.
2. Address the IT8800 Series to talk.

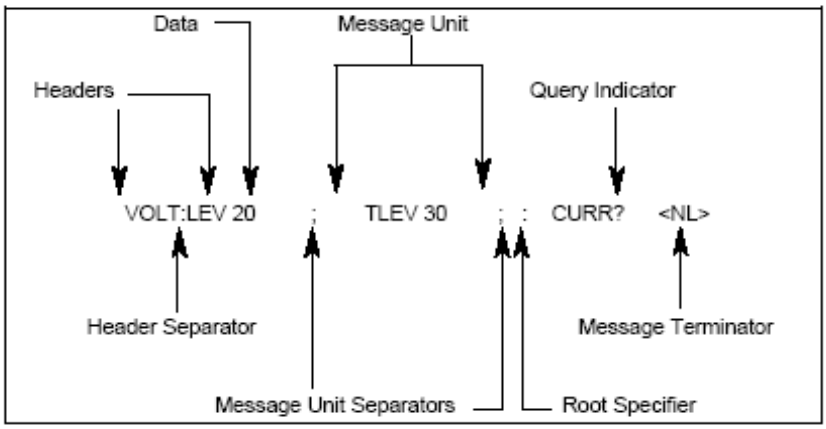
Rule 2. The complete response message must be received by the computer before another program message can be sent to the IT8800 Series.

1.7 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.

1.8 SCPI Command Complete

SCPI commands sent to the electronic load are processed either sequentially or in parallel. Sequential commands finish execution before a subsequent command begins. Parallel commands allow other commands to begin executing while the parallel command is still executing. Commands that affect trigger actions are among the parallel commands.

*WAI, *OPC, and *OPC:Common commands provide different ways of indicating when all transmitted commands, including any parallel ones, have completed their operations. Some practical considerations for using these commands are as follows:

*WAI: This prevents the electronic load from processing subsequent commands until all pending operations are completed.

*OPC?: This places a 1 in the Output Queue when all pending operations have completed. Because it requires your program to read the returned value before executing the next program statement, *OPC? can be used to cause the controller to wait for commands to complete before proceeding with its program.

*OPC: This sets the OPC status bit when all pending operations have completed. Since your program can read this status bit on an interrupt basis, *OPC allows subsequent commands to be executed.

Note

The trigger system must be in the Idle state in order for the status OPC bit to be true.

Therefore, as far as triggers are concerned, OPC is false whenever the trigger system is in the Initiated state.

Using Device Clear

You can send a device clear at any time to abort a SCPI command that may be hanging up the GPIB interface. The status registers, the error queue, and all configuration states are left unchanged when a device clear message is received. Device clear performs the following actions:

- The input and output buffers of the electronic load are cleared.
- The electronic load is prepared to accept a new command string.

The following statement shows how to send a device clear over the GPIB interface using GW BASIC:

CLEAR 705 IEEE-488 Device Clear

The following statement shows how to send a device clear over the GPIB interface using the GPIB command library for C or QuickBASIC.

IOCLEAR (705)

1.9 Remote Operation

IT8800 series electronic load is provided with three communication interfaces to communicate with a computer for selection, including RS232, USB, and GPIB.

GPIB function

All electronic load functions except setting the communication parameters are programmable over the GPIB. The IEEE 488.2 capabilities of the electronic load are described in the following table.

GPIB Capabilities	Response	Interface Function
Talker/Listener	All electronic load functions except for setting the communication paraments are programmable over the GPIB. The electronic load can send and receive messages over the GPIB. Status information is sent using a serial poll.	AH1, SH1,AH1, SH1, T6, L4T6, L4
Service Request	The electronic load sets the SRQ line true if there is an enabled service request condition.	SR1, SR1
Remote/Local	In local mode, the electronic load is controlled from the front panel but will also execute commands sent over the GPIB. If the electronic load is powered on in local mode, it remains in local mode until it receives a command over the GPIB. Once the electronic load is in remote mode the front panel REM indicator is on, all front panel keys (except**Shift**+**7**,**Local** and **Esc**) are disabled, and the display is in normal metering mode. Press **Local** on the front panel to returnto the local mode.	RL1
Device Trigger	The electronic load responds to the device triggers.	DT1
Group Execute Trigger	The electronic load responds to the group execute triggers.	GETGET
Device Clear	The electronic load responds to the Device Clear (**DCL**) and Selected Device Clear (**SDC**) interface commands. They command the electronic load to clear any activity that would prevent it from receiving and executing a new command (including ***WAI** and ***OPC?**). **DCL** and **SDC** do not change any programmed settings.	DCL,SDC

GPIB address

The electronic load operates from a GPIB address that is set from the front panel. To set the GPIB address, press the **Shift** + **System** on the front panel and enter the address using the keys. The address can be set from 0 to 30. The GPIB address is stored in non-volatile memory.

RS-232 function

The electronic load provides an RS-232 programming interface, which can be activated and set by pressing **Shift** + **System**on the front panel. All SCPI commands are available through RS-232 programming. When the RS-232 interface is selected, The EIA RS-232 Standard defines the interconnections between Data Terminal Equipment (DTE) and Data Communications Equipment (DCE).The electronic load is designed to be a DTE. It can be connected to another DTE such as a PC COM port through a null modem cable.

Note

The RS-232 settings in your program must match the settings specified in the front panel system menu. Press Shift + System to change the settings if they don't match.

You can break data transmissions by sending a ^C or ^X character string to the multimeter. This clears any pending operation and discards any pending output.

RS-232 data format

The RS-232 data is a 10-bit word with one start bit and one stop bit. The number of start and stop bits is not programmable. However, the following parity options are selectable after pressing **Shift** + **System**:

- EVEN: Seven data bits with even parity
- ODD: Seven data bits with odd parity
- NONE:Eight data bits without parity

Parity options are stored in non-volatile memory.

Baud rate

After pressing **Shift** + **System**,you can select one of the following baud rates: 4800/9600/19200/38400/57600/115200

The baud rate is stored in the non-volatile memory.

RS-232 flow control

The RS-232 interface supports the following flow control:

- **CTS/RTS**: the electronic load asserts its Request to Send (RTS) line to signal hold-off when its input buffer is almost full, and it interprets its Clear to Send (CTS) line as a hold-off signal from the controller.
- **XON/XOFF**: when the input queue of the electronic load becomes more than 3/4 full, the instrument issues an X_OFF command. The control program should respond to this and stop sending characters until The electronic load issues the X_ON, which it will do once its input buffer drops to below half-full. The electronic load recognizes X_ON and X_OFF sent from the controller. An X_OFF will cause the electronic load to stop outputting characters until it sees an X_ON.
- **NONE**: there is no flow control.

For each case, the electronic load will send a maximum of five characters after hold-off is asserted by the controller. The electronic load is capable of receiving as many as fifteen additional characters after it asserts hold-off.Flow control options are stored in the non-volatile memory.

RS-232 connections

The RS-232 serial port can be connected to the serial port of a controller (i.e., personal computer) using a straight through RS-232 cable ended with DB-9 connectors.

Note: Do not usea null modem cable.

If your computer uses a DB-25 connector for the RS-232 interface, you will need a cable or adapter with a DB-25 connector on one end and a DB-9 connector on the other, wired straight through.

The following figure and table show the pins for the connector.

The plug pins of RS232

Pin Number	Description
1	no connection
2	TXD:transmit data
3	RXD:receive data
4	no connection
5	GND:signal ground
6	no connection
7	CTS:clear to send
8	RTS:ready to send
9	no connection

RS-232 troubleshooting

If you are having trouble communicating over the RS-232 interface, check the following:

- The computer and the electronic load must be configured for the same baud rate, parity, number of data bits, and flow control options. Note that the electronic load is configured for 1 start bit and 1 stop bit (these values are fixed).
- The correct interface cables or adapters must be used, as described under RS-232 Connector. Note that even if the cable has the proper connectors for your system, the internal wiring may be incorrect.
- The interface cable must be connected to the correct serial port on your computer (COM1, COM2,etc.).

USB-TMC Capabilities of the Electronic Load

All electronic load functions are programmable over the USB.

The USB488 interface capabilities of the electronic load are described as follows:

- The interface is a 488.2 USB488 interface.
- The interface accepts REN_CONTROL, GO_TO_LOCAL, and LOCAL_LOCKOUT requests.
- The interface accepts the MsgID = TRIGGER USBTMC command message and forwards TRIGGER requests to the Function Layer.

The USB488 device capabilities of the electronic load are described follow:

- The device understands all mandatory SCPI commands.
- The device is SR1 capable
- The device is RL1 capable
- The device is DT1 capable

1.10 Queue

The IT8800 Series uses two queues, which are first-in, first-out (FIFO)

registers:

- Output Queue - used to hold reading and response messages
- Error Queue - used to hold error and status messages

The IT8800 Series status model shows how the two queues are structured with the other registers.

Output queue

The output queue holds data that are related to the normal operation of the instrument. For example, when a query command is sent, the response message is placed on the output queue.

When data is placed in the output queue, the Message Available (MAV) bit in the status byte register sets. A data message is cleared from the output queue when it is read. The output queue is considered cleared when it is empty. An empty output queue clears the MAV bit in the status byte register.

You can read a message from the output queue after a query is sent.

Error queue

The error queue holds error and status messages. When an error or status event occurs, a message that defines the error/status is placed in the error queue. This queue holds up to 32 messages.

When a message is placed in the error queue, the Error Available (EAV) bit in the status byte register is set. An error message is cleared from the error/status queue when it is read. The error queue is considered cleared when it is empty. An empty error queue clears the EAV bit in the status byte register. Read an error message from the error queue by sending :SYSTem:ERRor?command.

1.11 Status byte and service request (SRQ)

Service request is controlled by two 8-bit registers: the status byte register and the service request enable register.

Status byte register

The summary messages from the status registers and queues are used to set or clear the appropriate bits (B2, B3, B4, B5, and B7) of the status byte register.These bits do not latch, and their states (0 or 1) are solely dependent on the summary messages (0 or 1).For example, if the Standard event status register is read, its register is cleared.As a result, its summary message will reset to 0, which in turn will clear the ESB bit in the status byte register.Bit B6 in the status byte register is called the MSS bit.

The Master Summary Status (MSS) bit, sent in response to the *STB?indicates the enable status of the set bit.The Request for Service (RQS) bit, sent in response to a serial poll, indicates which device was requesting service by pulling on the SRQ line.

For a description of the other bits in the status byte register, see STB?

When reading the status byte register using the *STB? command, bit B6 is called the MSS bit.None of the bits in the status byte register are cleared when using the *STB? command to read them.

The IEEE-488.1 standard has a serial poll sequence that also reads the status byte register and is better suited to detect a service request (SRQ).When using the serial poll, bit B6 is called the RQS bit.Serial polling causes bit B6 (RQS) to reset.Serial polling is discussed in more detail later.

Any of the following operations clear all bits of the status byte register:

- Circulation power
- Sending the *CLS command

Note: The MAV bit may or may not be cleared.

