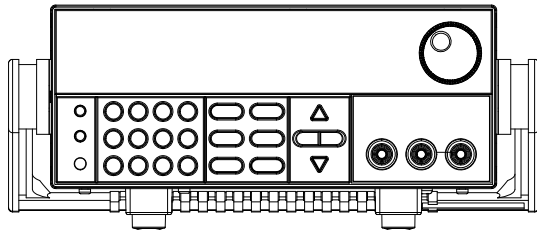


# Programmable HV Power Supply

## Series IT6700 Programming Guide



---

Model: IT6722/IT6722A/IT6723/IT6723B/  
IT6723C/IT6723G/IT6723H/IT6724/  
IT6724B/IT6724C/IT6724G/IT6724H/  
IT6726B/IT6726C/IT6726G/IT6726H/  
IT6726V

Version: V2.0

## Statement

© Itech Electronics, Co., Ltd. 2014  
No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior permission and written consent from Itech Electronics, Co., Ltd. as governed by international copyright laws.

### Manual Article No.

IT6700-402161

### Revision

Revision 2, published on

April 14, 2015

Itech Electronics, 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 *talīs qualīs* 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

IT6700 series power supply 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

	Direct current		ON (power)
	Alternating current		OFF (power)
	Both direct and alternating current		Power-on state
	Protective earth (ground) terminal		Power-off state
	Earth (ground) terminal		Reference terminal
	Caution		Positive terminal
	Warning (refer to this manual for specific Warning or Caution information)		Negative terminal
	A chassis terminal	-	-

## 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 therefrom.

### WARNING

- The power supply is provided with a three-core power cord during delivery and should be connected to a three-core junction box. Before operation, be sure that the power supply 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 power supply 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 IT6700 series power supply 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	Operating up to 2,000 meters
Installation category	II
Pollution degree	Pollution degree 2


**NOTE**

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

## Regulation tag

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

## Waste electrical and electronic equipment (WEEE) directive



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

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

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

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

## Compliance Information

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

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

Conforms with the following product standards:

### EMC Standard

IEC 61326-1:2012/ EN 61326-1:2013 <sup>123</sup>

#### Reference Standards

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

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

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

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

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

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

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

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

### Safety Standard

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

## Content

Certification and Quality Assurance .....	1
Warranty service .....	1
Limitation of Warranty .....	1
Safety signs .....	1
Safety precautions .....	2
Environmental conditions .....	2
Regulation tag .....	3
Waste electrical and electronic equipment (WEEE) directive .....	3
Compliance Information .....	4
<b>Chapter1 Remote Control .....</b>	<b>1</b>
1.1 Overview .....	1
1.2 SCPI Command Introduction .....	1
1.3 Command Type of SCPI .....	1
1.4 Command Format .....	3
1.5 Data Type .....	5
1.6 SCPI status register .....	6
1.7 Remote Operation .....	8
<b>Chapter2 IEEE488.2 Commands .....</b>	<b>12</b>
*CLS .....	12
*ESE .....	12
*ESR? .....	13
*IDN? .....	13
*OPC .....	13
*PSC .....	14
*RST .....	14
*SRE <enabled value> .....	14
*STB? .....	15
*TRG .....	15
*SAV .....	16
*RCL .....	16
<b>Chapter3 Status commands .....</b>	<b>17</b>
STATus:QUESTionable[:EVENT]? .....	17
STATus:QUESTionable:CONDition? .....	17
STATus:QUESTionable:ENABle <enabled value> .....	17
SYSTem:VERSion? .....	18
SYSTem:ERRor? .....	18
SYSTem:REMote .....	19
SYSTem:LOCal .....	19
SYSTem:RWLock .....	19
SYSTem:BEEPer .....	20
SYSTem:COMMunicate:GPIB:RDEVice:ADDRes .....	20
SYSTem:INTerface .....	20
<b>Chapter4 Display Commands .....</b>	<b>21</b>
DISPlay .....	21
DISPlay:TEXT < Reference value> .....	21
DISPlay:TEXT:CLEar .....	21
<b>Chapter5 Trigger Commands .....</b>	<b>22</b>
TRIGger .....	22
TRIGger:SOURce .....	22
<b>Chapter6 Output Commands .....</b>	<b>23</b>
OUTPut .....	23
OUTPut:TIMer .....	23

