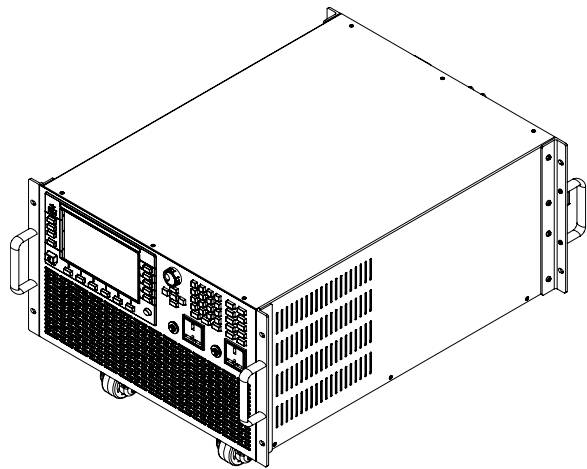


Programmable AC Power Supply

IT7600 Series Programming Guide



Model: IT7622/IT7624/IT7625/IT7626/IT7627/
IT7628/IT7628L/IT7630/IT7632/IT7634/IT7636
Version: V1.2

Notices

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

Manual Part Number

IT7600-402573

Revision

1st Edition: Apr. 19, 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 warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. ITECH shall not be held liable for errors or for incidental or indirect damages in connection with the furnishing, use or application of this document or of any information contained herein. Should ITECH and the user enter into a separate written agreement with warranty terms covering the materials in this document that conflict with these terms, the warranty terms in the separate agreement shall prevail.

Technology Licenses

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

Restricted Rights Legend

Restricted permissions of the U.S. government. Permissions for software and technical data which are authorized to the U.S. Government only include those for custom provision to end users. ITECH provides this customary commercial license in software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data – Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

Safety Notices

CAUTION

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

WARNING

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



NOTE

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

Quality Certification and Assurance

We certify that IT7600 series 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 if the product is:

- Damaged resulting from customer-wired circuits or customer-supplied parts or accessories;
- Modified or repaired by customer without authorization;
- Damaged resulting from customer-wired circuits or use in an environment not designated by us;
- 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)
	Alternating current		OFF (power)
	Both direct and alternating current		Power-on state
	Chassis (earth ground) symbol.		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

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 power line during delivery and should be connected to junction box. Before operation, be sure that the power supply 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 of electronic load without overheating. If there are multiple loads, each pair of the load power cord must be carry out the full rated short-circuit output current of the power securely.
- 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~40°C
Operating humidity	20%~80%(non-condensation)
Storage temperature	-10°C~70 °C
Altitude	Operating up to 2,000 meters
Installation category	II
Pollution degree	Pollution degree 2



Note

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

Regulatory Markings

	<p>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.</p>
	<p>The instrument 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.</p>
	<p>This symbol indicates the time period during which no hazardous or toxic substances are expected to leak or deteriorate during normal use. The expected useful 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.</p>

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 1 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 ¹²³

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

Quality Certification and Assurance	i
Warranty	i
Limitation of Warranty.....	i
Safety Symbols.....	i
Safety Precautions	ii
Environmental Conditions.....	iii
Regulatory Markings.....	iii
Waste Electrical and Electronic Equipment (WEEE) Directive.....	iii
Compliance Information.....	v
Chapter1 Remote Control.....	1
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 Communication Interface.....	6
Chapter2 System Commands	10
STATus:QUEStionable[:EVENT]?	10
STATus:QUEStionable:ENABLE.....	10
STATus:QUEStionable:PTRansition	11
STATus:QUEStionable:NTRansition.....	11
STATus:QUEStionable:CONDition?	12
STATus:OPERation[:EVENT]?	12
STATus:OPERation:ENABLE.....	13
STATus:OPERation:PTRansition.....	13
STATus:OPERation:NTRansition	14
STATus:OPERation:CONDition?	14
SYSTem:BEEPer:IMMediate	14
SYSTem:BEEPer[:STATe].....	15
SYSTem:VERSion?	15
SYSTem:ERRor?	15
SYSTem:CLEar	16
SYSTem:REMote.....	16
SYSTem:LOCal	16
SYSTem:RWLock	17
SYSTem:DATE	17
SYSTem:TIME	18
SYSTem:COMMunicate:GPIB[:SELF]:ADDRess	18
SYSTem:COMMunicate:LAN:CURRent:ADDRess	19
SYSTem:COMMunicate:LAN:CURRent:DGATeway	19
SYSTem:COMMunicate:LAN:CURRent:SMASK	19
SYSTem:COMMunicate:LAN:DHCP[:STATe].....	20

