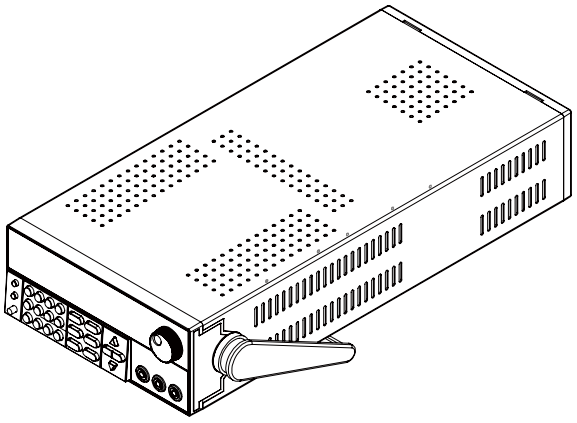
DC Programmable Power Supply

Series IT6800A/B Programming Guide



Model: IT6831A/IT6832A/IT6833A/IT6835A/IT6861A
/IT6862A/IT6863A/IT6872A/IT6873A/IT6874A
/IT6832B/IT6833B/IT6835B/IT6861B/IT6862B/
IT6863B/IT6872B/IT6873B/IT6874B

Version: 2.2

Notices

Manual Part Number

IT6800A/B-402200

Revision

2nd Edition: July 11, 2018

Itech Electronic, Co., Ltd.

Trademarks

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

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

Warranty

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

Technology Licenses

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

Restricted Rights Legend

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

Safety Notices

CAUTION

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

WARNING

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

NOTE

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

Quality Certification and Assurance

We certify that series IT6800A/B power supply meets all the published specifications at time of shipment from the factory.

Warranty

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

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

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

Limitation of Warranty

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

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

Safety Symbols

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

	Frame or chassis terminal	-	-

Safety Precautions

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

WARNING

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

CAUTION

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

Environmental Conditions

The instrument is designed for indoor use and an area with low condensation. The table below shows the general environmental requirements for the instrument.

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

Note

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

Regulatory Markings

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

Waste Electrical and Electronic Equipment (WEEE) Directive

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

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

Product Category

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

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

Compliance Information

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

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

Conforms with the following product standards:

EMC Standard

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

Reference Standards

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

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

Safety Standard

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

Content

Chapter1 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}

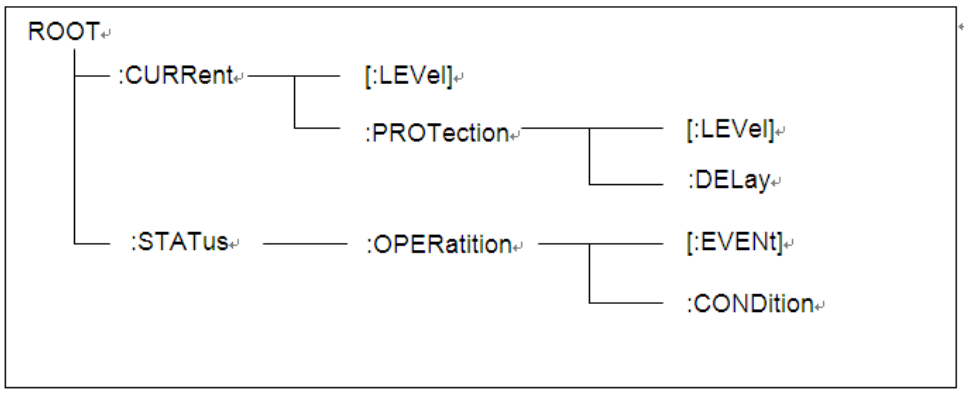
POLarity {NORMal|INVerted}

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 overall electronic load functions, such as reset, status, and synchronization. All common commands consist of a three-letter mnemonic preceded by an asterisk: *RST *IDN? *SRE 8.
- Subsystem commands perform specific electronic load functions. They are organized into an inverted tree structure with the "root" at the top. The following figure shows a portion of a 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. However, the short-form version is indicated by upper case characters. Examples:

:SYSTem:PRESet long-form

:SYST:PRES short form

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

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

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

Query

Observe the following precautions with queries:

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

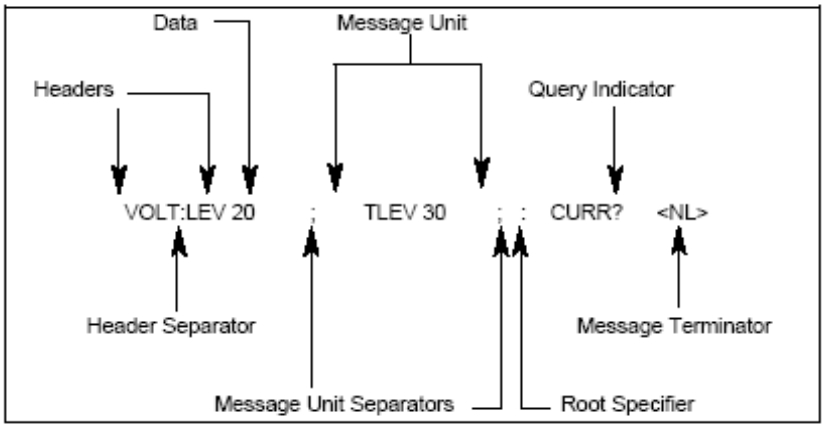
1.4 Message Type of SCPI

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

- program message: A program message consists of one or more properly formatted SCPI commands sent from the controller to the electronic load. The message, which may be sent at any time, requests the electronic load to perform some action.
- response message: A response message consists of data in a specific

SCPI format sent from the electronic load to the controller. The electronic load sends the message only when commanded by a program message called a "query."