OUTPut:TIMer:DATA.....	23
CURRent {<current value> MINimum MAXimum UP DOWN DEF} .....	24
CURRent:STEP .....	24
CURRent:TRIG{<current value> MINimum MAXimum} .....	25
VOLTage {<volts> MINimum MAXimum UP DOWN DEF} .....	25
VOLTage:STEP {<value> DEFault} .....	26
VOLT:TRIG{<volts> MINimum MAXimum}.....	27
VOLT:PROTection {<volts> MINimum MAXimum} .....	27
VOLT:PROTection:STATe {0 1 OFF ON} .....	28
VOLT:PROTection:TRIPed? .....	28
VOLT:PROTection:CLEar .....	28
VOLT:LIMIT <volts> .....	29
APPLY {<voltage value> DEF MIN MAX} [, {<current value> DEF MIN MAX}] .....	29
<b>Chapter7 Measurement Commands.....</b>	<b>30</b>
MEASure:CURRent? .....	30
FETCh:CURRent? .....	30
MEASure[:VOLTage]? .....	30
FETCh[:VOLTage]? .....	30
MEASure: POWer? .....	31
FETCh:POWer? .....	31
<b>Chapter8 List Commands.....</b>	<b>32</b>
LIST:FUNCTion .....	32
LIST:VOLTage .....	32
LIST:CURRent .....	32
LIST:TIMer .....	33
LIST:SAVE .....	34
LIST:LOAD.....	34
LIST:REPet .....	34
<b>Chapter9 Calibration Commands.....</b>	<b>35</b>
CALibration:SECure:[STATe] .....	35
CALibration:INITital .....	35
CALibration:SAVE .....	35
CALibration:VOLTage:LEVel.....	35
CALibration:VOLTage [:DATA] {<numeric value>} .....	36
CALibration:CURRent:LEVel.....	36
CALibration:CURRent[:DATA] {<numeric value>}.....	36
CALibration:STRing.....	36
CALibration:STRing?.....	37
CALibration:INITital .....	37



# 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}

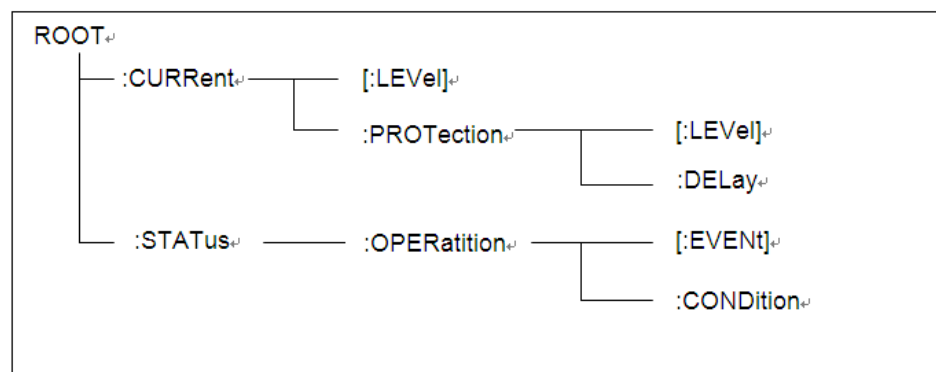
POLarity {NORMal|INVerted}

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 subsysteme 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 command subsystem tables in Section 5 provide the in the long-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 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:

DISPly {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:

DISPly: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..."