Service request enable register

This register is programmed by you and serves as a mask for the status summary message bits (B2, B3, B4, B5, and B7) of the status byte register.When masked, a set summary bit in the status byte register cannot set bit B6 (MSS/RQS) of the status byte register. Conversely, when unmasked, a set summary bit in the status byte register sets bit B6.

A status summary message bit in the status byte register is masked when the corresponding bit in the service request enable register is cleared.When the masked summary bit in the status byte register sets, it is ANDed with the corresponding cleared bit in the service request enable register.The logic “1” output of the AND gate is applied to the input of the OR gate and, thus, sets the MSS/RQS bit in he status byte register.

The individual bits of the service request enable register can be set or cleared by using the following common command:

*SRE <NRf>*SRE <NRf>

To read the service request enable register, use the *SRE? query command. The service request enable register clears when power is cycled or a parameter (n) value of zero is sent with the *SRE command *SRE 0).

1.12 Serial poll and SRQ

Any enabled event summary bit that goes from 0 to 1 will set RQS and generate a service request (SRQ). In your test program, you can periodically read the status byte register to check if a service request (SRQ) has occurred and what caused it. If an SRQ occurs, the program can, for example, branch to an appropriate subroutine that will service the request. Typically, service requests (SRQs) are managed by the serial poll sequence of the electronic load. If an SRQ does not occur, bit B6 (RQS) of the status byte register will remain cleared and the program will simply proceed normally after the serial poll is performed. If an SRQ does occur, bit B6 of the status byte register will set and the program can branch to a service subroutine when the SRQ is detected by the serial poll.

The serial poll automatically resets RQS of the status byte register. This allows subsequent serial polls to monitor bit B6 for an SRQ occurrence generated by other event types. After a serial poll, the same event can cause another SRQ, even if the event register that caused the first SRQ has not been cleared.

A serial poll clears RQS but does not clear MSS. The MSS bit stays set until all status byte event summary bits are cleared.

1.13 Trigger Model (GPIB Operation)

This section describes how the electronic load operates over the GPIB bus. It is called the trigger model because operation is controlled by SCPI commands from the Trigger subsystem. Key SCPI commands are included in the trigger model.

Trigger Model Operation

Once the instrument is taken out of idle state, operation proceeds through the trigger model down to the device action.

A control source is used to hold up operation until the programmed event

occurs. The control source options are explained as follows:

- HOLD: only the FORCE:TRIG command will generate a trigger in HOLD mode. All other trigger commands are ignored.
- MANual: event detection is ended by pressing the TRIG key.
- TIMer: this generates triggers that are in synchronization with the electronic load's internal oscillator as the trigger source. The internal oscillator begins running as soon as this command is executed. Send TRIG:TIM to program the oscillator period.
- EXTernal: event detection is ended when an input trigger via the TRIGGER LINK connector is received by the electronic load.
- BUS: event detection is ended when a bus trigger (GET or *TRG) is received by the electronic load.

Chapter2 SCPI Register

2.1 Status Register

You can use status register programming to determine the operating condition of the electronic load at any time.For example, you may program the electronic load to generate an interrupt (assert SRQ) when an event such as a current protection occurs.

The Standard Event, Status Byte, Service Request Enable registers, and the Output Queue perform standard GPIB functions as defined in the IEEE 488.2 Standard Digital Interface for Programmable Instrumentation. The Operation Status and Questionable Status registers implement functions that are specific to the electronic load.

The following table explains the status signals.

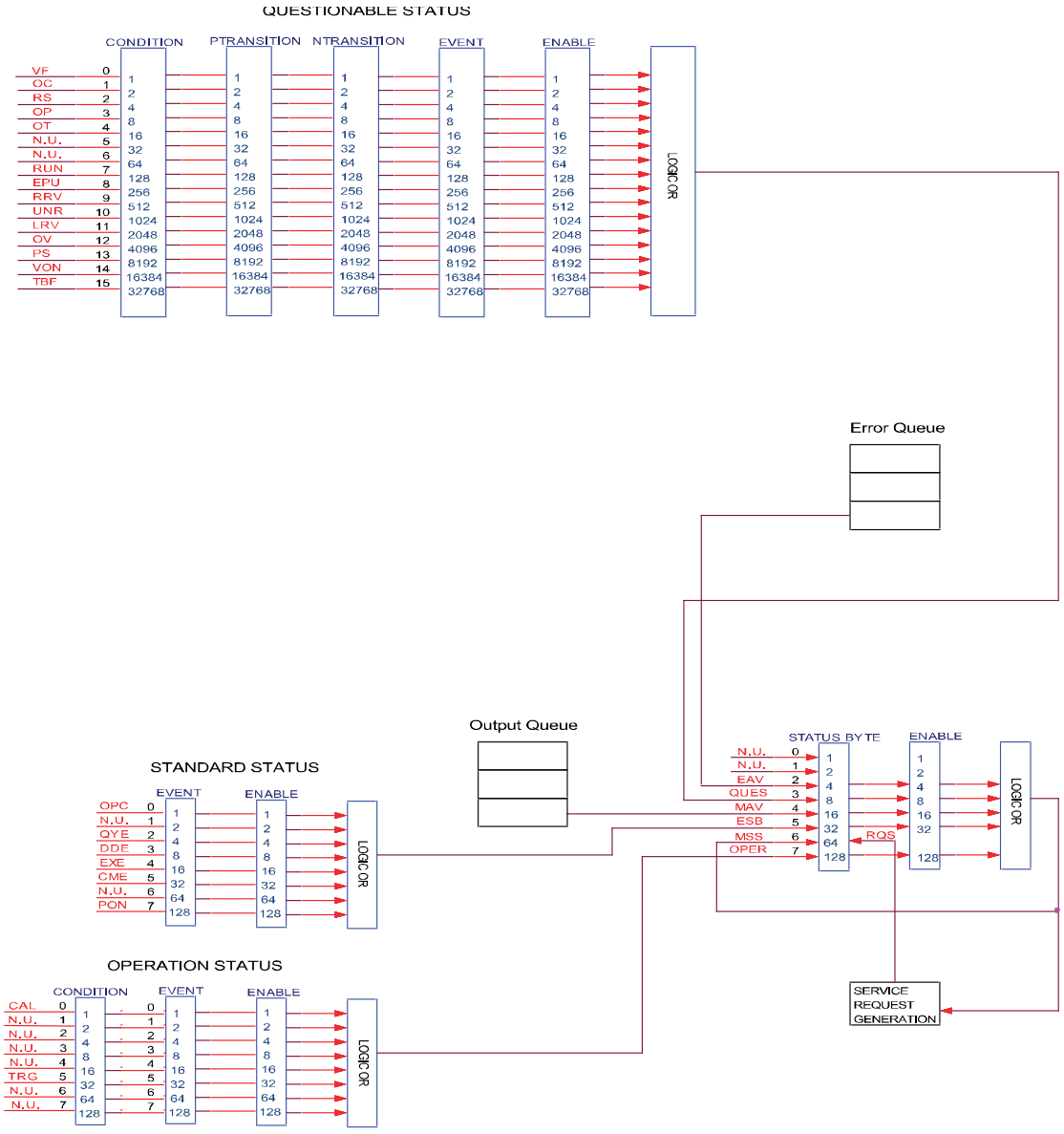
Bit	Signal	Description
Operation status group		
6	CAL	Calibrating: The electronic load iscalculated a new calibration constant.
5	TRG	Waiting: The electronic load is waiting for a trigger
Channel status group/Questionable status group		
0	VF	Voltage Fault. Either an overvoltage or a reverse voltage has 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.
1	OC	Over current: An over-current condition hasoccurred. This occurs if the current exceeds 110% of the rated current or if it exceeds the user-programmed current protection level. Removing the over-current condition clears the bit. If the condition persists beyond the user programmable delay time, PS bit is also set and the input is turned off. Both bits remain set until the condition is removed and PROT:CLE is programmed.
2	RS	Remote Sense: When the real pannel sense is connected, this bit is true or else false.
3	OP	Overpower: An overpower condition has occurred. This occurs if the unit exceeds the max power or it excees the user-programmed power protection level, removing the overpower condition clears the bit. If the condition persists beyond the user programmable delay time, PS bit is also set and the input is turned off. Both bits remain set until the condition is removed and PROT:CLE is programmed.
4	OT	Over temperature: An over-temperature condition has occurred, both this bit and bit PS are set and the input is turned off. Both bits remain set until the unit is cooled down and PROT:CLE is programmed.
7	RUN	List run or stop status when list is runnig, this bit is true else false.
8	EPU	Extended Power Unavailable: This bit is not used.
9	RRV	Remote Reverse Voltage: A reverse voltage condition has occurred on the sense terminals, both this bit and VF bit are set. Removing the reverse voltage clears this bit but does not clear VF bit. VF Bit remains set until PROT:CLE is programmed.
10	UNR	Unregulated: The input is unregulated, when the input is regulated the bit is cleared.
11	LRV	Local Reverse Voltage: A reverse voltage condition has occurred on the input terminals, both this bit and VF bit are set. Removing the reverse voltage clears this bit but does not clear VF bit. VF bit remains set until PROT:CLE is programmed.
12	OV	Over voltage: An over voltage condition has occurred, both this bit and VF bit0 are set and the load are turned off. Both bits remain set until the condition is removed and PROT:CLE is programmed.
13	PS	Protection Shutdown: The protection shutdown circuit has tripped because of an Over-current, over-power, or over-temperature condition. The bit remains set until PROT:CLE is programmed.

14	VON	Voltage of sink current on: When the voltage of input exceeds the user-programmed Von level, this bit is true else false.
15	TBF	Trace Buffer Full.

Standard event status group		
0	OPC	Operation Complete: The load has completed all pending operations, *OPC must be programmed for this bit to be set when pending operations are complete.
2	QYE	Query Error: 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.
3	DDE	Device-Dependent Error: Device-Dependent Error. Memory was lost or self test failed. Errors in the range of -399 through -300, 0 through 99 and 100 through 32767 can set this bit.
4	EXE	Execution Error: 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.
5	CME	Command Error: A syntax or semantic error has occurred or the load received a <get> within a program message. Errors in the range of 200 through 100 can set this bit.
7	PON	Power-On: The unit has been turned off and then on since this bit was last read.

Status and service request enable register		
0	CSUM	Channel Summary.: Indicates whether an enabled channel event has occurred.
2	EAV	Error Available Summary: Indicates if the Error Queue contains data.
3	QUES	Questionable Status Summary: Indicates if an enabled questionable event has occurred.
4	MAV	Message Available Summary: Indicates if the Output Queue contains data.
5	ESB	Event Status Summary: Indicates if an enabled standard event has occurred.
6	RQS/ MSS	Request Service: During a serial poll, RQS is returned and cleared. Master Status Summary: For an *STB? query, MSS is returned without being cleared.
7	OPER	Operation Status Summary: Indicates if an operation event has occurred.

The following figure shows the status register structure of the electronic load.



2.2 Condition register

As you can see from the figure above, channel statusregister and operation status register sets have a condition register, A condition register is a real-time, read-only register that constantly updates to reflect the current operating conditions of the instrument.