The next figure illustrates SCPI message structure:



The Message Unit

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

VOLTage 20<NL>

Headers

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

Query Indicator

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

Message Unit Separator

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

Root Specifier

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

Message Terminator

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

- newline (<NL>),decimal 10 or hexadecimal 0X0A in ASCII.

- end or identify (<END>)
- both of the above (<NL><END>).

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

Command execution rules

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

1.5 Response Data Type

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

<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 a query command.

Sending a response message

After sending a query command, the response message is placed in the Output Queue. When the IT6800A/B 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 IT6800A/B 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 IT6800A/B 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 IT6800A/B Series to talk.

- Rule 2.The complete response message must be received by the computer before another program message can be sent to the IT6800A/B Series.

1.6 Command Format

Formats for command display are as follows:

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

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

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

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

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

Colon (:)

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

APPL:SIN 455E3,1.15,0.0

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

Semicolon (;)

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

TRIG:SOUR EXT; COUNT 10

has the same effect as sending the following two commands:

TRIG:SOUR EXT
TRIG:COUNT 10

Question mark (?)

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

TRIG:COUN 10

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

TRIG:COUN?

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

TRIG:COUN?MIN
TRIG:COUN?MAX

Comma (,)

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

Space

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

Generic commands (*)

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

1.7 Data Type

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

- Numerical parameter

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

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

- <NR1>: There is an implicit decimal point in the last bit,such as 273
- <NR2>: There is an explicit decimal point,such as .273
- <NR3>: There are an explicit decimal point and exponential,such as 2.73E+22.73E+2
- <Nrf>:The extensible form includes <NR1>, <NR2> and <NR3>,such as 273 273. 2.73E2273 273. 2.73E2
- <Nrf+>: The extensible decimal form includes <NRf> and MIN MAX DEF, such as 273 273. 2.73E2 MAX. MIN and MAX are the minimum and maximum finite number. Within the range of the parameter definition, DEF is the default of the parameter.

- Discrete parameter

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

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

- Boolean parameter

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

DISPlay {OFF|0|ON|1}

- ASCII string Parameter

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

DISPlay:TEXT <quoted string>

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

DISP:TEXT "WAITING..."

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

DISP:TEXT 'WAITING...'

1.6 Remote Operation

IT6800A series power supply provides two standard communication interfaces: RS232, USB. IT6800B series power supply provides three standard communication interfaces: RS232, USB, GPIB. User can select anyone to communicate with the computer.

1.6.1 RS232 interface

There is a COM port (DB9)connector at the rear of the power supply, when connect to computer, you need to connect a cable with COM port on both side;

To active connection, you need to press the front panel composite key (Shift)+ I-set to configurate settings the same as computer configuration settings. RS-232 interface can be used to program all of the SCPI commands.

NOTE

The RS232 settings on the computer side must match the settings in the system menu of the instrument. If any change, please press (Shift)+ I-set key to enter the system menu and then implement the changes.

RS-232 data format

RS-232 data is 10-bit words contain a start bit and a stop bit. The start bit and stop bit can't be edited. However, you can select the parity items by pressing (Shift) + I-set key on the front panel and enter the system menu(SYSTEM SET).

Parity options are stored in nonvolatile memory

Baud Rate

The front panel (Shift)+ I-set button allows the user to select a baud rate which is stored in the non-volatile memory: 4800/9600/19200 /38400/57600/115200

RS-232 connection

Use a RS232 cable with DB-9 interface, RS-232 serial port can connect

with the controller (eg PC). Do not use blank Modem cable. Table 2-2 shows the plug pins.

If your computer is using a RS-232 interface with DB-25 connector, you need an adapter cable with a DB-25 connector at one end and the other side is a DB-9(not blank modem cable)

Pin introduction of RS-232 connector

Pin number	Description
1	No connection
2	TXD, transfer date
3	RXD, receive data
4	No connection
5	GND, ground
6	No connection
7	CTS, clear transfer
8	RTS, ready to transfer
9	No connection

RS-232 Troubleshooting:

If there is RS-232 connection problem, check the following:

1. Computer and power supply must configure the same baud rate, parity, data bits and flow control options. Note that the power configuration as a start bit and a stop bit (these values are fixed).
2. As described before in RS-232 connector, you must use the correct interface cable or adapter. Note that even if the cable has the right plug, the internal wiring may be wrong.
3. Interface cable must be connected to the correct serial port on the computer (COM1, COM2, etc.).

Communication Settings

Before communication, you should first make the following parameters on the power supply and PC matches.

Baud Rate: 9600 (4800,9600,19200,38400,57600,115200). You can enter the system menu from the front panel, and then set the baud rate.

- Data bits: 8
- Stop Bits: 1
- calibration (none, even, odd)
- EVEN 8 data bits, have even parity
- ODD 8 data bits have odd parity
- NONE 8 data bits, no parity
- MODE: （SIG,MUX）
- SIG a single instrument is connected to the communication
- MUX multiple sets of instrument communication with PC at the same time, in this situation, instrument address must be set to distinguish among all the units(address 0 ～30, default set is 0）

Local Address: (0 ~ 30, the factory default setting is 0)

Parity=None	Start Bit	8 Data Bits	Stop Bit

1.6.2 USB interface

Use a Cable with two USB port to connect the power and the computer. All power functions can be programmed via USB.