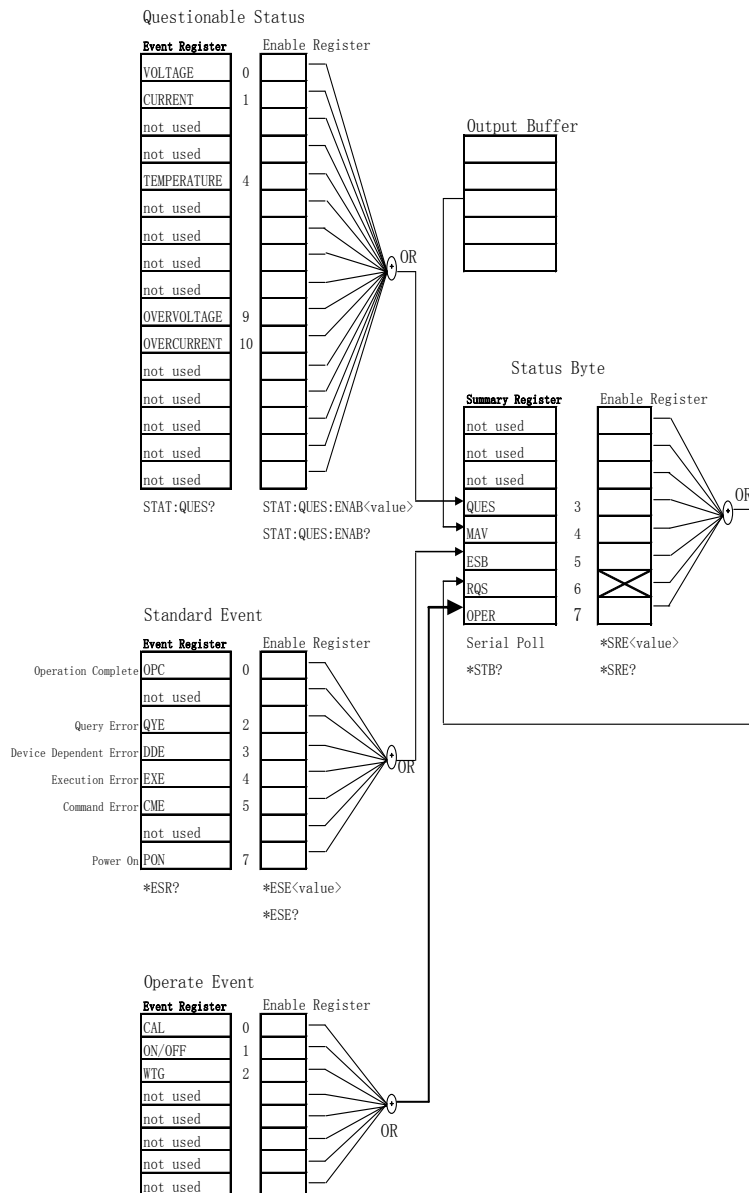
Single quotation marks may also be used to display the same message.

DISP:TEXT 'WAITING...'

## 1.6 SCPI status register

IT6700H series power records variable instrument status via three kinds of status register, the three kinds of registers are standard event register, query status register and status bytes register. Status byte register records the information of other status registers.

The below picture will give you more detailed information:



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	Definition
0	OPC Operation Complete. All orders prior to and including an *OPC order have been executed.
1	Not Used Always set to 0.
2	QYE Query Error. The power supply tried to read the output buffer but it was empty. Or, new order line was received before a previous query had been read. Or, both the input and output buffers are full.

3	DDE	Device Error. A self-test or calibration error occurred (see error numbers 601 through 750 in chapter 5).
4	EXE	Execution Error. An execution error occurred (see error numbers -211 through -224 in chapter 5).
5	CME	Order Error. A order syntax error occurred (see error numbers -101 through -178 in chapter 5).
6	Not Used	Always set to 0.
7	PON	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	Definition
0	Voltage The power supply is/was in the constant current mode.
1	Current The power supply is/was in the constant voltage mode.
2-3	Not used Always set to 0.
4	Over temperature The fan has a fault condition.
5-8	Not used Always set to 0.
9	Over voltage The overvoltage protection circuit has tripped.
10	Over Current The over current protection circuit has tripped.
11-15	Not used 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 corresponding bit in status byte register will subsequently be changed.

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

## 1.7 Remote Operation

IT6700 series power supply is provided with three communication interfaces to communicate with a computer for selection, including S232, USB, RS485 and GPIB. IT6722A has not standard GPIB communication interface.