You can see the :CONDition? command in the STATus Subsystem to read the condition registers.

2.3 Event register

Each status register set has an event register. An event register is a latched, read-only register whose bits are set by the corresponding condition register. Once a bit in an event register is set, it remains set (latched) until the register is cleared by a specific clearing operation. The bits of an event register are logically ANDed with the bits of the corresponding enable register and applied to an OR gate. The output of the OR gate is applied to the status byte register.Send the *ESR? command to read the standard event register.All other

event registers can be read by sending the :EVENt? query command.

An event register is cleared when it is read. The following operations clear all event registers:

- Cycling power
- Sending *CLS

2.4 Enable register

Each status register set has an enable register. An enable register is programmed by you and serves as a mask for the corresponding event register. An event bit is masked when the corresponding bit in the enable register is cleared (0). When masked, a set bit in an event register cannot set a bit in the status byte register (1 AND 0 = 0).

To use the status byte register to detect events (i.e., serial poll), you must unmask the events by setting the appropriate bits of the enable registers.

To program and query the Standard Event Status Register, use the *ESE and *ESE?.

All other enable registers are programmed and queried using the :ENABle and :ENABLe? Command.

An enable register is not cleared when it is read. The following operations affect the enable registers:

- Circulationpower: Clear all the enable register.
- STATus:PREset clears the following enable registers.
  - ✧ Operation event enable register
  - ✧ Questionable event enable register
  - ✧ Channel overview Event enable register
- *ESE 0 clears the standard event status enable register.

Chapter3 Programming Examples

This chapter displays the programming examples to remotely control IT8800 load using SCPI commands.

Note

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

Example 1: Identifying the Load in Use

You can verify whether you are communicating with the right IT8800 load.

To query the identification of the load, send the command:

```
*IDN?
```

To check the power supply error queue, send the command:

```
SYST:ERR?
```

Example 2: Common input commands

```
SYSTem:REMote
FUNCtion CURRent
CURRent 3
FUNCtion VOLTage
VOLTage 10
FUNCtion POWer
POWer 10
INPut ON
MEASure:VOLTage?
MEASure:CURRent?
MEASure:POWer?
```

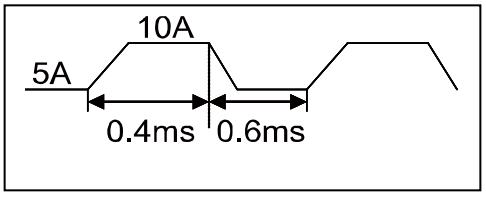
Example 3: Programming Transients

Transient operation is used to synchronize input changes with internal or external trigger signals, and simulate loading conditions with precise control of timing, duration, and slew. The following transient modes can be generated:

Continuous Transients

In continuous operation, a repetitive pulse train switches between two load levels. The rate at which the level changes is determined by the slew rate (see slew rate descriptions for CV, CR, or CV mode as applicable). In addition, Use the following commands to program continuous transients:

```
CURRent:TRANsient:MODE CONTinuous
CURRent:TRANsient:ALEVel 5
CURRent:TRANsient:AWIDth 0.6mS
CURRent:TRANsient:BLEVel 10
CURRent:TRANsient:BWIDth 0.4mS
TRANsient ON
```



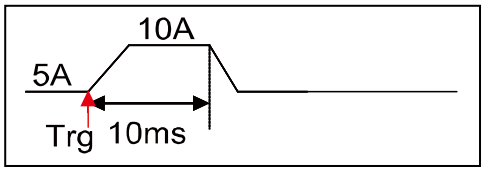
```
INPut ON
TRIGger:IMMediate
```

Pulse Transients

Pulsed transient operation generates a load change that returns to level B state after some time period.

Use the following commands to program pulsed transients:

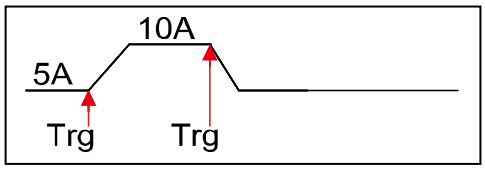
```
CURRent:TRANsient:MODE PULSe
CURRent:TRANsient:ALEVel 10
CURRent:TRANsient:BLEVel 5
CURRent:TRANsient:AWIDth 10mS
TRANsient ON
INPut ON
TRIGger:IMMediate
```



Toggled Transients

Under toggle mode, after enabling dynamic test operation, the load will be switched continuously between A value and B value after receipt of every trigger signal. Use the following commands to program toggled transients:

```
CURRent:TRANsient:MODE TOGGle
CURRent:TRANsient:ALEVel 10
CURRent:TRANsient:BLEVel 5
TRANsient ON
INPut ON
TRIGger:IMMediate
```



Example 4: Programming Lists

The following procedure shows how to generate a simple 4-step list of current changes.

```
FUNC CURRent
LIST:RANGe   40
LIST:COUNt   10000
LIST:STEP 4
LIST:LEVel   1, 5
LIST:SLEW  1, 1
LIST:WIDth 1, 10ms
LIST:LEVel   2, 10
LIST:SLEW  2, 1
LIST:WIDth 2, 10ms
LIST:LEVel   3, 20
LIST:SLEW  3, 1
LIST:WIDth 3, 10ms
LIST:LEVel   4, 15
LIST:SLEW  4, 1
LIST:WIDth 4, 10ms
FUNCtion:MODE   LIST
```



```
TRIGger:IMMediate
```

Example 5: Trace function

The following is an example of how to use the commands of the Trace subsystem:

```
TRACe:CLEar                    //Clear the buffer of readings.
TRACe:POINts 2000              //Specify the size of the buffer.
TRACe:FEED TWO    //Select the source of readings to be placed in the buffer.
TRACe:FEED:CONTrol NEXT      //Select the buffer control.
TRACe:TIMer 0.00002            //Select the interval for timer.
TRACe:DELay 1                  //Select the delay time for trigger in buffer.
TRIGger                 //Trigger the instrument to enter data storage status.
TRACe:DATA?         //Read the data stored in the buffer to the PC interface.
```

Chapter4 IEEE488.2 Commands

Common commands begin with an * and consist of three letters (command) or three letters and a ?(query). They are defined by the IEEE 488.2 standard to perform common interface functions. Common commands and queries are categorized under System, Status, or Trigger functions and are listed at the end of this chapter.

Common Commands

Common commands begin with an * and consist of three letters (command) IEEE 488.2 standard to perform some common interface functions. The electronic loads respond to the required common commands that control status reporting, synchronization, and internal operations. The electronic loads also respond to optional common commands that control triggers, power-on conditions, and stored operating parameters.

Common commands and queries are listed alphabetically. If a command has a corresponding query that simply returns the data or status specified by the command, then both command and query are included under the explanation for the command. If a query does not have a corresponding command or is functionally different from the command, then the query is listed separately. The description for each common command or query specifies any status registers affected. Refer to chapter “Programming the Status Registers”, which explains how to read specific register bits and use the information that they return.

Menmonic	Name	Description
*CLS	Clear status	Clear all event registers and Error Queue
*ESE <NRf>	Event enable command	Edit Standard Event Enable Register.
*ESE?	Event enable query	Read Standard Event Enable Register.
*ESR?	Event status query	Read Standard Event Status Register and clear it
*IDN?	Identification query	Return the manufacture, model number, serial number and the firmware revisions of the instrument.
*OPC	Operation complete command	Set the Operation Complete bit in the Standard Event Status Register after all pending commands have been executed.
*OPC?	Operation complete query	Places an ASCII “1” into the output queue when all pending selected device operations have been completed.
*RCL <NRf>	Recall Command	Returns the Load the setup configuration stored in the specified memory location.
*RST	Reset Command	Returned the Load to the *RST default conditions
*SAV <NRf>	Save Command	Saves the current setup to the specified memory location.
*SRE <NRf>	Service request enable command	Programs the Service Request Enable register.
*SRE?	Service request enable query	Read Service Request Enable register
*STB?	Read status byte query	Read Status Byte Register
*TRG	Trigger Command	Send a trigger to Load.
*TST?	Self-test query	Performs a self-test and returns the result.
*WAI	Wait to continue command	Wait until all previous commands are executed.

*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 OR 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
Power-on Value	see *PSC
Examples	*ESE 129
Query Syntax	*ESE?
Returned Parameters	<NR1>
Related Commands	***ESR?** *PSC *STB?

*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 comma.

Query Syntax	*IDN?
Parameters	None
Examples	ITECH Ltd, IT88XX, XXXXXXXXXXXXXXXXX, 1.21-1.28
Returned Parameters	<AARD> **Field** **Information** ITECH Ltd Technologies Manufacturer IT88XX Model XXXX….. Serial Number 1.21-1.28 Firmware Revision

*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

*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, operate 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, operate 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

*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.

NOTE: *SAV does not save the programmed trigger values ([SOURce:]CURRent:TRIGGer, [SOURce:]RESistance:TRIGGer, [SOURce:]VOLTage:TRIGGer). Programming an *RCL or a *RST command causes the triggered settings to revert to their [IMMediate] settings.

Command Syntax	*SAV <NRf>
Parameters	0 to 100
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 OR 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

*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	<NR1> (register value)
Related Commands	*SRE *ESR *ESE

*TRG

This command generates a trigger to any system that has BUS selected as its source (for example, TRIG:SOUR BUS). The command has the same affect as the Group Execute Trigger (<GET>) command.

Command Syntax	*TRG
Parameters	None
Related Commands	ABOR INIT TRIG:IMM

*TST?

This command causes the electronic load to do a self-test and report any error.

Command Syntax	TST?
Parameters	None
Returned Parameters	<NR1> 0 indicates the electronic load has passed selftest. Non-zero indicates an error code(see appendix C）

*WAI

This command instructs the electronic load not to process any further commands until all pending operations are completed. Pending operations are complete when:

All commands sent before *WAI 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 input voltage or state, relays, and trigger actions are overlapped with subsequent commands sent to the electronic load. The *WAI command prevents subsequent commands from being executed before any overlapped commands have been completed.

All triggered actions are completed and the trigger system returns to the Idle state. *WAI can be aborted only by sending the electronic load a GPIB DCL (Device Clear) command.

Command Syntax	WAI?
Parameters	None
Related Commands	*OPC

Chapter5 Essential Commands

STATus Subsystem

Those commands configure the status registers of the electronic load.

STATus:QUEStionable?

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> (register value)
Related Commands	*CLS

STATus:QUEStionable:ENABle

The command set or read quest enable register. The register can make the special bit of the quest event register set the query status bit register overview bit (QUES) of the status byte register. The bit (bit 3) is the logic OR of all query event register, is enabled by the quest status enable register.

Command Syntax	STATus:QUEStionable:ENABle <NR1>
Parameters	0 to 65535
Default value	0
Examples	STAT:QUES:ENAB 32 STAT:QUES:ENAB 1
Query Syntax	**STATus:QUEStionable:ENABle?**
Returned Parameters	<NR1> (register value)
Related Commands	STAT:QUES?