The USB488 interface functions of the power supply described as below:

- interface is 488.2 USB488 interface.
- Interface Receive REN_CONTROL, GO_TO_LOCAL, and LOCAL_LOCKOUT request.
- Interface receive MsgID = TRIGGER USBTMC order information, and will pass TRIGGER order to the functional layer.

Power USB488 device functions described as follows:

- device can read all of the mandatory SCPI orders.
- device is SR1 enabled.
- device is RL1 enabled.
- device is DT1 enabled.

To communicate via USB interface, you should first select the communication interface as USB in the system menu, the detailed operation is:

1. Press (Shift)+ I-set to enter the menu.
2. Press ▽ to select SYST SET, and then press Enter, “P-MEM KEEP” is displayed, then press ▽ to select COMM RS232, press Enter.
3. Press ▽ to select USB, and Enter to confirm.
4. Press Esc to escape the menu.

NOTE

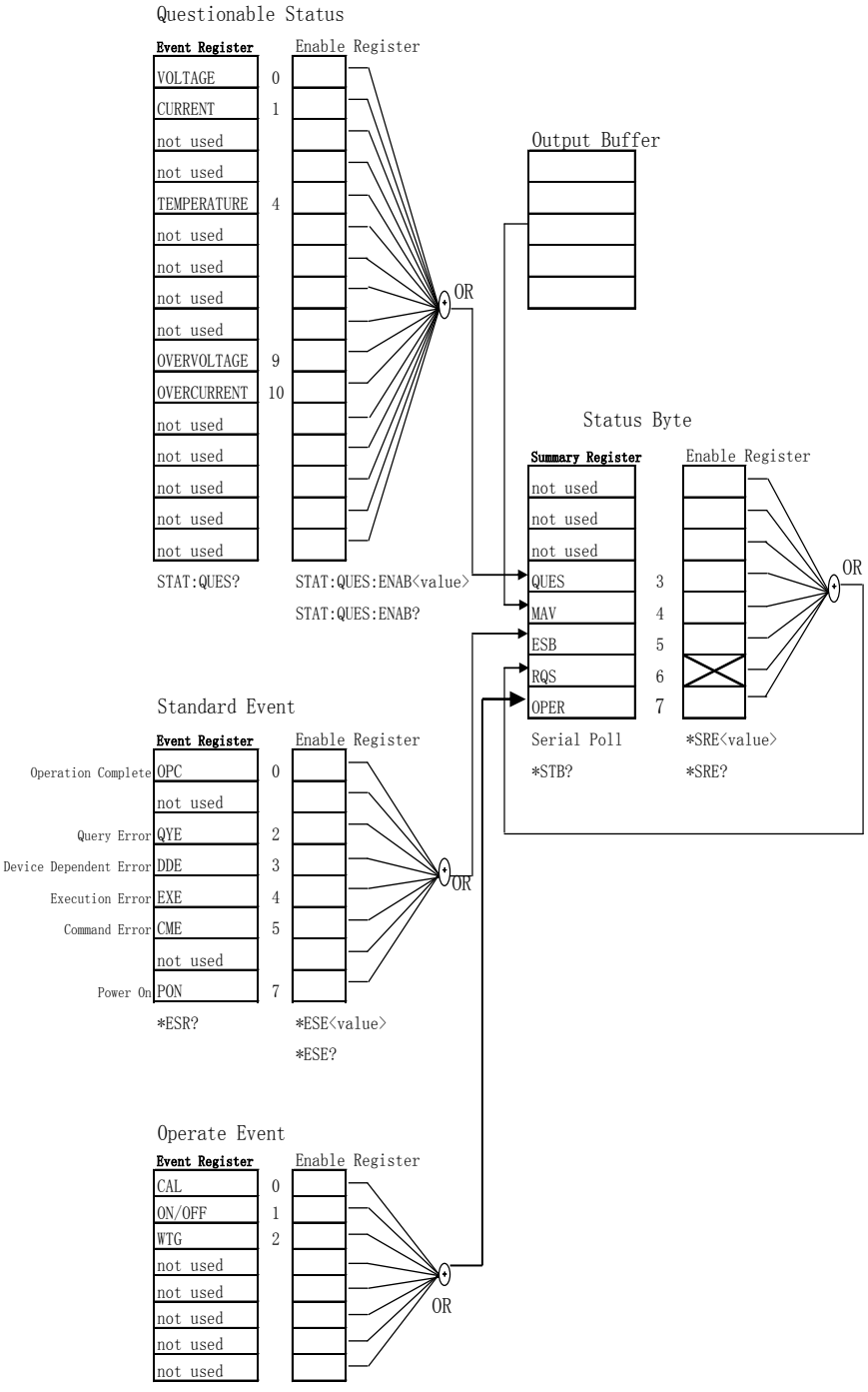
when communicate via USB interface, only SCPI commands are supported, so you should make sure the LANG(SCPI/FRAME) item in the system menu is selected as SCPI, if not, please change to “SCPI”. IT6800B series only support SCIP.

1.6.3 GPIB interface (Only for IT6800B series)

First, Connect the GPIB interface on the power supply and the GPIB card on computer via IEEE488 bus, must be full access and tighten the screws. Then set the address, the address range of the power : 0 to 30, can set by the function key on the front panel, press the (Shift)+ I-set key to enter the system menu function, find the GPIB address setting by ▼ button, type the address, Enter key to confirm. GPIB address is stored in nonvolatile memory line.