SYSTem:COMMunicate:LAN:SOCKetport	20
SYSTem:COMMunicate:LAN:MACAddress?	21
SYSTem:COMMunicate:SELEct	21
SYSTem:COMMunicate:SERial:BAUDrate	22
SYSTem:BACKlight:BRIGHtness	22
SYSTem:POSetup	22
SYSTem:DEVice:NUM	23
SYSTem:MEASure:MODE	23
SYSTem:SOURce:MODE	24
SYSTem:TRIG:SOURce	24
SYSTem:RUN:NETWork?	24
SYSTem:RUN:DEVNum?	25
SYSTem:RUN:GRPNum?	25
Chapter3 Measurement Commands	26
FETCh[:SCALar]:VOLTage?	26
FETCh[:SCALar]:CURRent?	26
FETCh[:SCALar]:POWer[:REAL]?	26
FETCh[:SCALar]:POWer:APParent?	26
FETCh[:SCALar]:POWer:PFACTOR?	27
FETCh[:SCALar]:FREQUency?	27
FETCh[:SCALar]:CFACtor?	27
FETCh[:SCALar]:CURRent:PEAK:PLUS?	28
FETCh[:SCALar]:CURRent:PEAK:MINUs?	28
FETCh[:SCALar]:CURRent:ISURge?	28
MEASure[:SCALar]:VOLTage?	29
MEASure[:SCALar]:CURRent?	29
MEASure[:SCALar]:POWer[:REAL]?	29
MEASure[:SCALar]:POWer:APParent?	29
MEASure[:SCALar]:POWer:PFACTOR?	30
MEASure[:SCALar]:FREQUency?	30
MEASure[:SCALar]:CFACtor?	30
MEASure[:SCALar]:CURRent:PEAK:PLUS?	31
MEASure[:SCALar]:CURRent:PEAK:MINUs?	31
MEASure[:SCALar]:CURRent:ISURge?	31
FETCh?	31
MEASure?	32
Chapter4 Vector Commands	33
VECTor:ORDER	33
VECTor:DATA?	33
VECTor:TYPE	34
Chapter5 Harmonic Commands	35
MEASure[:SCALar]:HARMonics?	35
MEASure[:SCALar]:VOLTage:HARMonics[:AMPLitude]?	35

MEASure[:SCALar]:VOLTage:HARMonics:DISort?	35
MEASure[:SCALar]:PHASe:HARMonics?	36
MEASure[:SCALar]:FREQUency:HARMonics:FUNDamental?	36
FETCh[:SCALar]:HARMonics?	36
FETCh[:SCALar]:VOLTage:HARMonics[:AMPLitude]?	37
FETCh[:SCALar]:VOLTage:HARMonics:DISort?	37
FETCh[:SCALar]:FREQUency:HARMonics:FUNDamental?	38
Chapter6 Oscilloscope Commands.....	39
WAVE:TRIGger:MODE	39
WAVE:TRIGger:MODE?	39
WAVE:TRIGger:SLOPe	39
WAVE:TRIGger:SLOPe?	40
WAVE:TRIGger:SOURce	40
WAVE:TRIGger:SOURce?	40
WAVE:TRIGger:VOLTage[:LEVel]	40
WAVE:TRIGger:VOLTage[:LEVel]?	41
WAVE:TRIGger:CURRent[:LEVel].....	41
WAVE:TRIGger:CURRent[:LEVel]?	42
WAVE:TRIGger:DELay:TIME	42
WAVE:TRIGger:DELay:TIME?	42
WAVE:VOLTage:DATA[:NORMAlization]?	43
WAVE:CURRent:DATA[:NORMAlization]?	43
WAVE:AUTO.....	43
WAVE:SINGle	44
WAVE:RUN	44
WAVE:STOP	44
WAVE:TRIGger[:STATe]?	44
WAVE:SCOPE:DIVTime	45
WAVE:SCOPE:DIVTime?	45
WAVE:SCOPE:SELection	45
WAVE:SCOPE:SELection?	46
WAVE:KNOB:SELection	46
WAVE:KNOB:SELection?	46
WAVE:VOLTage:BASE	47
WAVE:VOLTage:BASE?	47
WAVE:CURRent:BASE	47
WAVE:CURRent:BASE?	48
WAVE:VOLTage:RANGe	48
WAVE:VOLTage:RANGe?	49
WAVE:CURRent:RANGe.....	49
WAVE:CURRent:RANGe?.....	49
WAVE:ADDR	50
WAVE:ADDR?	50
Chapter7 Self-defined wave Commands.....	51