STATus:QUEStionable:PTRansition

The command set or read queries the value of the positive change enable register, when the query register bit changes from 0 to 1, and the corresponding bit of positive change enable register is 1, then the corresponding bit of the quest event register is 1.

Command Syntax	STATus:QUEStionable:**PTRansition** <NR1>
Parameters	0 to 65535
Default value	0
Examples	STAT:QUES:**PTRansition** 32 STAT:QUES:**PTRansition** 1
Query Syntax	**STATus:QUEStionable:PTRansition?**
Returned Parameters	<NR1> (register value)
Related Commands	STAT:QUES?

STATus:QUEStionable:NTRansition

The command set or read queries the value of the negative change enable register, when the query register bit changes from 1 to 0, and the corresponding bit of positive change enable register is 1, then the corresponding bit of the quest event register is 1.

Command Syntax	STATus:QUEStionable:**NTRansition** <NR1>
Parameters	0 to 65535
Default value	0
Examples	STAT:QUES:**NTRansition** 32 STAT:QUES:**NTRansition** 1
Query Syntax	**STATus:QUEStionable:NTRansition?**
Returned Parameters	<NR1> (register value)
Related Commands	STAT:QUES?

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?
Returned Parameters	<NR1> (register value)
Related Commands	STAT:OPER:COND?

STATus:OPERation?

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> (register value)
Related Commands	*CLS

STATus:OPERation:ENABle

The command and its query set and read the parameters 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	0 to 255
Default value	0
Examples	STAT:OPER:ENAB 32 STAT:OPER:ENAB 1

Query Syntax	**STATus:OPERation:ENABle?**
Returned Parameters	<NR1> (register value)
Related Parameters	STAT:OPER?

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> (register value)
Related Commands	STAT:QUES:COND?

STATus:PRESet

After executing this command, SCPI event register will get the following affection: All the bits of below register will be cleared：

- Quest event enable register
- Operation event enable register

Note that the register not list above will not be affected by the command.

Command Syntax	STATus:PRESet
Parameters	None
Examples	STAT:PRES

Chapter6 System Commands

System commands controls the system-level function of the load, and those function will not affect on the input control or test function.

SYSTem:PRESet

This command gets the load to a state that is suitable for panel operation.

Command Syntax	SYSTem:PRESet
Parameters	None

SYSTem:POSetup

This command is used to select the default value when the load power on. If RST is selected, then the instrument powers up to the *RST default conditions. When the SAV0 parameter is selected, the instrument powers-on to the setup that is saved in the specified location using the *SAV command.

Command Syntax	SYSTem:POSetup <CRD>
Parameters	RST | SAV0
***RST value**	RST
Examples	SYST:POS RST
Query Syntax	**SYSTem:POSetup?**
Returned Parameters	<CRD>
Related Commands	*RST *SAV

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
Examples	SYST:VERS?
Returned Parameters	<NR2>

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 occur, 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	SYST:ERR?

SYSTem:CLEar

This action is used to clear the error sequence information.

Command Syntax	SYSTem:CLEar
Parameters	None
Examples	SYST:CLE
Related Commands	SYST:ERR?

SYSTem:LOCal

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

Command Syntax	SYSTem:LOCal
Parameters	None
Examples	SYST:LOC
Related Commands	SYST:REM SYST:RWL

SYSTem:REMote

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

Command Syntax	SYSTem:REMote
Parameters	None
Examples	SYST:REM
Related Commands	SYST:LOC SYST:RWL

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

SYSTem:KEY

This command is used to simulate the key pressing.

Command Syntax	**SYSTem:KEY** <NR1>
Parameters	0 to 255

Default value	0
Examples	SYST:KEY 1
Query Syntax	**SYSTem:KEY?**
Returned Parameters	<NR1> (register value)

DISPlay[:WINDow]:MODE

This command is used to set the display mode of the VFD screen. NORMAL indicates display normally, TEXT indicates text display mode.

Command Syntax	DISPlay[:WINDow]:MODE <CRD>
Parameters	NORMal | TEXT
***RST value**	NORMal
Examples	DISP:MODE TEXT
Query Syntax	**DISPlay[:WINDow]:MODE?**
ReturnedParameters	<CRD>
Related Commands	DISP:TEXT

DISPlay[:WINDow]:TEXT

When VFD is in TEXT display mode, this command is used to display the string already set.

Command Syntax	DISPlay[:WINDow]:TEXT <NR1>, <SRD>
Examples	DISP:TEXT 0, “HELLO!”
Related Commands	DISP:MODE

Chapter7 Measure Commands

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

FETCh:VOLTage[:DC]?

This command returns the last measured input voltage stored in the communications buffer of the electronic load. A new measurement is not initiated by this command.

Group	Measurement
Syntax	FETCh:VOLTage[:DC]?
Returns	<NR2> is the measured input voltage in volts.
Example	FETC:VOLT? might return 5.0011, which would be the measured voltage across the electronic load inputs in volts.

FETCh:VOLTage:MAX?

This command returns the last measured input maximum voltage stored in the communications buffer of the electronic load. A new measurement is not initiated by this command.

Group	Measurement
Syntax	FETCh:VOLTage:MAX?
Returns	<NR2> is the measured input maximum voltage in volts.
Example	FETC:VOLT:MAX? might return 100.0011, which would be the measured maximum voltage across the electronic load inputs in volts.

FETCh:VOLTage:MIN?

This command returns the last measured input minimum voltage stored in the communications buffer of the electronic load. A new measurement is not initiated by this command.

Group	Measurement
Syntax	FETCh:VOLTage:MIN?
Returns	<NR2> is the measured input minimum voltage in volts.

Example FETC:VOLT:MIN? might return 1.0011, which would be the measured minimum voltage across the electronic load inputs in volts.

FETCh:CURRent[:DC]?

This command returns the last measured input current stored in the communications buffer of the electronic load. A new measurement is not initiated by this command.

Group Measurement

Syntax FETCh:CURRent[:DC]?

Returns <NR2> is the measured input current in amperes.

Example FETC:CURR? might return 3.001, which would be the measured current across the electronic load inputs in amperes.

FETCh:CURRent:MAX?

This command returns the last measured input maximum current stored in the communications buffer of the electronic load. A new measurement is not initiated by this command.

Group Measurement

Syntax FETCh:CURRent:MAX?

Returns <NR2> is the measured input maximum current in amperes.

Example FETC:CURR:MAX? might return 40.001, which would be the measured maximum current across the electronic load inputs in amperes.

FETCh:CURRent:MIN?

This command returns the last measured input minimum current stored in the communications buffer of the electronic load. A new measurement is not initiated by this command.

Group Measurement

Syntax FETCh:CURRent:MIN?

Returns <NR2> is the measured input minimum current in amperes.

Example FETC:CURR:MIN? might return 1.001, which would be the measured minimum current across the electronic load inputs in amperes.

FETCh:POWer[:DC]?

This command returns the last measured input power stored in the communications buffer of the electronic load. A new measurement is not initiated by this command.

Group Measurement

Syntax FETch:POWer[:DC]?

Returns <NR2> is the measured input power in watt.

Example FETC:POW? might return 5.01, which would be the measured power across the electronic load inputs in watt.

FETCh:CAPability?

This command returns the last measured discharging capability stored in the communications buffer of the electronic load. A new measurement is not initiated by this command.

Group Measurement

Syntax FETch:CAPability?

Returns <NR2> is the measured input discharging capability in ampere-hour.

Example FETC:CAP? might return 5.011, which would be the measured discharging capability of the electronic load in ampere-hour.

FETCh:TIME?

This command returns the last measured discharging time stored in the communications buffer of the electronic load. A new measurement is not initiated by this command.

Group Measurement

Syntax FETch:TIME?

Returns <NR2> is the measured discharging time in second.

Example FETC:TIME? might return 5.1, which would be the measured discharging time of the electronic load in second.

MEASure:VOLTage[:DC]?

This command initiates and executes a new voltage measurement, and returns the measured input voltage of the electronic load.

Group Measurement

Syntax MEASure:VOLTage[:DC]?

Returns <NR2> is the measured input voltage in volts.

Example MEAS:VOLT? might return 5.0011, which would be the measured voltage across the electronic load inputs in volts.

MEASure:VOLTage:MAX?

This command initiates and executes a new maximum voltage measurement, and returns the measured input maximum voltage of the electronic load.

Group Measurement

Syntax MEASure:VOLTage:MAX?

Returns <NR2> is the measured input maximum voltage in volts.

Example MEAS:VOLT:MAX? might return 100.0011, which would be the measured maximum voltage across the electronic load inputs in volts.

MEASure:VOLTage:MIN?

This command initiates and executes a new minimum voltage measurement, and returns the measured input minimum voltage of the electronic load.

Group Measurement

Syntax MEASure:VOLTage:MIN?

Returns <NR2> is the measured input minimum voltage in volts.

Example MEAS:VOLT:MIN? might return 1.0011, which would be the measured minimum voltage across the electronic load inputs in volts.

MEASure:CURRent[:DC]?

This command initiates and executes a new current measurement, and returns the measured input current of the electronic load.

Group	Measurement
Syntax	MEASure:CURRent[:DC]?
Returns	<NR2> is the measured input current in amperes.
Example	MEAS:CURR? might return 3.001, which would be the measured current across the electronic load inputs in amperes.

MEASure:CURRent:MAX?

This command initiates and executes a new maximum current measurement, and returns the measured input maximum current of the electronic load.

Group	Measurement
Syntax	MEASure:CURRent:MAX?
Returns	<NR2>is the measured input maximum current in amperes.
Example	MEAS:CURR:MAX? might return 40.001, which would be the measured maximum current across the electronic load inputs in amperes.

MEASure:CURRent:MIN?

This command initiates and executes a new minimum current measurement, and returns the measured input minimum current of the electronic load.

Group	Measurement
Syntax	MEASure:CURRent:MIN?
Returns	<NR2> is the measured input minimum current in amperes.
Example	MEAS:CURR:MIN? might return 1.001, which would be the measured minimum current across the electronic load inputs in amperes.

MEASure:POWer[:DC]?

This command initiates and executes a new power measurement, and returns the measured input power of the electronic load.

Group	Measurement
Syntax	MEASure:POWer[:DC]?

Returns <NR2> is the measured input power in watt.

Example MEAS:POW? might return 5.01, which would be the measured power across the electronic load inputs in watt.

MEASure:CAPability?

This command initiates and executes a new discharging capability measurement, and returns the measured input discharging capability of the electronic load.

Group Measurement

Syntax MEASure:CAPability?

Returns <NR2>is the measured input discharging capability in ampere-hour.

Example MEAS:CAP? might return 5.011, which would be the measured discharging capability of the electronic load in ampere-hour.

MEASure:TIME?

This command initiates and executes a new discharging time measurement, and returns the measured input discharging time of the electronic load.

Group Measurement

Syntax MEASure:TIME?

Returns <NR2> is the measured discharging time in second.

Example MEAS:TIME? might return 5.1, which would be the measured discharging time of the electronic load in second.

Chapter8 Trigger Subsystem

Trigger system consists of commands and configuration trigger mode subsystems.