Chapter2 SCPI Status Register

The IT6800A/B series power supply records the different status of the instrument through the three status register group, the three status register group are: standard event register, query status register and status bit enable register. Status bit enable register records the information of the other status register



Event register is read only register, used to store the implementation status of the power, the data used in the event register latches in the form, once the data is stored, subsequent data will be completely ignored. Even

can't changed by re-setting order (* RST) or equipment restart, but if query the data in event register or send clear order *CLS(clear status), the event register will be automatically cleared. The main content which the standard event register records is: whether power output is turned on, order syntax errors, order execution errors, self-test or calibration errors, query errors and so on.

Bit		Decimal Value	Definition
0	OPC	1	Operation Complete. All commands prior to and including an *OPC command have been executed.
1	Not Used	0	Always set to 0.
2	QYE	4	Query Error. The power supply tried to read the output buffer but it was empty. Or new command line was received before a previous query had been read. Or both the input and output buffers are full.
3	DDE	8	Device Error. A self-test or calibration error occurred (see error numbers 601 through 750 in chapter 5).
4	EXE	16	Execution Error. An execution error occurred (see error numbers -211 through -224 in chapter 5).
5	CME	32	Command Error. A command syntax error occurred (see error numbers -101 through -178 in chapter 5).
6	Not Used	0	Always set to 0.
7	PON	128	Power On. Power has been turned off and on since the last time the event register was read or cleared

Query status register provide some information of the power, such as over voltage, over temperature, over current. You can also monitor the change of the constant current and voltage status via the register, for example, data bit 0 is the constant current mode of the power supply, data bit 1 is constant voltage mode of the power supply and so on.

Bit		Decimal Value	Definition
0	Voltage	1	The power supply is/was in the constant current mode.
1	Current	2	The power supply is/was in the constant voltage mode.
2-3	Not used	0	Always set to 0.
4	Over temperature	16	The fan has a fault condition.
5-8	Not used	0	Always set to 0.
9	Over voltage	512	The overvoltage protection circuit has tripped.
10	Over Current	1024	The over current protection circuit has tripped.
11-15	Not used	0	Always set to 0.

Status byte register records the information of other registers. The query data is temporally stored in the output buffer of the power supply, and feedback to

customer through BIT4 bit. The data bits in Status byte group will not be latched, when the information in event register is changed, the correspo0nding bit in status byte register will subsequently be changed.

Bit		Decimal Value	Definition
0-2	Not used	0	Always set to 0.
3	QUES	8	One or more bits are set in the questionable status register (bits must be enabled in the enable register).
4	MAV	16	Data is available in the power supply output buffer.
5	ESB	32	One or more bits are set in the standard event register (bits must be enabled in the enable register).
6	RQS	64	The power supply is requesting service (serial poll).
7	Not used	0	Always set to 0.

Chapter3 SCPI Essential Order

SYSTem:ERRor?

This order is used to query the error information of the power supply. When the error LED lights on the front panel, it tells us that there is one or more hardware or order syntax error happened. There are at most 30 groups of error message.

Send the order once will read one error message from the error queue.

- Error messages follow the FIFO (first-in-first-out) principle. The first error will be returned first. When you read all the error tips information in the error queue. The ERROR led will turn off. When error occur the buzzer of the power supply will beep once.
- If more than 30 errors occurred, the last one stored in the queue will be replaced by “-350”, Meaning "too many mistakes". If the error messages in the error message queue are not read out, the error following will not be saved to the queue. If there is no error message, when read error information, it will return “+0”, meaning “no error”.
- If turn off the power supply or send “CLS (clear status) order, the error messages in the error queue will be cleared. *RST order will not clear the error messages in the error queue.

Order syntax

SYSTem:ERRor?

Parameter

None

Return parameter

〈NR1〉,〈SRD〉

SYSTem:VERSion?

This order can query the version of the SCPI order. Returned value is a string like “YYYY.V”, in which the YYYY is the year of this version, V is the version.

Order syntax

SYSTem:VERSion?

Parameter

None

Return parameter

<NRf>

SYSTem:REMote

This order is used to set the power supply to remote control mode through the RS232 interface. Except for the Local key on the front panel, other keys are

locked and can't be used. Send or receive orders without first sending the order to configure the remote control may cause unpredictable results.

Order syntax

SYSTem:REMote

Parameter

None

SYSTem:LOCal

This order is used to set the power supply to panel control mode through the RS232 interface. After execute this order, all the buttons on the front panel can be used.

Order syntax

SYSTem:LOCal

Parameter

None

SYSTem:RWLock

This order is used to set the power supply to remote control mode through the RS232 interface, and LOCAL is not available. Executing this order will set the power supply to remote control mode, the same result as order SYST:REM function, the only difference is this order will lock all the buttons on the front panel including LOCAL button (except Shift and Meter key).

Order syntax

SYSTem:RWLock

Parameter

None

SYSTem:BEEPer

This order is used to test the beeper, after execute this order, the power supply will issue a beep sound.

Order syntax

SYSTem:BEEPer

Parameter

None

Return parameter

None

Chapter4 Trigger Commands

TRIGger

This order is used to generate a trigger signal when trigger source is in BUS mode. The function of this order is similar to *TRG.