SELFdefine:NAME	51
SELFdefine:RECALL:NAME?	51
SELFdefine:SAVE	51
SELFdefine:DATA	51
SELFdefine:RECALL	52
SELFdefine:RECALL:NUM?	52
Chapter8 Output Commands	53
[SOURCE:]OUTPut[:STATe]	53
[SOURCE:]OUTPut[:STATe]?	53
[SOURCE:]RELAY:MODE	53
[SOURCE:]RELAY:MODE?	54
[SOURCE:]RANGe	54
[SOURCE:]RANGe?	54
[SOURCE:]PROTection:CLEar	55
[SOURCE:]PROTection?	55
[SOURCE:]LOOPspeed	55
[SOURCE:]LOOPspeed?	56
[SOURCE:]DEVIce:TYPE	56
[SOURCE:]DEVIce:TYPE?	56
[SOURCE:]NORMal:VOLTage:AC[:LEVel][:IMMediate][:AMPLitude]	57
[SOURCE:]NORMal:VOLTage:AC:LIMit[:LEVel]	57
[SOURCE:]NORMal:VOLTage:DC[:LEVel][:IMMediate]	58
[SOURCE:]NORMal:VOLTage:DC:LIMit[:LEVel]	58
[SOURCE:]NORMal:FREQUency[:LEVel][:IMMediate]	59
[SOURCE:]NORMal:FREQUency:LIMit[:LEVel]	59
[SOURCE:]NORMal:PHASe:STARt[:LEVel][:IMMediate]	59
[SOURCE:]NORMal:PHASe:STARt:LIMit[:LEVel]	60
[SOURCE:]NORMal:PHASe:STOP[:LEVel][:IMMediate]	60
[SOURCE:]NORMal:PHASe:STOP:LIMit[:LEVel]	61
[SOURCE:]NORMal:MODE	61
[SOURCE:]NORMal:WAVE	62
[SOURCE:]NORMal:THD:ORDer:DATA	62
[SOURCE:]NORMal:THD:DOWNload	62
Chapter9 STEP Commands	64
STEP:VOLTage:STARt	64
STEP:VOLTage:STARt?	64
STEP:VOLTage:STOP	64
STEP:VOLTage:STOP?	65
STEP:VOLTage:INTERval	65
STEP:VOLTage:INTERval?	65
STEP:MODE	66
STEP:MODE?	66
STEP:FREQ:STARt	66
STEP:FREQ:STARt?	67

STEP:FREQ:STOP.....	67
STEP:FREQ:STOP?.....	67
STEP:FREQ:INTERval	67
STEP:FREQ:INTERval?	68
STEP:TIME:INTERval.....	68
STEP:TIME:INTERval?.....	68
STEP:PRiority<VOLT FREQ>	69
STEP:PRiority?	69
STEP:REPetition	70
STEP:REPetition?	70
Chapter10 Protection commands.....	71
PROTect:RMS:CURRent.....	71
PROTect:RMS:CURRent?.....	71
PROTect:RMS:TIME	71
PROTect:RMS:TIME?	72
PROTect:RMS:CTR	72
PROTect:RMS:CTR?	73
PROTect:PEAK:CURRent	73
PROTect:PEAK:CURRent?	73
PROTect:PEAK:TIME	74
PROTect:PEAK:TIME?	74
PROTect:PEAK:CTR	74
PROTect:PEAK:CTR?	75
PROTect:RMS:TIME:MAX?	75
PROTect:RMS:TIME:MIN?	75
PROTect:PEAK:TIME:MAX?	76
PROTect:PEAK:TIME:MIN?	76
Chapter11 List Operation Commands.....	77
LIST:STATe	77
LIST:STATe?	77
LIST:REPeat	77
LIST:REPeat?	78
LIST:ENDState.....	78
LIST:ENDState?	78
LIST:JUMPTo.....	79
LIST:JUMPTo?.....	79
LIST:RUN:RECOrd?	79
LIST:RUN:REPEat?	79
LIST:RECOOrder.....	80
LIST:RECOOrder?.....	80
LIST:CONFigure	81
LIST: CONFigure?	81
LIST:NAME	81
LIST:NAME?	82

LIST:RECall	82
LIST:RECOrd:NUM.....	82
LIST:RECOrd:NUM?.....	83
LIST:SAVE	83
LIST:RECAII:NAME?	83
LIST:RECAII:NUM?.....	84
LIST:RUN.....	84
LIST:STOP.....	84
LIST:RUN:STATe?	84
Chapter12 Common Commands	86
*CLS	86
*ESE	86
*ESE?	87
*ESR?	87
*IDN?	88
*OPC.....	88
*OPC?.....	89
*RST	89
*SRE	89
*SRE?	90
*STB?	90
*TRG.....	91
*TST?.....	91
*WAI.....	92