TRIGger

When the trigger system has been initiated, this command generates a trigger signal regardless of the selected trigger source.

Command Syntax	TRIGger[:IMMediate]
Parameters	None
Examples	TRIG
Related Parameters	TRIG:SOUR TRIG:TIM

TRIGger:SOURce

This command selects the trigger source.

BUS Accepts a GPIB <GET> signal or a *TRG command as the trigger source. This selection guarantees that all previous commands are complete before the trigger occur.

EXTernal Selects the electronic load's trigger input as the trigger source. This trigger is processed as soon as it is received.

HOLD Only TRIG:IMM command will generate a HOLD mode trigger. All the other trigger commands are not considered.

MANUal The event occurs when the Trig key is pressed.

TIMer The trigger will synchronize with the crystal within the electronic load. Once the command is executed a synchronous oscillator will start running. Use the TRIG: TIM to edit crystal cycle.

Command Syntax	TRIGger:SOURce <CRD>
Parameters	BUS | EXTernal | HOLD | MANUal | TIMer
*RST value	MANUal
Examples	TRIG:SOUR BUS TRIG:SOUR EXT
Query Syntax	**TRIGger:SOURce?**
Returned Parameters	<CRD>
Related Parameters	ABOR TRIG TRIG:DEL

TRIGger:TIMer

This command specifies the period of the triggers generated by the internal trigger generator.

Command Syntax	TRIGger:TIMer <NRf+>
Parameters	0.01□1o 999.99s | MINimum | MAXimum | DEFault
Units	*seconds*
*RST value	0.01
Example	TRIG:TIM 0.25 TRIG:TIM MAX
Query Syntax	**TRIGger:TIMer?** [MINimum | MAXimum |

	DEFault]
Returned Parameter	<NR3>
Relataed Commands	ABOR TRIG TRIG:SOUR TRIG:DEL

Chapter9 Trace Subsystem

The commands in this subsystem are used to configure and control data storage into the buffer.

TRACe:CLEar

This action command is used to clear the buffer of readings. If you do not clear the buffer, a subsequent store will overwrite the old readings. If the subsequent store is aborted before the buffer becomes full, you could end up with some “old” readings still in the buffer.

Command Syntax TRACe:CLEar
Parameters None
Example STAT:PRES

TRACe:FREE?

This command is used to read the status of storage memory. After sending this command and addressing the electronic to talk, two values separated by commas are sent to the computer. The first value indicates how many bytes of memory are available, and the second value indicates how many bytes are reserved to store readings.

Query Syntax TRACe:FREE?
Returned Parameters <NR1>, <NR1>
Examples TRAC:FREE?

TRACe:POINts

This command is used to specify the size of the buffer.

Command Syntax TRACer:POINts <NRf+>
Parameters 2 to 1000| MINimum | MAXimum | DEFault
***RST Value** 1000
Examples TRAC:POIN 10
Query Syntax TRACe:POINts? [MINimum | MAXimum | DEFault]
Returned Parameters <NR1>
Related Commands TRAC:FEED

TRACe:FEED

This command is used to select the source of readings to be placed in the buffer. With VOLTage selected, voltage readings are placed in the buffer, TRAC:POIN maximum values is 2000. With CURRent selected, current readings are placed in the buffer, TRAC:POIN maximum values is 2000 . With TWO selected, voltage and current are placed in the buffer when storage is performed, TRAC:POIN maximum values is 1000.

Command Syntax TRACe:FEED <CRD>
Parameters VOLTage | CURRent | TWO
***RST Value** TWO
Examples TRAC:FEED VOLT
Query Syntax TRACe:FEED?

Returned Parameters <CRD>
Related Commands TRAC:POIN

TRACe:FEED:CONTrol

This command is used to select the buffer control. With NEVer selected, storage into the buffer is disabled. When NEXT is selected, the storage process starts, fills the buffer and then stops. The buffer size is specified by the :POINts command.

Command Syntax TRACe:FEED:CONTrol <CRD>
Parameters NEVer | NEXT
***RST Value** NEVer
Examples TRAC:FEED:CONT NEXT
Query Syntax TRACe:FEED:CONT?
Returned Parameters <CRD>
Related Commands TRAC:FEED

TRACe:DATA?

When this command is sent and the electronic load is addressed to talk, all the readings stored in the buffer are sent to the computer.

Query Syntax TRACe:DATA?
Returned Parameters {<NR3>}

TRACe:FILTer

This command is used to select whether the data in cache is the data filtered.

Command Syntax TRACe:FILTer[:STATe] <BOOL>
Parameters 0 | 1 | ON | OFF
***RST Value** OFF
Examples TRAC:FILT 1
Query Syntax TRACe:FILTer[:STATe]?
Returned Parameters <NR1>

TRACe:DELay

This command is used to select the delay time for the trigger in cache.

Command Syntax TRACe:DELay <NRf>
Parameters 0 to 3600s | MINimum | MAXimum | DEFault
UNIT S (second)
***RST Value** 0
Examples TRAC:DEL 1
Query Syntax TRACe:DELay? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>

TRACe:TIMer

This command is used to select the interval of the cache.

Command Syntax TRACe:TIMer <NRf>
Parameters 0.00002 to 3600s | MINimum | MAXimum | DEFault
UNIT S (second)
***RST Value** 1

Examples TRAC:TIM 0.1
Query Syntax TRACe:TIMer? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>

Chapter10 Source Subsystem

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

[SOURce:]INPut

These commands enable or disable the electronic load inputs. The state of a disabled input is a high impedance condition.

Command Syntax [SOURce:]INPut[:STATe] <bool>
Parameters 0 | 1 | OFF | ON
***RST Value** OFF
Examples INP 1
Query Syntax INPut[:STATe]?
Returned Parameters 0 | 1
Related Commands *RCL *SAV

[SOURce:]INPut:SHORt

This command programs the specified electronic load module to the maximum current that it can sink in the present operating range.

Command Syntax [SOURce:]INPut:SHORt[:STATe] <bool>
Parameters 0 | 1 | OFF | ON
***RST Value** OFF
Examples INP:SHOR 1
Query Syntax INPut:SHORt:STATe?
Returned Parameters 0 | 1
Related Commands INP

[SOURce:]INPut:TIMer

These commands enable or disable the load on timer.

Command Syntax [SOURce:]INPut:TIMer[:STATe] <bool>
Parameters 0 | 1 | OFF | ON
***RST Value** OFF
Examples INP:TIM 1
Query Syntax INPut:TIMer[:STATe]?
Returned Parameters 0 | 1
Related Commands INP:TIM:DEL

[SOURce:]INPut:TIMer:DELay

This command specifies the load on timer.

Command Syntax [SOURce:]INPut:TIMer <NRf+>
Parameters 1 to 60000s | MINimum | MAXimum | DEFault
Unit *seconds*
***RST Value** 10
Examples INP:TIM:DEL 5

Query Syntax [SOURce:]INPut:TIMer:DELay? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands INP:TIM

[SOURce:]REMote:SENSe

This command is used to select the remote measure state of the electronic load.

Command Syntax [SOURce:]REMote:SENSe[:STATe] <BOOL>
Parameters 0 | 1 | OFF | ON
***RST Value** 0
Examples REM:SENS 0
Query Syntax [SOURce:] REMote:SENSe [:STATe]?
Returned Parameters <CRD>

[SOURce:]FUNCtion

These equivalent commands select the input regulation mode of the electronic load.

CURRent constant current mode
RESistance constant resistance mode
VOLTage constant voltage mode
POWer constant power mode
Command Syntax [SOURce:]FUNCtion <function>
Parameters CURRent | RESistance | VOLTage | POWer |
***RST Value** CURRent
Examples FUNC RES
Query Syntax [SOURce:]FUNCtion?
Returned Parameters <CRD>

[SOURce:]FUNCtion:MODE

This command determines whether the input regulation mode is controlled by values in a list or by the command FUNCtion settings.

FIXed The regulation mode is determined by the FUNCtion or MODE command.
LIST The regulation mode is determined by the active list.
Command Syntax [SOURce:]FUNCtion:MODE <mode>
Parameters FIXed | LIST
***RST Value** FIXed
Examples FUNC:MODE FIX
Query Syntax [SOURce:]FUNCtion:MODE?
Returned Parameters <CRD>
Related Commands FUNC

[SOURce:]TRANsient

This command turns the transient generator on or off.

Command Syntax [SOURce:]TRANsient[:STATe] <bool>
Parameters 0 | 1 | OFF | ON
***RST Value** OFF

Examples TRAN 1
Query Syntax [SOURce:]TRANsient[:STATe]?
Returned Parameters 0 | 1
Related Commands CURR**:**TRAN:CURR:MODE
CURR:TRAN:ALEV

[SOURce:]PROTection:CLEar

This command clear the latch that disables the input when a protection condition such as overvoltage (OV) or overcurrent (OC) is detected. All conditions that generated the fault must be removed before the latch can be cleared. The input is then restored to the state it was in before the fault condition occurred.

Command Syntax [SOURce:]PROTection:CLEar
Parameters None
Examples INP:PROT:CLE

[SOURce:]CURRent

This command sets the current that the load will regulate when operating in constant current mode.

Command Syntax [SOURce:]CURRent[:LEVel][:IMMediate] <NRf+>
Parameters 0 through MAX | MINimum | MAXimum | DEFault
Unit *A* (amperes)
***RST Value** MINimum
Examples CURR 5 CURR:LEV 0.5
Query Syntax [SOURce:]CURRent[:LEVel][:IMMediate]? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands CURR:RANG

[SOURce:]CURRent:RANGe

This command sets the current range of the electronic load module. There are two current ranges.

High Range: model dependent, see Table 4-1
Low Range: model dependent, see Table 4-1

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.

NOTE: When this command is executed, the IMMediate, TRANsient, TRIGgered, and SLEW current settings are adjusted as follows:

If the existing settings are within the new range:

If the existing settings are outside the new range: The levels are set to the maximum value of the new range.

Command Syntax [SOURce:]CURRent:RANGe <NRf+>
Parameters 0 through MAX | MINimum | MAXimum | DEFault
Unit *A* (amperes)

***RST Value** MAXimum (high range)
Examples SOUR:CURR:RANGE MIN
Query Syntax [SOURce:]CURRent:RANGe? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands CURR CURR:SLEW

[SOURce:]CURRent:SLEW

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. MAXimum sets the slew to the fastest possible rate. MINimum sets the slew to the slowest rate.

Command Syntax [SOURce:]CURRent:SLEW[:BOTH] <NRf+>
Parameters MINimum to MAXimum | MAXimum | MINimum | DEFault
Unit *A* (amps per micro second)
***RST Value** MAXimum
Examples CURR:SLEW MAX
Related Commands CURR:SLEW:NEG CURR:SLEW:POS

[SOURce:]CURRent:SLEW:POSitive

This command sets the slew rate of the current for positive going transitions. MAXimum sets the slew to the fastest possible rate. MINimum sets the slew to the slowest rate.