### RS232 interface


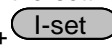
There is a DB9 connector at the rear of the power supply, when connect to computer, you need to select a cable with COM port on both side; to active




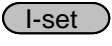
communication, you need to enable the settings in menu to be the same with the PC communication configuration.



**NOTE**

The RS232 settings must match the settings in front panel system information. If any change, please press  (Shift)+  key to modify the menu: SYST SETCOMM.

**RS-232 data format**

RS-232 data is a 10-bit words which has a start bit and a stop bit. The start bit and stop bit can't be edited. However, you can select the parity items with  (Shift) +  key on the front panel.

Parity options are stored in nonvolatile memory.

**Baud Rate**

The front panel  (Shift) +  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. Below Table 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).



RS232 Pins of Plug

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

**RS-232 Troubleshooting:**

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

- 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).
- 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.
- 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 of 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

Local Address: (0 ~ 31, the factory default setting is 0)

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

## 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 Receiver 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:

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

## RS485 interface

IT6726C power supply, via the RS485 interface, provides multi-unit control function for up to 30 units (If connecting more than 10 units, add a 120Ω resistor terminator to the last unit as shown in the figure below). Access the menu tree Menu → SYST SET → COMM → RS485 to set the RS485 settings.

**User can set the following parameters of the RS485 interface:**

Baud Rate: 9600(4800/9600/19200/38400/57600/115200)

Parity and data bit: NONE/8BIT、EVEN/8BIT、ODD/8BIT

MODE: SIGNAL、MUX

SIGNAL (single-unit mode)





MUX (multi-unit mode), ADDR(Address), (0~30, the factory default setting is 0)

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

To set the multi-unit connection mode, access the menu tree MENU → SYST SET → COMM → RS485 → MODE → MUX to choose the multi-unit mode.

Set each unit with a different Address (0 ~ 30). Then by using RS485 connect the first power supply in the chain to a PC. Now, multiple units daisy-chained via RS485 can be controlled by one PC by using the commands specific for multi-unit connection. See “Programming Guide” section for details.

## GPIB interface

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)+  key to enter the system menu function, find the GPIB address setting by  button, type the address,  key to confirm. GPIB address is stored in nonvolatile memory line.

---

## Chapter2 IEEE488.2 Commands

---

### **\*CLS**

This order can clean the register as follows:

- Standard event status register
- Quest condition register
- Operation event register
- Status byte register
- Error code

#### **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:**

0~255

#### **Reset value:**

Consult \*PSC order

#### **Example:**

\*ESE 128

#### **Quest syntax:**

\*ESE?

#### **Return parameter:**

<NR1>

#### **Reference order:**

\*ESR? \*PSC \*STB?

## \*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>

### For example:

ITECH Ltd, IT6700H, 0123456789AF, 1.00

## \*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.

**1 OR ON:** When power supply is reset, operation event enable register, query event enable register and standard event status register are all reset.

**0 OR OFF:** The data of status byte register, operation event enable register, quest event enable register and standard event status enable register is stored in nonvolatile register, and is recalled when power supply is reset.

**Order syntax:**

\*PSC <bool>

**Parameter:**

0|1|ON|OFF

**Quest syntax:**

\*PSC?

**Return parameter:**

0|1

**Reference order:**

\*ESE \*SRE STAT:OPER:ENAB STAT:QUES:ENAB

**\*RST**

This order reset the power supply to default setting.

CURR	CURR:STEP	CURR:TRIG	CURR:PROT
DISP ON	OUTP OFF	TRIG:SOUR BUS	VOLT 0V
VOLT:STEP	VOLT:TRIG	VOLT:PROT	VOLT:PROT:STAT
ON			

**Order syntax:**

\*RST

**Parameter:**

None

**\*SRE <enabled value>**

This order can set the parameter of standard event 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. The function is the same as **TRIGger[:IMMediate]**.