Order syntax

TRIGger[:IMMediate]

Parameters

none

Related orders

*TRG

TRIG:SOUR

TRIGger:SOURce

This order is used to choose the source of trigger signal. Power supply can receive the signal from front panel i.e by pressing Trigger button, or receive a trigger signal by BUS. When you execute order *RST, the trigger mode should be set in MANUAL.

Order syntax

TRIGger:SOURce <mode>

Parameters

BUS|MANUAL

Query syntax

TRIGger:SOURce?

*RST value

MANUAL

Chapter5 Output Commands

OUTPut

This order is used to open or close the output of power supply. When output is off, the voltage and current of power supply is 0V/1mA.

Order syntax

OUTPut[:STATe] <bool>

Parameters

0|1|OFF|ON

Query syntax

OUTPut?

*RST value

0

OUTPut:TIMer

This order is used to set the state of output timer.

Order syntax

OUTPut:TIMer[:STATe] <bool>

Parameters

0|1|OFF|ON

Query syntax

OUTPut:TIMer?

OUTPut:TIMer:DATA

This order is used to set output timer's time.

Order syntax

OUTPut:TIMer:DATA <NRf>

Parameters

0.1-99999.9

Unit

S

Query syntax

OUTPut:TIMer:DATA?

Parameters

<NRf>

Chapter6 Current Control Commands

CURRent{<*current value*>|MIN|MAX|UP|DOWN|DEF}

This order is used to set the output current. No matter what the current range you select currently, the output current is subject to the newest setting value. Besides, MIN and MAX can be used as parameters. MIN represents 0A, MAX represents the max value of the current range. You may send CURR? MIN or CURR? MAX orders to quire the minimum and maximum current within the range. DEF represents default value and it also can be used to set value.

We can also send CURR UP and CURR DOWN orders to increase or decrease current, the stepping value should be set with order “CURR:STEP”.

Oder syntax

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

Parameters

MIN to MAX

Unit

A

*RST value

MAX

Query syntax

CURRent?[MINimum|MAXimum]

Return parameters

<NR2>

Example

```
CURR:STEP 0.01      //set the stepping value to be 0.01A
CURR UP             //enable the output current increase value one time
CURR:STEP 0.02      //set the stepping value to be 0.02A
CURR DOWN           //enable the output current decrease value one time
```

CURRent:STEP

This order is used to service for CURR UP and CURR DOWN two orders. It can set the stepping value of current. Besides, CURR:STEP? DEF order can inquire the stepping resolution of current of this unit. For example, the stepping setting value is 0.01, it represents 10mA.

Order syntax

[SOURce:]CURRent[:LEVel][:IMMediate]:STEP[:INCRement] <NRf>

Parameters

MIN to MAX

*RST value

the stepping resolution of current of this unit.

Query order

CURRent:STEP?

Return parameters

<NR2>

Chapter7 Voltage Control Commands

VOLTage {<*voltage value*>|MIN|MAX|UP|DOWN|DEF}

This order is used to set the output voltage. No matter what the voltage range you select currently, the output voltage is subject to the newest setting value. Besides, MIN and MAX can be used as parameters.MIN represents 0V,MAX represents the max value of the voltage range. You may send VOLT? MIN or VOLT? MAX orders to quire the minimum and maximum voltage within the range. DEF represents default value, it also can be used to set value.

We can also send VOLT UP and VOLT DOWN orders to increase or decrease voltage, the stepping value should be set with order " VOLTage:STEP".

Order syntax

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

Parameters

MIN to MAX

Unit

V

*RST value

MIN

Query syntax

VOLTage? [MINimum|MAXimum]

Return parameters

<NR2>

Example

```
VOLT:STEP 0.01        //set the stepping value to be 0.01V
VOLT UP               //enable the output voltage increase one time
VOLT:STEP 0.02        //set the stepping value to be0.02V
VOLT DOWN             //enable the output value decrease one time
```

VOLTage:STEP {<*value*>|DEFault}

This order is used to service for VOLT UP and VOLT DOWN two orders. It can set the stepping value of voltage. Besides, VOLT:STEP? DEF order can inquire the stepping resolution of voltage of this unit. For example, the stepping setting value is 0.01, it represents 10mV.

Order syntax

[SOURce:]VOLTage[:LEVel][:IMMediate]:STEP[:INCRement] <NRf>

Parameters

MIN to MAX

*RST value

the stepping resolution of voltage of this unit.

Query order

VOLT:STEP?

Return parameters

<NR2>

VOLT:PROTection{<*vaule*>|MINimum|MAXimum}

This order is used to set the upper limit of over voltage protection point. If the output voltage value is higher than the upper limit of OVP, then we can assume that internal parts has been shorted. Query state register ”OV” bit has been set. When power supply is protected, then we can through order VOLT:PROT:CLE to clear the OVP state.

Order syntax

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

Parameters

MIN to MAX

Unit

V

Query syntax

VOLT:PROT? { MINimum|MAXimum}

Return parameters

<NR2>

VOLT:PROTection:[STATe] {0|1|OFF|ON}

This order is used to open or close OVP function i.e to set the state of OVP.

Order syntax

[SOURce:]VOLTage:PROTection:STATe

Parameters

0|1|OFF|ON

Query syntax

VOLTage:PROTection:[STATe]?

Return parameters

0|1

VOLT:PROTection:TRIPed?