Command Syntax [SOURce:]CURRent:SLEW:POSitive <NRf+>
Parameters MINimum to MAXimum | MAXimum | MINimum | DEFault
Unit *A* (amps per micro second)
***RST Value** MAXimum
Examples CURR:SLEW:POS MAX
Query Syntax [SOURce:]CURRent:SLEW:POSitive? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands CURR:SLEW

[SOURce:]CURRent:SLEW:NEGative

This command sets the slew rate of the current for negative going transitions. MAXimum sets the slew to the fastest possible rate. MINimum sets the slew to the slowest rate.

Command Syntax [SOURce:]CURRent:SLEW:NEGative <NRf+>
Parameters MINimum to MAXimum | MAXimum | MINimum | DEFault
Unit *A* (amps per micro second)
***RST Value** MAXimum
Examples CURR:SLEW:NEG MAX
Query Syntax [SOURce:]CURRent:SLEW:NEGative? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands CURR:SLEW

[SOURce:]CURRent:SLEWrate:STATe

This command enables or disables the over-current slow rise mode.

Command Syntax [SOURce:]CURRent:SLEWrate[:STATe] <Bool>
Parameters 0 | 1 | OFF | ON
***RST Value** OFF
Examples CURR:SLEW:STAT 1
Query Syntax [SOURce:]CURRent:SLEWrate:STATe?
Returned Parameters <NR3>
Related Commands CURR:SLEW

[SOURce:]CURRent:PROTection:STATe

This command enables or disables the over-current protection feature.

Command Syntax [SOURce:]CURRent:PROTection:STATe <Bool>
Parameters 0 | 1 | OFF | ON
***RST Value** OFF
Examples CURR:PROT:STAT 1
Query Syntax [SOURce:]CURRent:PROTection:STATe?
Returned Parameters <NR3>
Related Commands CURR:PROT

[SOURce:]CURRent:PROTection:LEVel

This command sets the soft current protection level. If the input current exceeds the soft 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 0 through MAX
Unit *A* (amperes)
***RST Value** MAXimum
Examples CURR:PROT:LEV 2
Query Syntax [SOURce:]CURRent:PROTection:LEVel?
Returned Parameters NR3
Related Commands CURR:PROT:DEL CURR:PROT:STAT

[SOURce:]CURRent:PROTection:DELay

This command specifies the time that the input current can exceed the protection level before the input is turned off.

Command Syntax [SOURce:]CURRent:PROTection:DELay <NRf+>
Parameters 0 to 60 seconds | MINimum | MAXimum | DEFault
Unit *seconds*
***RST Value** 3
Examples CURR:PROT:DEL 5
Query Syntax [SOURce:]CURRent:PROTection:DELay? [MINimum | MAXimum | DEFault]

Returned Parameters <NR1>
Related Commands CURR:PROT CURR:PROT:STAT

[SOURce:]CURRent:TRANsient:MODE

This command selects the operating mode of the transient generator as follows in constant current mode.

CONTinuous The transient generator puts out a continuous pulse stream after receipt a trigger signal.

PULSe The transient generator puts out a single pulse upon receipt of a trigger.

TOGGle The transient generator toggles between two levels upon receipt of a trigger.

Command Syntax [SOURce:]CURRent:TRANsient:MODE <mode>
Parameters CONTinuous | PULSe | TOGGle
***RST Value** CONTinuous
Examples CURR:TRAN:MODE TOGG
Query Syntax [SOURce:]CURRent:TRANsient:MODE?
Returned Parameters <CRD>
Related Commands CURR:TRAN:ALEV TRAN

[SOURce:]CURRent:TRANsient:ALEVel

[SOURce:]CURRent:TRANsient:BLEVel

This command specifies the transient level of the input current. The transient function switches between the level a and level b.

Command Syntax [SOURce:]CURRent:TRANsient:ALEVel <NRf+>
[SOURce:]CURRent:TRANsient:BLEVel <NRf+>
Parameters 0 through MAX | MINimum | MAXimum | DEFault
Unit *A* (amperes)
***RST Value** ALEVEL MAXimum , BLEVel MINnum
Examples CURR:TRAN:ALEV 5 CURR:TRAN:BLEV 0.5
Query Syntax [SOURce:]CURRent:TRANsient:ALEVel?
[MINimum | MAXimum | DEFault]
[SOURce:]CURRent:TRANsient:BLEVel?
[MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands CURR

[SOURce:]CURRent:TRANsient:AWIDth

[SOURce:]CURRent:TRANsient:BWIDth

This command specifies the transient pulse width of the input current.

Command Syntax [SOURce:]CURRent:TRANsient:AWIDth <NRf+>
[SOURce:]CURRent:TRANsient:BWIDth <NRf+>
Parameters 0.00002-3600
Unit S (second)
***RST Value** 500uS

Examples CURR:TRAN:AWID 0.001 CURR:TRAN:BLEV 0.02
Query Syntax [SOURce:]CURRent:TRANsient:AWIDth? [MINimum | MAXimum | DEFault]
[SOURce:]CURRent:TRANsient:BWIDth? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands CURR

[SOURce:]CURRent:HIGH

[SOURce:]CURRent:LOW

This command sets the high and low level of the voltage when in constant current mode.

Command Syntax [SOURce:]CURRent:HIGH <NRf+>
[SOURce:]CURRent:LOW <NRf+>
Parameters MINimum through MAX | MINimum | MAXimum | DEFault
Unit *V* (volts)
***RST Value** MAXimum
Examples CURR:HIGH 5
Query Syntax [SOURce:]CURRent:HIGH? [MINimum|MAXimum|DEFault]
[SOURce:]CURRent:LOW? [MINimum|MAXimum|DEFault]
Returned Parameters <NR3>

[SOURce:]VOLTage

This command sets the voltage that the load will regulate when operating in constant voltage mode.

Command Syntax [SOURce:]VOLTage[:LEVel][:IMMediate] <NRf+>
Parameters MINimum through MAX | MINimum | MAXimum | DEFault
Unit *V* (volts)
***RST Value** MAXimum
Examples VOLT 5
Query Syntax [SOURce:]VOLTage[:LEVel][:IMMediate]? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands VOLT:RANG

[SOURce:]VOLTage:RANGe

This command sets the voltage range of the electronic load module. There are only one voltage ranges.

Command Syntax [SOURce:]VOLTage:RANGe <NRf+>
Parameters MINimum through MAX | MINimum | MAXimum | DEFault
Unit *V* (volts)
***RST Value** MAXimum

Examples VOLT:RANG 15
Query Syntax [SOURce:]VOLTage:RANGe? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands VOLT

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

This command sets the voltage range of the electronic load module.

Command Syntax [SOURce:]VOLTage:RANGe:AUTO[:STATe] <bool>
Parameters 0 | 1 | ON | OFF
***RST Value** 1
Examples VOLT:RANG:AUTO 1
Query Syntax [SOURce:]VOLTage:RANGe:AUTO[:STATe]?
Returned Parameters <NR1>

[SOURce:]VOLTage:ON

This command sets the voltage of sink current on.

Command Syntax [SOURce:]VOLTage[:LEVel]:ON <NRf+>
Parameters 0 through MAX | MINimum | MAXimum | DEFault
Unit *V* (volts)
***RST Value** MINimum
Examples VOLT:ON 5
Query Syntax [SOURce:]VOLTage[:LEVel]:ON? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands VOLT:LATCh

[SOURce:]VOLTage:LATCh

This command sets the action type of Von.

Command Syntax [SOURce:]VOLTage:LATCh[:STATe] <b>
Parameters 0 | 1 | ON | OFF
***RST Value** ON
Examples VOLT:LATC 1
Query Syntax [SOURce:]VOLTage:LATCh[:STATe]? [MINimum | MAXimum | DEFault]
Returned Parameters 0 | 1
Related Commands VOLT:ON

[SOURce:]VOLTage:TRANsient:MODE

This command selects the operating mode of the transient generator as follows in constant voltage mode.

CONTinuous The transient generator puts out a continuous pulse stream after receipt of a trigger signal.

PULSe The transient generator puts out a single pulse upon receipt of a trigger.

TOGGle The transient generator toggles between two levels upon receipt of a trigger.

Command Syntax [SOURce:] VOLTage:TRANsient:MODE <mode>
Parameters CONTinuous | PULSe | TOGGle
***RST Value** CONTinuous
Examples VOLT:TRAN:MODE TOGG
Query Syntax [SOURce:] VOLTage:TRANsient:MODE?
Returned Parameters <CRD>
Related Commands VOLT:TRAN:ALEV TRAN

[SOURce:]VOLTage:TRANsient:ALEVel

[SOURce:]VOLTage:TRANsient:BLEVel

This command specifies the transient level of the input voltage. The transient function switches between the level a and level b.

Command Syntax [SOURce:] VOLTage:TRANsient:ALEVel <NRf+>
[SOURce:] VOLTage:TRANsient:BLEVel <NRf+>
Parameters MIN through MAX | MINimum | MAXimum | DEFault
Unit *V* (volts)
***RST Value** ALEVEL MAXimum , BLEVel MINnum
Examples VOLT:TRAN:ALEV 5 VOLT:TRAN:BLEV 0.5
Query Syntax [SOURce:] VOLTage:TRANsient:ALEVel? [MINimum | MAXimum | DEFault]
[SOURce:] VOLTage:TRANsient:BLEVel? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands VOLT

[SOURce:]VOLTage:TRANsient:AWIDth

[SOURce:]VOLTage:TRANsient:BWIDth

This command specifies the transient pulse width of the input voltage.

Command Syntax [SOURce:] VOLTage:TRANsient:AWIDth <NRf+>
[SOURce:] VOLTage:TRANsient:BWIDth <NRf+>
Parameters 0.00002-3600
Unit S (second)
***RST Value** 1000uS
Examples VOLT:TRAN:AWID 0.001 VOLT:TRAN:BLEV 0.02
Query Syntax [SOURce:] VOLTage:TRANsient:AWIDth? [MINimum | MAXimum | DEFault]
[SOURce:] VOLTage:TRANsient:BWIDth? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands VOLT

[SOURce:]VOLTage:HIGH

[SOURce:]VOLTage:LOW

This command sets the high and low level of the current when in constant voltage mode.

Command Syntax [SOURce:]VOLTage:HIGH <NRf+>
[SOURce:]VOLTage:LOW <NRf+>
Parameters MINimum through MAX | MINimum | MAXimum | DEFault
Unit *A* (amps)
***RST Value** MAXimum
Examples VOLT:HIGH 5
Query Syntax [SOURce:]VOLTage:HIGH? [MINimum|MAXimum|DEFault]
[SOURce:]VOLTage:LOW? [MINimum|MAXimum|DEFault]
Returned Parameters <NR3>

[SOURce:]RESistance

This command sets the resistance of the load when operating in constant resistance mode.

Command Syntax [SOURce:]RESistance[:LEVel][:IMMediate] <NRf+>
Parameters MINimum through MAX | MINimum | MAXimum | DEFault
Unit R□(ohms)
***RST Value** MAXimum
Examples RES 5 RES:LEV 3.5
Query Syntax [SOURce:]RESistance[:LEVel][:IMMediate]? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands RES:RANG

[SOURce:]RESistance:RANGe

This command sets the resistance range of the electronic load module.