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}

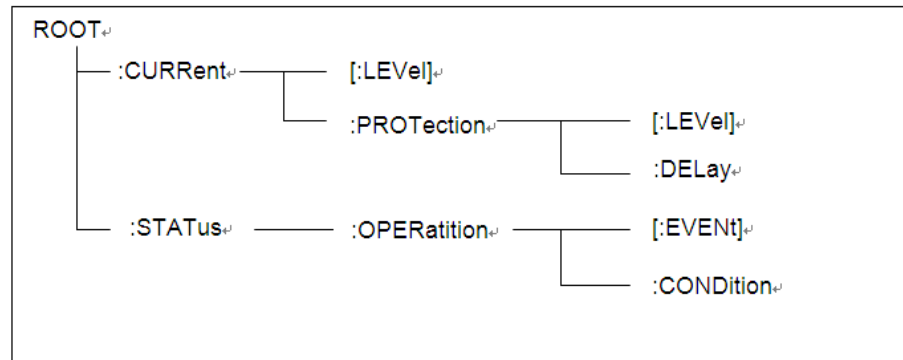
POLARITY {NORMAL|INVERTED}

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 subsystem command tree, from which you access the commands located along the various paths.



Multiple commands in a message

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

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

We consider the head path as a string which will be inserted in front of every command of a message. As for the first command of a message, the head path is a null string; for each subsequent command, the head path is a string which is defined to form the current command until and including the head of the last colon separator. A message with two combined commands: `CURR:LEV 3;PROT:STAT OFF`

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

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

Movement in the subsystem

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

```
PROTection:CLEAr;:STATus:OPERation:CONDition?
```

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

```
POWEr:LEVel 200;PROTection 28; :CURRent:LEVel 3;PROTection:STATe ON
```

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

Including Common Commands

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

```
VOLTage:TRIGgered 17.5;:INITialize;*TRG
OUTPut OFF;*RCL 2;OUTPut ON
```

Case sensitivity

Common commands and SCPI commands are not case sensitive. You can use upper or lower, for example:

```
*RST = *rst
:DATA? = :data?
:SYSTem:PRESet = :system:preset
```