This order is used to inquire the execute state of OVP. If the return value is “1”, this represents that the OVP circuit has been triggered and the OVP state does not be cleared. If the return value is “0”, then it represents the OVP circuit does not be triggered.

Order syntax

[SOURce:]VOLTage:PROTection:TRIPed?

Return parameters

0|1

VOLT:PROTection:CLEar

This order is used to clear the state of OVP. After you executing this order, the output value will be restored to the previous output and the upper limit of OVP will be kept. Of course, before you sending this order, you should enable the output voltage lower than the upper limit of OVP. In addition, please note that making sure to remove the external power supply before you sending this order.

Order syntax

[SOURce:]VOLTage:PROTection:CLEar

VOLT:LIMIT <*volts*>

This order is used to set the upper limit of the output voltage.

Order syntax

[SOURce:]VOLTage:LIMIT[:LEVel] <NRf>

Parameters

MIN to MAX

Unit

V

Query syntax

VOLTage:LIMIT?

Return parameters

<NR2>

[SOURce:]VOLTage:RANGe[high/low]

IT6860A/IT6860B/IT6870A/IT6870B series are dual-range power supply. This order is used to set the voltage range of the instrument.

Order syntax

[SOURce:]VOLTage:RANGe[high/low]

Parameters

high/low

Chapter8 Compound Control Command

APPLy {<*voltage value*>|DEF|MIN|MAX} [{<*current value*>|DEF|MIN|MAX}]

This order has combined two kinds of orders: VOLTage and CURRent. When sending this order to unit, power supply will output voltage and current according to the current setting of this order. Of course the precondition is the setting value is within the set range. If not, then a execution error will occur. You can also use DEF、MIN or MAX as the setting parameters.DEF represents factory setting. MIN will enable the voltage and current to be 0. MAX will enable the output to be the highest value in the range.

Order syntax

[SOURce:]APPLy <NRf>

Parameters

MIN to MAX

Unit

V, A

Query syntax

APPLy?

Return parameters

<NR2>

Chapter9 Measurement Commands

MEASure:CURRent?

This order is used to measure and return output current value.

Order syntax

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

Return parameters

<NR2>

FETCh:CURRent?

This order is used to read the latest current to be processed from sampling buffer. When you send this order, then our unit will communicate with PC, and sending the current data to PC. This order will not affect our unit's setting or trigger the measurement operation. It only needs the nearest reading it can get. The returned readings will keep the old before it get a new data.

Order syntax

FETCh:CURRent[:DC]?

Return parameters

<NR2>

MEASure[:VOLTage]?

This order can inquire the current output voltage value.

Order syntax

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

Return parameters

<NR2>

FETCh[:VOLTage]?

This order is used to read the latest preprocessed voltages from sampling buffer.

Order syntax

FETCh[:VOLTage][:DC]?

Return parameters

<NR2>

MEASure: POWer?

This order is used to measure the current output power value.

Order syntax

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

Return parameters

<NR2>

FETCh:POWer?

This order can read the latest power value from the sampling buffer.

Order syntax

FETCh:POWer[:DC]?

Return parameters

<NR2>

Chapter10 List Commands

LIST:FUNCtion

This order is used to select the state of list mode.

Order syntax

[SOURce:]LIST:FUNCtion <0|1>

Query syntax

[SOURce:]LIST:FUNCtion?

Return parameters

0|1

LIST:VOLTage

This order is used to set the steps of list file and voltage of current step.

Order syntax

[SOURce:]LIST:VOLTage <NRf>

Parameters

MIN to MAX

Unit

V

Example

LIST:VOLT 1, 3V //set the first step's voltage to be 3V.

Query syntax

LIST:VOLTage?

Example

LIST:VOLT? 1 //inquire the first step's voltage.

Return parameters

<NR2>

LIST:CURRent

This order is used to set the steps of list file and current value of current step.

Order syntax

[SOURce:]LIST:CURRent <NRf>

Parameters

MIN to MAX

Unit

A

Example

LIST:CURR 1,3A // set the first step's current to be 3A.

Query syntax

LIST:CURRent?

Example

LIST:CURR? 1 // inquire the first step's current.

Return parameters

<NR2>

LIST:TIMEr

This order is used to set the steps of list file and delay time of current step.

Order syntax

[SOURce:]LIST:TIMEr

Parameters

0.1 to 99999.9

Unit

S

Example

LIST:TIME 1,3

Query syntax

LIST:TIMEr?

Example

LIST:TIME? 1

Return parameters

<NR2>

LIST:SAVE

This order is used to save the list file to specified memory region.

Order syntax

[SOURce:]LIST:SAVE <NR1>

Parameters

0 to 8

Example

LIST:SAVE 1 //save the edited list file to the first set of memory region

LIST:LOAD

This order can recall the list file saved before from specified memory region. Sending query orders can inquire which group file you have recalled.

Order syntax

[SOURce:]LIST:LOAD[:IMMediate] <NR1>

Parameters

0 to 3

Example

LIST:LOAD 1

Query syntax

LIST:LOAD[:IMMediate]?

LIST:REPet

This order can set the repeat count of LIST steps.

Order syntax

[SOURce:]LIST:REPet <NR1>

Parameters

1 to 65535

Query syntax

LIST:REPet?

Chapter11 Calibration Commands

CALibration:SECure:[STATe]

Set protection mode enable or disable when calibrating the power supply.