Command Syntax [SOURce:]RESistance:RANGe <NRf+>
Parameters MINimum through MAX | MINimum | MAXimum | DEFault
Unit R□(ohms)
***RST Value** MAXimum (high range)
Examples RES:RANG 15 SOUR:RES:RANGE MIN
Query Syntax [SOURce:]RESistance:RANGe? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>

[SOURce:]RESistance:TRANsient:MODE

This command selects the operating mode of the transient generator as follows in constant resistance mode.

CONTinuous The transient generator puts out a continuous pulse stream after receipt of a trigger.

PULSe The transient generator puts out a single pulse upon receipt of a trigger.

TOGGle The transient generator toggles between two levels upon receipt of a trigger.

Command Syntax [SOURce:] RESistance:TRANsient:MODE <mode>
Parameters CONTinuous | PULSe | TOGGle
***RST Value** CONTinuous
Examples RES:TRAN:MODE TOGG
Query Syntax [SOURce:] RESistance:TRANsient:MODE?
Returned Parameters <CRD>
Related Commands RES**:**TRAN:ALEV TRAN

[SOURce:]RESistance:TRANsient:ALEVel

[SOURce:]RESistance:TRANsient:BLEVel

This command specifies the transient level of the resistance. The transient function switches between the level a and level b.

Command Syntax [SOURce:] RESistance:TRANsient:ALEVel <NRf+>
[SOURce:] RESistance:TRANsient:BLEVel <NRf+>
Parameters MIN through MAX | MINimum | MAXimum | DEFault
Unit R□(ohms)
***RST Value** ALEVEL MAXimum , BLEVel MINnum
Examples RES:TRAN:ALEV 5 POW:TRAN:BLEV 0.5
Query Syntax [SOURce:] RESistance:TRANsient:ALEVel?
[MINimum | MAXimum | DEFault]
[SOURce:] RESistance:TRANsient:BLEVel?
[MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands RES

[SOURce:]RESistance:TRANsient:AWIDth

[SOURce:]RESistance:TRANsient:BWIDth

This command specifies the transient pulse width of the input resistance.

Command Syntax [SOURce:] RESistance:TRANsient:AWIDth <NRf+>
[SOURce:] RESistance:TRANsient:BWIDth <NRf+>
Parameters 0.00002-3600
Unit S (second)
***RST Value** 1000uS
Examples RES:TRAN:AWID 0.001 RES:TRAN:BLEV 0.02
Query Syntax [SOURce:] RESistance:TRANsient:AWIDth?
[MINimum | MAXimum | DEFault]
[SOURce:] RESistance:TRANsient:BWIDth?
[MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands RES

[SOURce:]RESistance:HIGH

[SOURce:]RESistance:LOW

This command sets the high and low level of the voltage when in constant resistance mode.

Command Syntax [SOURce:]RESistance:HIGH <NRf+>
[SOURce:]RESistance:LOW <NRf+>
Parameters MINimum through MAX | MINimum | MAXimum | DEFault
Unit *V* (volts)
***RST Value** MAXimum
Examples RES:HIGH 5
Query Syntax [SOURce:]RESistance:HIGH? [MINimum|MAXimum|DEFault]
[SOURce:]RESistance:LOW? [MINimum|MAXimum|DEFault]
Returned Parameters <NR3>

[SOURce:]RESistance:VDRop

This command sets the constant resistance mode of the loads and the LED cut-off voltage value when LED test function is ON.

Command Syntax [SOURce:] RESistance:VDRop <NRf+>
Parameters 0 through MAX | MINimum | MAXimum | DEFault
Unit *V* (volts)
***RST Value** MINimum
Examples RES:VDR 5
Query Syntax [SOURce:] RESistance:VDRop? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands VOLT:VDR

[SOURce:]RESistance:LED[:STATe]

This command selects to enable or disable the LED test selection in constant resistance mode.

Command Syntax [SOURce:]RESistance:LED[:STATe] <b>
Parameters 0 | 1 | ON | OFF
***RST Value** ON
Examples RES:LED 1
Query Syntax [SOURce:]RESistance:LED[:STATe]?
Returned Parameters 0 | 1
Related Commands VOLT:ON

[SOURce:]POWer

This command sets the power of the load when operating in constant power mode.

Command Syntax [SOURce:]POWer[:LEVel][:IMMediate] <NRf+>

Parameters MINimum through MAX | MINimum | MAXimum | DEFault
Unit W (power)
***RST Value** MINinum
Examples POW 5 POW:LEV 3.5
Query Syntax [SOURce:]POWer[:LEVel][:IMMediate]? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands POW:RANG

[SOURce:]POWer:RANGe

This command sets the power range of the electronic load module.

Command Syntax [SOURce:] POWer:RANGe <NRf+>
Parameters MINimum through MAX | MINimum | MAXimum | DEFault
Unit W (power)
***RST Value** MAXimum (high range)
Examples POW:RANG 15 SOUR:POW:RANGE MIN
Query Syntax [SOURce:]POWer:RANGe? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>

[SOURce:]POWer:TRANsient:MODE

This command selects the operating mode of the transient generator as follows in constant current mode.

CONTinuous The transient generator puts out a continuous pulse stream after receipt of a trigger.

PULSe The transient generator puts out a single pulse upon receipt of a trigger.

TOGGle The transient generator toggles between two levels upon receipt of a trigger.

Command Syntax [SOURce:] POWer:TRANsient:MODE <mode>
Parameters CONTinuous | PULSe | TOGGle
***RST Value** CONTinuous
Examples POW:TRAN:MODE TOGG
Query Syntax [SOURce:] POWer:TRANsient:MODE?
Returned Parameters <CRD>
Related Commands POW:TRAN:ALEV TRAN

[SOURce:]POWer:TRANsient:ALEVel

[SOURce:]POWer:TRANsient:BLEVel

This command specifies the transient level of the input power. The transient function switches between the level a and level b.

Command Syntax [SOURce:] POWer:TRANsient:ALEVel <NRf+>
[SOURce:] POWer:TRANsient:BLEVel <NRf+>
Parameters 0 through MAX | MINimum | MAXimum | DEFault
Unit W (power)

***RST Value** ALEVEL MAXimum , BLEVel MINnum
Examples POW:TRAN:ALEV 5 POW:TRAN:BLEV 0.5
Query Syntax [SOURce:] POWer:TRANsient:ALEVel? [MINimum | MAXimum | DEFault]
[SOURce:] POWer:TRANsient:BLEVel? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands POW

[SOURce:]POWer:TRANsient:AWIDth

[SOURce:]POWer:TRANsient:BWIDth

This command specifies the transient pulse width of the input power.

Command Syntax [SOURce:] POWer:TRANsient:AWIDth <NRf+>
[SOURce:] POWer:TRANsient:BWIDth <NRf+>
Parameters 0.00002-3600
Unit S (second)
***RST Value** 1000uS
Examples POW:TRAN:AWID 0.001 POW:TRAN:BLEV 0.02
Query Syntax [SOURce:] POWer:TRANsient:AWIDth? [MINimum | MAXimum | DEFault]
[SOURce:] POWer:TRANsient:BWIDth? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands POW

[SOURce:]POWer:HIGH

[SOURce:]POWer:LOW

This command sets the voltage high and low level in constant power mode.

Command Syntax [SOURce:]POWer:HIGH <NRf+>
[SOURce:]POWer:LOW <NRf+>
Parameters MINimum through MAX | MINimum | MAXimum | DEFault
Unit *V* (volts)
***RST Value** MAXimum
Examples POW:HIGH 5
Query Syntax [SOURce:]POWer:HIGH? [MINimum|MAXimum|DEFault]
[SOURce:]POWer:LOW? [MINimum|MAXimum|DEFault]
Returned Parameters <NR3>

[SOURce:]POWer:PROTection

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

Note: Use POW:PROT:DEL command to stop the protection of instantaneous power, which is caused by stopping edit over power protection.

Command Syntax [SOURce:]POWer:PROTection[:LEVel] <NRf+>
Parameters 0 through MAX | MINimum | MAXimum | DEFault
Unit W (power)
***RST Value** MAXimum
Examples POW:PROT 100
Query Syntax [SOURce:]POWer:PROTection[:LEVel]? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands POW:PROT:DEL

[SOURce:]POWer:PROTection:DELay

This command specifies the time that the input power can exceed the protection level before the input is turned off.

Command Syntax [SOURce:]POWer:PROTection:DELay <NRf+>
Parameters 0 to 60 seconds | MINimum | MAXimum | DEFault
Unit *seconds*
***RST Value** 0
Examples POW:PROT:DEL 5
Query Syntax [SOURce:]POWer:PROTection:DELay? [MINimum | MAXimum | DEFault]
Returned Parameters <NR1>
Related Commands POW:PROT

[SOURce:]POWer:CONFig

This command sets the hard power protection level.

Command Syntax [SOURce:]POWer:CONFig[:LEVel] <NRf+>
Parameters 0 through MAX | MINimum | MAXimum | DEFault
Unit W (power)
***RST Value** MAXimum
Examples POW:CONFig 100
Query Syntax [SOURce:]POWer:CONFig[:LEVel]? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands POW:PROT

Chapter11 List Commands

List commands let you program complex sequences of input changes with rapid, precise timing, and Command allows you to edit fast, accurate timing and complex input changes list that can synchronized with trigger signal.

Each function for which lists can be generated has a list of values that specify the input at each list step.

[SOURce:]LIST:RANGe

This command sets the current range for list mode.

Command Syntax [SOURce:]LIST:RANGe <NRf>
Parameters MIN through MAX
Unit None
Examples LIST:RANGE 30
Query Syntax [SOURce:]LIST:RANGe?
Returned Parameters <NR3>
Related Commands LIST:LEV

[SOURce:]LIST:COUNt

This command sets the number of times that the list is executed before it is completed. The command accepts parameters in the range 1 through 65536, but 65536 is interpreted as infinity.

Command Syntax [SOURce:]LIST:COUNt <NRf+>
Parameters 1 to 65536 | MINimum | MAXimum
Examples LIST:COUN 3
Query Syntax [SOURce:]LIST:COUNt? [MINimum | MAXimum]
Returned Parameters <NR3>
Related Commands LIST:STEP

[SOURce:]LIST:STEP

This command sets the steps of the list.

Command Syntax [SOURce:]LIST:STEP <NRf+>
Parameters 2 to 84 | MINimum | MAXimum
Examples LIST:STEP 5
Query Syntax [SOURce:]LIST:STEP? [MINimum | MAXimum]
Returned Parameters <NR3>
Related Commands LIST:LEV

[SOURce:]LIST:LEVel?

This command specifies the setting for each list step.