**Order syntax:**

\*TRG

**Parameter:**

none

**Reference order:**

TRIG TRIG:SOUR

**\*SAV**

This order can save the parameters of power supply to register. These parameter contains:

CURR	CURR:STEP	CURR:TRIG	CURR:PROT	DISP
OUTP	TRIG:SOUR			
VOLT	VOLT:STEP	VOLT:TRIG	VOLT:PROT	
VOLT:PROT:STAT				

**Order syntax:**

\*SAV &lt;NRf&gt;

**Parameter:**

1~72

**Example:**

\*SAV 3

**Reference order:**

\*RCL

**\*RCL**

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

**Order syntax:**

\*RCL &lt;NRf&gt;

**Parameter:**

1~72

**Example:**

\*RCL 3

**Reference order:**

\*SAV



---

## Chapter3 Status commands

---

### **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:**

<NR1>

#### **Reference order:**

STATus:QUEStionable:ENABLE

### **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 CC mode, if 2 is returned, then the status is CV. If 3 is returned, then there is error occurred.

#### **Quest syntax:**

STATus:QUEStionable: CONDition?

#### **Parameter:**

None

#### **Return parameter:**

<NR1>

### **STATus:QUEStionable:ENABLE <enabled value>**

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 <NRf>

**Parameter:**

0~255

**Reset value:**

Consult \*PSC order

**Example:**

STATus:QUEStionable:ENABLE 16

**Quest syntax:**

STATus:QUEStionable:ENABLE?

**Return parameter:**

<NR1>

**Reference order:**

\*PSC

## 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:**

SYST:VERS?

**Parameter:**

None

**Return parameter:**

<NR2>

**Example:**

1.00, 1991.1

## 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 20 groups of error message. Send the order once will read one error message from the error queue.

1. 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.

2. If more than 20 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”.

3. 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

**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:**

SYST:REM

**Parameter:**

None

**Quest syntax:**

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:**

SYST:LOC

**Parameter:**

None

**Quest syntax:**

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 too. 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.

**Order syntax:**

SYST:RWL

**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[:IMMEDIATE]

**Example:**

SYST:BEEP

**SYSTem:COMMunicate:GPIB:RDEvice:ADDRess**

This order is used to set the GPIB communication address.

**Order Syntax:**

SYSTem:COMMunicate:GPIB:RDEvice:ADDRess <NR1>

**Parameter:**

0-30

**Quest syntax:**

SYSTem:COMMunicate:GPIB:RDEvice:ADDRess?

**Returned parameter:**

<NR1>

**SYSTem:INTerface**

This order is used to switch the communication interface.

**Order syntax:**

SYSTem:INTerface <GPIB|USB|RS232>

**Parameter:**

GPIB|USB|RS232

---

## Chapter4 Display Commands

---

### DISPlay

This order is used to turn on/off the VFD display. When the display is off, the output and will not be displayed on the VFD, at the same time, all the LED except for ERROR led will not be off. When the control mode is local, the display will turn on automatically, press LOCAL button can switch the control mode from remote to local.

#### Order syntax:

DISPlay[:WINDow][:STATe] <bool>

#### Parameter:

0|1|OFF|ON

#### Example:

DISPlay 1

#### quest syntax:

DISPlay?

#### Returned parameter:

0|1

### DISPlay:TEXT <Reference value>

This order is used to display a text on the VFD. TA message can display up to 12 characters, extra characters will be ignored. Commas, periods and semicolons will not be treated as a separate character but will be contained to the character before and display.

#### Order syntax:

DISPlay[:WINDow]:TEXT[:DTAT]

#### Quest syntax:

DISPlay:TEXT?

### DISPlay:TEXT:CLEAr

This order is used to clear the text displayed on the front panel.

#### Order syntax:

DISPlay[:WINDow]:TEXT:CLEAr

---

## Chapter5 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:

TRIG:SOUR <mode>

#### Parameters:

BUS|MANUAL

#### Query syntax:

TRIGger:SOURce?

#### \*RST value:

MANUAL

---

## Chapter6 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:**

OUTP[:STATe] <bool>

**Parameters:**

0|1|OFF|ON

**Query syntax:**

OUTPut?

**\*RST value:**

0|OFF

### 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:**

0-99999.9

**Unit:**

S

**Query syntax:**

OUTPut:TIMer:DATA?

**Parameters:**

<NRf>

**CURRent {<current value>|MINimum|MAXimum|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, 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". If adjusted value has exceeded the current range, then error information will be return:error-222.