Order syntax

CALibration:SECure:[STATe] {<ON|OFF>[<password>]}

Parameters

0|1|ON|OFF 'XXXX'

Example

CAL:SEC 1 'XXXX'; //XXXX is the instrument model

CAL:SEC OFF

Query syntax

CALibration:SECure:STATe?

CALibration:INITital

This order is used to reset the calibration parameters to default setting.

Order syntax

CALibration:INITital

Parameters

None

CALibration:SAVe

This order is used to save the calibration parameters in a nonvolatile memory.

Order syntax

CALibration:SAVe

Parameters

None

CALibration:VOLTage:LEVel

This order can set voltage calibration point. P1, P2, P3 and P4 must be calibrated orderly.

Order syntax

CALibration:VOLTage:LEVel <point>

Parameters

P1|P2|P3|P4

CALibration:VOLTage [:DATA] {<numeric value>}

Return actual output voltage value of calibration point.

Order syntax

CALibration:VOLTage [:DATA] <NRf>

Parameters

<NRf>

Example

CAL:VOLT 30.0002V

CALibration:CURRent:LEVel

This order can set current calibration point. P1、P2 must be calibrated orderly.

Order syntax

CALibration:CURRent:LEVel <point>

Parameters

P1|P2

CALibration:CURRent [:DATA] {<numeric value>}

Return actual output current value to calibration point.

Order syntax

CALibration:CURRent[:DATA] <NRf>

Parameters

<NRf>

Examples

CAL:VOLT 3.0002A

CALibration:STRing

Set the calibration information.

Order syntax

CALibration:STRing <parameters>

Parameters

The longest length is a string including 24 letters i.e related information the customer records during running the calibration process, such as the calibration

time, number of times and so on.

Example

CAL:STR 2005-1-9 20:12

CALibration:STRing?

Check original calibration information.

Query syntax

CALibration:STRing?

Return parameters

calibration information saved in power supply

Chapter12 IEEE488.2 Common Order

This chapter introduces IEEE488.2 Common Order for IT6800A/B series power supply.

*CLS

This order can clean the register as follows:

- Standard event status register
- Quest condition register
- Status byte register

Order syntax

*CLS

Parameter

None

*ESE

This order can set the parameter of standard event enable register. Setting parameter can determine which bit value of standard event register is 1 and the byte will enable ESB of status byte register is 1.

Order syntax

*ESE <NR1>

Parameter

0 to 255

Reset value

Consult *PSC order

Example

*ESE 128

Quest syntax

*ESE?

Return parameter

<NR1>

Reference order

*ESR?
*PSC
*STB?

Bit determination of standard event enable register:

Bit position	Bit Name	Bit Weight
7	PON（Power-on）	128
6	not used	-
5	CME（Command error）	32
4	EXE（Execution error）	16
3	DDE（Device-dependent error）	8
2	QYE（Query error）	4
1	not used	-
0	OPC（Operation complete）	1

*ESE?

This order can read the value of standard event status register.

Quest syntax

*ESE?

Parameter

None

Return parameter

<NR1>

*ESR?

This order can read the value of standard event status register. After executing this order, standard event status register is reset. Bit definition of standard event status register is as the same as the standard event status enable register.

Quest syntax

*ESR?

Parameter

None

Return parameter

<NR1>

Reference order

*CLS
*ESE
*ESE?

*OPC

*IDN?

This order can read information about power supply. The parameter it returns contains 4 segments divided by comma.

Quest syntax

*IDN?

Parameter

None

Return parameter

<AARD>

Example

ITECH,6800A,00000000000004,V1.01-V1.00

*OPC

When all orders before this order are executed, OPC is 1 for the standard event status register. Sending query order will return 1 to output buffer.

Order syntax

*OPC

Parameter

None

Quest syntax

*OPC?

Return parameter

<NR1>

*PSC

This order control if power supply send a query or not when it is reset.

Order syntax

*PSC

Parameter

1

Quest syntax

*PSC?

Reference order

*ESE

*SRE

STAT:OPER:ENAB

STAT:QUES:ENAB

*RST

This order reset the power supply to default setting.

Order syntax

*RST

Parameter

None

*SRE <enabled value >

This order can set the parameter of status bit enable register. When query status bit enable register, the power will return a decimal number, this number is the binary weighted of enable register.

Order syntax

*SRE <NRf>

Parameter

0~255

Reset value

Consult *PSC order

Example

*SRE 128

Quest syntax

*SRE?

Return parameter

<NR1>

Reference Order

*ESE

*ESR?

*PSC

*STB?

*STB?

This order can read the data from status byte register. This order is similar to a statistics of series, but the equivalent of another instrument orders, it returns the value the same as series statistics, but after this order is executed, the bit 6 value of status byte register is cleared, while the status bit will not be cleared when system statistics implemented.

Quest syntax

*STB?

Parameter

None

Return parameter

<NR1>

Reference order

*CLS

*ESE

*ESR

*TRG

When the trigger mode of the power supply is BUS order trigger mode, the order will generates a trigger signal.

Order syntax

*TRG

Parameter

none

*SAV

This order can save the parameters of power supply to register.

Order syntax

*SAV <NRf>

Parameter

1 to 71

*RCL

This order can recall the parameter you saved before from the register.

Order syntax

*RCL <NRf>

Parameter

1 to 71

*TST?

This command is used to query the result of the instrument self-test. 0 represents a successful self-test while other returned parameters means failure. What's more, the instrument will generate error information explaining the cause of the failure.

Order syntax

*TST?