Command Syntax [SOURce:]LIST:LEVel <NR1>, <NRf>
Parameters 1 to steps**,** MIN to MAX
Unit NONE, NONE
Examples LIST:LEV 1, 10 LIST:LEV 2, 15.2
Query Syntax [SOURce:]LIST:LEVel? <NR1>
Returned Parameters <NR3>

Related Commands LIST:RANG

[SOURce:]LIST:SLEW

This command sets the slew rate for each step. This command programs both positive and negative going slew rates. MAXimum sets the slew to its fastest possible rate. MINimum sets the slew to its slowest rate. LIST:SLEW? returns the number of points programmed.

Command Syntax [SOURce:]LIST:SLEW[:BOTH] <NR1> ,<NRf>
Parameters 1 to steps**,** MIN to MAX
Unit NONE, NONE
Examples LIST::SLEW1, 1.5 LIST:SLEW 2, MAX
Query Syntax [SOURce:]LIST:SLEW[:BOTH]? <NR1>
Returned Parameters <NR3>
Related Commands CURR:SLEW VOLT:SLEW RES:SLEW

[SOURce:]LIST:WIDth

This command sets the time width of each step of a LIST. Each value represents the time in seconds that the input will remain at the particular list step point before completing the step. If times exceed 16383S, the input remains at the present level until a trigger sequences the next point in the list. Else At the end of the dwell time, the input automatically changes to the next point in the list.

Command Syntax [SOURce:]LIST:WIDth <NR1>, <NRf>
Parameters 1 to steps**,** 20uS to max
Unit NONE**,** *s* (seconds)
Examples LIST:WID 1, 0.02 LIST:WID 2, 0.5
Query Syntax [SOURce:]LIST:WIDth? <NR1>
Returned Parameters <NR3>

[SOURce:]LIST:SAV

This command stores the present list file of the electronic load to a specified location in memory. Up to 7 files can be stored. file in saved in locations 1-7 are volatile, the data are nonvolatile, the data will be saved when power is removed.

Command Syntax [SOURce:]LIST**:**SAV <NR1>
Parameters 1 to 7
Example LIST:SAV 3
Related Commands [SOURce:]LIST:RCL

[SOURce:]LIST:RCL

This command restores a list file that was previously stored in memory with a LIST:SAV command to the specified location.

Command Syntax [SOURce:]LIST**:**RCL <NR1>
Parameters 1 to 7
Example LIST:RCL 3
Related Commands [SOURce:]LIST:SAV

Chapter12 Sense Subsystem

The Sense Subsystem is used to configure and control the measurement functions of the electronic load. A function does not have to be selected before you program its various configurations. A function can be selected any time after it has been programmed.

SENSe:AVERage:COUNt

The command is used to specify the filter count. In general, the filter count is the number of readings that are acquired and stored in the filter buffer for the averaging calculation. The larger the filter count, the more filtering that is performed.

Group	Sense
Syntax	SENSe:AVERage:COUNt <NR1> SENSe:AVERage:COUNt?
Arguments	2-16
Returns	<NR1>
Example	SENSe:AVERage:COUNt 3

SENSe:TIME:VOLTage1

SENSe:TIME:VOLTage2

The command sets the output voltage point of the load when measuring the rise and fall time of voltage output.

Command Syntax SENSe:TIME:VOLTage1 <NRf+>
SENSe:TIME:VOLTage2 <NRf+>
Parameters 0 through MAX | MINimum | MAXimum | DEFault
Unit *V* (volts)
***RST Value** MINimum / MAXimum
Examples VOLT:TIME:VOLT1 5
Query Syntax [SOURce:]SENSe:TIME:VOLTage? [MINimum | MAXimum | DEFault]
Returned Parameters <NR3>
Related Commands FETC:TIME?

SYSTem:SENSe[:STATe] <BOOL>

This command enables or disables remote sense measurement functions.

Group	Sense

Syntax	SYSTem:SENSe[:STATe] <BOOL> SYSTem:SENSe[:STATe]?
Arguments	0 | 1 | OFF | ON
Returns	0|1
Example	SYSTem:SENSe 1

Chapter13 Calibration Commands

CALibration Subsystem

Calibration commands' function:

- Enable or disable calibration mode
- Calibrate input function, current offset or gain, and save the newest calibration constant in nonvolatile memory.

CALibrate:SECure[:STATe]

This command can enable or disable the calibration mode. You must enable calibration mode so that you can receive other calibration commands. The first parameter defines enable or disable status. The second parameter represents the code. If the calibration mode is enabled while the current code is not 0,in this circumstance, you need to configure the second parameter. If the code is not set or incorrect, a error will be generated and calibration mode remains disabled. Query command can only return the status of calibration mode but code. Everytime, when calibration status changed from enabled status to disabled status, any new calibration constants will be lost after power-down unless you have utilized the save command CALibrate:SAVE.

Command Syntax CALibrate:SECure[:STATe] <bool> [,<SRD>]
Parameters 0 | 1 | OFF | ON [,<password>]
***RST Value** ON
Examples CAL:SEC 0, N3301A CAL:SEC ON
Query Syntax CALibrate:SECure[:STATe]?
Returned Parameters <NR1>
Related Commands CAL:SAVE CAL:INIT

CALibrate:INITial

This command only be used in calibration mode. It re-saves the factory calibration constants in nonvolatile memory.

Command Syntax CALibrate:INITial
Parameters None
Examples CAL:INIT
Related Commands CAL:STAT CAL:INIT

CALibrate:SAVe

This command only be used in calibration mode. Save the new calibration constants in nonvolatile memory(after finishing the voltage or current calibration process).

Command Syntax CALibrate:SAVE
Parameters None
Examples CAL:SAVE
Related Commands CAL:STAT CAL:INIT

CALibrate:CURRent:POINt

This command is used to set the calibration point in CC mode, only effective in

calibration mode.P1,P2 are used in low current range. P3, P4 are used in high current range.

Command Syntax CALibrate:CURRent:POINt <point>
Parameters P1 | P2 | P3 | P4
Examples CAL:CURR:POIN P2
Related Commands CAL:STAT CAL:SAV

CALibrate:CURRent[:LEVel]

This command is only used in calibration mode. Input the calibration current value according to the external measuring meter. But you must select a calibration grade as a reference to the value you just input. These constants do not exist in the nonvolatile memory before you send the save command CALibrate:SAVE.

Command Syntax CALibrate:CURRent[:LEVel] <NRf>
Parameters <external reading>
Unit A (amps)
Examples CAL:CURR 3.2223
Related Commands CAL:STAT CAL:SAV

CALibrate:CURRent:METEr:POINt

This command is used to set the calibration point in CC mode, only effective in calibration mode.P1,P2 are used in low current range. P3, P4 are used in high current range. This can only calibrate the ammeter that selected through command CONF:CURR.

Command Syntax CALibrate:CURRent:METEr:POINt <point>
Parameters P1 | P2 | P3 | P4
Examples CAL:CURR:METEr:POIN P2
Related Commands CAL:STAT CAL:SAV

CALibrate:CURRent:METEr[:LEVel]

This command is only effective in calibration mode. Input the calibration current value according to reading of external measuring meter. Before you input this value, please select a calibration grade firstly (using the command CAL:CURR:METE:POIN).These constant do not exist in the nonvolatile memory unless you send the command CALibrate:SAVE.

Command syntax CALibrate:CURRent:METEr[:LEVel] <NRf>
Parameters <external reading>
Unit A (amps)
Examples CAL:CURR 3.2223
Related Commands CAL:STAT CAL:SAV

CALibrate:VOLTage:POINt

This command is only effective in calibration mode, is used to set the calibration point in CV mode.P1, P2 are used in low voltage range,P3,P4 are used in high voltage range. Only be used to calibrate voltage source and voltmeter.

Command Syntax CALibrate:VOLTage:POINt <point>
Parameters P1 | P2 | P3 | P4
Examples CAL:VOLT:POIN P2
Related Commands CAL:STAT CAL:SAV

CALibrate:VOLTage[:LEVel]

This command is only effective in calibration mode. According to the reading of external measuring meter to input the calibration voltage. Please select a calibration grade before you input a calibration value(using the command CALibrate:VOLTage:POIN).These constant do not exist in the nonvolatile memory unless you have sent the command CALibrate:SAVE.

Command Syntax CALibrate:VOLTage[:LEVel] <NRf>
Parameters <external reading>
Unit V (volts)
Examples CAL:VOLT 3.2223
Related Commands CAL:STAT CAL:SAV

CALibrate:VOLTage:METEr:POINt

This command is only effective in calibration mode, is used to set the calibration point in CV mode.P1, P2 are used in low voltage range, P3, P4 are used in high voltage range. Only be used to calibrate voltage source and voltmeter.

Command Syntax CALibrate:VOLTage:METEr:POINt <point>
Parameters P1 | P2 | P3 | P4
Examples CAL:VOLT:METEr:POIN P2
Related Commands CAL:STAT CAL:SAV

CALibrate:VOLTage:METEr[:LEVel]

This command only be effective in calibration mode. According to the reading of external measuring meter to input the calibration voltage. Before you input the value, you should firstly select a calibration grade(with command CALibrate:VOLTage:POINt).All these constant do not exist in the nonvolatile memory unless you have send the command CALibrate:SAVE to save them.

Command Syntax CALibrate:VOLTage:METEr[:LEVel] <NRf>
Parameters <external reading>
Unit V (volts)
Examples CAL:CURR 3.2223
Related Commands CAL:STAT CAL:SAV

Chapter14 Error Information

Error Number List

This appendix describes the error numbers and descriptions that are returned by the electronic load. Error numbers are returned on the front panel in two ways:

- Error number with messages after the SYSTem:ERRor? query
- Error number with an NR1 and a string after the SYSTem:ERRor? query

Errors from 100 to 199 (Set bit #5 of standard event status register) are explained as follows.

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

(-150) String data error
(-151) Invalid string data [e.g., END received before close quote]
(-158) String data not allowed
(-160) Block data error
(-161) Invalid block data [e.g., END received before length satisfied]
(–168) Block data not allowed
(-170) Expression error
(-171) Invalid expression
(-178) Expression data not allowed

Execute errors from -200 to -299 (Set bit #4 of standard event register) are explained as follows.

(-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
(-230) Data Corrupt or Stale
(-270) Macro error
(–272) Macro execution error
(–273) Illegal macro label
(–276) Macro recursion error
(–277) Macro redefinition not allow

System errors from -300 to -399 (Set bit 3 of standard event register) are explained as follows.

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

Query errors from -400 to -499 (Set Bit2 of standard event register) are explained as follows.

(–499) (sets Standard Event Status Register bit #2)
(-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]

Checking errors from 0 to 99 (Set bit 3 of standard event register) are explained as follows.

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

Equipment related errors from 100 to 32767 (Set bit 3 of standard event register) are explained as follows.

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
408	Output mode switch must be in NORMAL position
600	Lists inconsistent [lists have different list lengths]
601	Too many sweep points
602	Command only applies to RS-232 interface
603	FETCH of data that was not acquired
604	Measurement overrange
605	Command not allowed while list initiated
610	Corrupt update data
611	Not Updating

Contact Us

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

1. Visit ITECH website: www.itechate.com.
2. Select the most convenient contact method for further information.