**Oder syntax:**

[SOUR:]CURR[:LEVel][:IMMediate][:AMPLitude] <NRf>

**Parameters:**

MIN to MAX

**Unit:**

A

**\*RST value:**

MIN

**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~MAX

**\*RST value:**

the stepping resolution of current of this unit.

**Query order:**

CURRent:STEP?

**Return parameters:**

<NR2>

**CURRent:TRIG{<current value>|MINimum|MAXimum}**

This order is used to set a current to be triggered. At first, this current value is saved and only when receiving a trigger signal will the power supply output with this current value. Sending order CURRent will not affect the setting value in this order. Sending query order, unit will return the previous setting value. If customer does not utilize order **CURRent:TRIG** at the beginning, then the return parameters will be the value you set with order CURRent. CURR:TRIG? MAX or CURR:TRIG? MIN orders will let you know the max and min current you can set.

**Order syntax:**

[SOURce:]CURRent[:LEVel]:TRIGgered[:IMMEDIATE][:INCRement] <NRf>

**Parameters:**

MIN to MAX

**Unit:**

A

**Query syntax:**

CURRent:TRIG? [MINimum|MAXimum]

**Return parameters:**

<NR2>

**VOLTage**

**{<volts>|MINimum|MAXimum|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 inquire 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". If adjusted value has exceeded the voltage range, then error information will be return:error-222.

**Order syntax:**

```
[SOUR:]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 be 0.02V
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~MAX

**\*RST value:**

the stepping resolution of voltage of this unit.

**Query order:**

VOLT:STEP?

**Return parameters:**

<NR2>

**VOLT:TRIG{<volts>|MINimum|MAXimum}**

This order is used to set a voltage to be triggered. At first, this voltage value is saved and only when receiving a trigger signal will the power supply output with this voltage value. Sending order VOLTage will not affect the setting value in this order. Sending query order, unit will return the previous setting value. If customer does not utilize order **VOLT:TRIG** at the beginning, then the return parameters will be the value you set with order CURRent. CURR:TRIG? MAX or CURR:TRIG? MIN orders will let you know the max and min voltage you can set.

**Order syntax:**

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

**Parameters:**

MIN to MAX

**Unit:**

V

**Query syntax:**

VOLT:TRIG? [MINimum|MAXimum]

**Return parameters:**

<NR2>

**VOLT:PROTection {<volts>|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:**

&lt;NR2&gt;

**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~MAX

### **Unit:**

V

### **Query syntax:**

VOLTage:LIMIT?

### **Return parameters:**

<NR2>

## **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~MAX

### **Unit:**

V, A

### **Query syntax:**

APPLy?

### **Return parameters:**

<NR2>

---

## Chapter7 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 need 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>

---

## Chapter8 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~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~MAX
```

**Unit:**

```
A
```

**Example:**

```
LIST:CURR 1, 2A // 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. This order is used to set delay time of every steps.

**Order syntax:**

```
[SOURce:]LIST:TIMER
```

**Parameters:**

```
MIN~MAX|MIN|MAX
```

**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:

0~9

### 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:

0~9

### 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>
```

### Query syntax:

```
LIST:REPET?
```

---

## Chapter9 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, '5811

#### **Example:**

CAL:SEC 1, '5811; CAL:SEC OFF

#### **Query syntax:**

CALibration:SECure:STATe?

#### **Parameters:**

none

### **CALibration:INITial**

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

#### **Order syntax:**

**CALibration: INITial**

#### **Parameters:**

none

### **CALibration:SAVE**

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

#### **Order syntax:**

**CALibration:INITial**

#### **Parameters:**

none

### **CALibration:VOLTage:LEVel**

This order can set voltage calibration point. P1、P2、P3、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 he running the calibration process, such as the calibration time and 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.

**CALibration:INITital**

This order is used to reset parameters to factory default.

**Order syntax:**

CALibration: INITital

**Parameters:**

none

## **Contact Us**

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

1. Refer to accompanying data disk and relevant manual.
2. Visit ITECH website: [www.itechate.com](http://www.itechate.com)
3. Select the most convenient contact method, for further information.