Parameter

None

Returned parameter

<NR1>

STATus:QUEStionable[:EVENt]?

This order can read the parameter from quest event register. The power supply will return a decimal number which is corresponding to the binary weighted sum of each bit of the register, these bits have been latched. After executing, quest event register is reset.

Quest syntax

STATus:QUEStionable[:EVENt]?

Parameter

None

Return parameter

<NR2>

Reference order

STATus:QUEStionable:ENABle

Bit determination of quest event register:

Bit Position	Bit Name	Bit Value	Bit Position	Bit Name	Bit Value
15	no use	-	7	no use	-
14	no use	-	6	no use	-
13	no use	-	5	no use	-
12	no use	-	4	OT	16
11	no use	-	3	OP	8

10	no use	-	2	no use	-
9	no use	-	1	OC	2
8	no use	-	0	OV	1

STATus:QUEStionable:CONDition?

This order can be used to read the value of query condition register and to get the status of the power, if it is CC or CV. The power supply will return a decimal number corresponding to the binary-weighted sum of each bit of the register. These bits are not latched. If 0 is returned, then the output status of the power is OFF or uncertain. If 1 is returned, then the status is CV mode, if 2 is returned, then the status is CC. If 3 is returned, then there is error occurred.

Quest syntax

STATus:QUEStionable:CONDition?

Parameter

None

Return parameter

<NR2>

STATus:QUEStionable:ENABle

This order can set the parameter of quest event enable register. The power supply will return a decimal number which is the binary-weighted sum of the enable register.

Order syntax

STATus:QUEStionable:ENABle <NR2>

Parameter

0 to 255

The high position 256 to 65535 is empty.

Reset value

Consult *PSC order

Example

STATus:QUEStionable:ENABle 128

Quest syntax

STATus:QUEStionable:ENABle?

Return parameter

<NR2>

Chapter13 Error Information

Error Code List

Error code	Error information	Example
110	"No input command"	Send empty command.
120	"Parameter overflowed"	The setting value exceeds the range. For example, CURRent 100.0
130	"Wrong units for parameter"	For example, CURRent 5.0V, the unit is wrong.
140	"Wrong type of parameter"	For example, OUTPut:TIMer 100001.0.
150	"Wrong number of parameter"	For example, CURRent 5.0,6.
160	"Unmatched quotation mark"	For example, CALibrate:SECure 0,"6831'.
165	"Unmatched bracket"	For example, CURRent (5.
170	"Invalid command"	The command is invalid or the format is wrong, including the spelling problems. For example, CUR 5.0.
180	"No entry in list"	The command needs three parameters while the program only processes two commands practically.
191	"Too many char"	The sending data length is over 256 while using serial port or USB.
-200	"Execution error"	The command execution is invalid due to wrong command status or wrong setting. For example, sending *TRG command must be under the BUS trigger method. If the trigger method is MANUAL, *TRG command is invalid.
-310	"System error"	Undetermined error code.
-350	"Too many errors"	Too much error in the stack.
-410	"Query INTERRUPTED"	While using USB communication, no parameters is returned after sending query command, the instrument will prompt an error. Some status or parameter setting command need to be followed by a query command i.e. you need to send the query command after sending this setting command, if you send another setting command after sending this setting command, the instrument will prompt an error.
-430	"Query DEADLOCKED"	While using USB communication, if you send query command when the query region is full, the instrument will prompt an error.
0	"No error"	No error.
2	"Mainframe Initialization Lost"	During the start-up program initialization, if the instrument can't read the user setting parameters and data from EEPROM, the system will prompt an

		error.
3	"Module Calibration Lost"	During the start-up program initialization, if the instrument can't read the calibration data from EEPROM, the system will prompt an error.
4	"Eeprom failure"	EEPROM data is stored unsuccessfully.
6	"Output Locked"	While using the dual-range instrument, you can't turn the output on in 1s after switching the range, otherwise the system will prompt an error.
40	"Flash write failed"	-
41	"Flash erase failed"	-
217	"RS-232 receiver parity"	The odd and even parity of RS-232 communication is wrong.
223	"Front panel buffer overrun"	The communication buffer is overflow while using serial port.
224	"Front panel timeout"	The communication is overtime while using serial port.
402	"CAL password is incorrect"	While sending calibration command, if the password is wrong and you can't enter the calibration mode, the system will prompt an error. For example, for model IT6831A, the correct calibration command is CALibrate:SECure 0,"6831", if you send CALibrate:SECure 0,"6811", the system will prompt an error.
403	"CAL not enabled"	If you haven't switched into calibration mode, sending related calibration command will cause an error. For example, if you haven't switched into calibration mode, sending CALibrate:CURRent:LEVel P1 command will cause an error.
404	"readback cal are incorrect"	If you don't calibrate the instrument according to the function requirement, the system will prompt an error. For example, while calibrating one current point, the setting current value is over the maximum range, an error will occur.
405	"programming cal are incorrect"	If you don't calibrate the instrument according to the function requirement, the system will prompt an error. For example, if you don't send CALibrate:CURRent 0.1 after sending CALibrate:CURRent:LEVel P1, then send CALibrate:CURRent:LEVel P2 directly, an error will occur.

Contact US

Thank you for purchasing ITECH products. If you have any doubt about this product, please contact us as follow.

1. Please refer to the CD-ROM of related user's manual in package.
2. Visit ITECH website www.itechate.com .
3. Select the most convenient contact for further consultation.