Long-form and short-form versions

A SCPI command word can be sent in its long-form or short-form version. 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:

```
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 Communication Interface

IT7600 have five standard communication interfaces: LAN, USB, RS232, GPIB and CAN. The customer can choose any one according to his demands.

1.6.1 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 you need to press **[Shift]+[Setup]** (Menu) button to enable the settings in menu to be the same with the PC communication configuration. You could do the secondary development with all SCPI commands.


Note

The setup about RS232 in the program should agree with the configuration in the system set. If you want to change the communication parameters, please press [Shift]+[Setup] (Menu) button to enter the system set.

RS-232 Data style

RS232 data is a 10-bit word with one start bit and one stop bit. The number of start bit is not programmable. Stop bit is selectable between 1&2. Besides, you can set the parity bit in the menu using the front panel [Shift]+[Setup] (Menu) key. Parity options are stored in a non-volatile memory.


Baudrate

The front panel [**Shift**]+[**Setup**] (Menu) button allows the user to set baud rate which is stored in the non-volatile memory: 4800/9600/19200/38400/57600/115200.

RS-232 connection

Adopt the RS232 cable with DB-9 interface because the serial port of RS232 can be connected with that of the controller (e.g. PC). The modulating cable of the air-conditioner is not recommended. The pins of plug are shown as the following table.

If your computer is provided with a RS232 interface with DB-25 plug, a cable and a adapter with DB-25 plug (one end) and DB-9 plug (the other end) are required (not the modulating cable of the air-conditioner)

 <p>RS-232 plug pins</p>	Pin	description
	1	connectionless
	2	TXD, transmit data
	3	RXD, receive data
	4	connectionless
	5	GND, ground
	6	connectionless
	7	CTS, clear transfer
	8	RTS, ready to transfer
9	connectionless	

RS-232 Troubleshooting

In case of connection failure of RS232, perform following check:

- Check if the computer and power supply are provided with same Baud rate, parity check bit, data bit and flow control. The power supply shall be configured with one start bit (fixed) and one or two stop bits.
- Just as described in the RS232 connector, appropriate interface cable or adapter shall be adopted. Notes: even if the cable is equipped with right plug, internal wiring may be incorrect.
- The interface cable must be connected to the right serial port (COM1 and COM2) of the computer.

Communication Setup

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

baudrate: 9600(4800, 9600, 19200, 38400, 57600, 115200). You could enter the system menu to set the baudrate.

Data bit: 8

Stop bit: 1

Parity bit: (none, even, odd)

EVEN **8 data bits have even parity**

ODD **8 data bits have odd parity**

NONE **8 data bits have no parity**

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

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

1.6.2 USB Interface

Use a cable with two USB ports to connect power supply and PC. You can program through USB interface to achieve all functions of power supply.

The functions of USB488 interface are as follows:

- Interface is 488.2 USB488 interface
- Receive the following request: REN_CONTROL, GO_TO_LOCAL, and LOCAL_LOCKOUT
- When the interface receives MsgID = TRIGGER USBTMC command, it will transmit the TRIGGER command to the function layer

The functions of power supply's USB488 are as follows:

- receive all SCPI commands
- device is SR1 enabled
- device is RL1 enabled
- device is DT1 enabled

1.6.3 LAN Interface

Press **[Shift]+[Setup]** (Menu) on the front panel together to access the menu. Select LAN in the Communication Configure from System and then configure Gateway, IP and Mask in the LAN option.

Connect the LAN interface of power supply to the computer with a reticle (crossed).

1.6.4 GPIB Interface

First connect GPIB interface to GPIB card of PC with IEEE488 BUS. They must be sufficient contact and tighten the screws. Then set the address within 1-30. Press **[Shift]+[Setup]** (Menu) on the front panel. Then press "Communication Configure" and set "Current Comm" as GPIB. Set the GPIB Address and press **[Enter]**. The power supply operates from a GPIB address set from the front panel. The GPIB address is stored in non-volatile memory.

1.6.5 CAN Communication Port

There is one DB25 interface at the rear panel, and the pin definition is shown below. The user can use this terminal for PC connection; to activate connection, be sure that the values set in the System menu are same as the corresponding values set in PC.



Note

CAN setting in the program shall be consistent with the one set in the System menu of front panel. To query and change, press the composite key **[Shift] + [Setup](Menu)** to enter the setting screen in System menu for query and change..

Baud Rate

Under the Communication Configure menu of the front panel **[Shift] + [Setup](Menu)**, the user can set such parameters as Baudrate Prescaler, Propagation Time Segment and Phase Buffer Segment. Based on calculation conditions and formulas given by the interface, obtain specific Baud Rate value.

The interface provides conditions and formulas:

The Clock of CAN is 42MHz.

$$\text{Baudrate} = 42\text{MHz} / \text{Prescaler} / (\text{BS1} + \text{BS2} + 1)$$

Examples for specific calculation method of baud rate are given below:

Set Prescaler=6, Bit segment1=5 and Bit segment2=8. According to the calculation formula:

$$\text{Baudrate} = 42\text{MHz} / 6 / (5 + 8 + 1) = 500\text{K}, \text{ the Baud Rate is } 500\text{K}.$$

CAN Pin Definition

Use DB25 interface for connection. CAN interface pin is as follows.

CAN interface pin

Pin No.	Description
1	CANH
14	CANL

CAN Troubleshooting:

If CAN connection fails, check that:

1. The PC and power supply have same Baud rate.
2. Appropriate interface pin or adapter is used, as described in CAN connector.
3. The interface cable is correctly connected (CAN_H to CAN_H, CAN_L to CAN_L).

Setting Communication

Before communication operation, match the power supply with the following PC parameters.

Prescale (Pres): Set this parameter as needed

PTS (BS1): Set this parameter as needed

PBS (BS2): Set this parameter as needed

Calculate the Baud Rate according to the above three parameters.

ID (Can ID): Set Can Port ID based on computer.

Chapter2 System Commands

STATus:QUEStionable[:EVENT]?

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

Arguments

None

Returns

<value>

Related Command

STATus:QUEStionable:ENABLE

STATus:QUEStionable:ENABLE

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

Description of the register.:

Bit 0	VO, Overvoltage
Bit 1/2	CO, Overcurrent
Bit 3	Retain 1
Bit 4	OP, Overpower
Bit 5	TO, Over-temperature
Bit 6	TtIO, Over-total power
Bit 7	MO, Memory error
Bit 8	Retain 2

Command syntax

STATus:QUEStionable:ENABLE <value>

Arguments

<value> 0~65535

Example

STATus:QUEStionable:ENABLE 16

Query syntax

STATus:QUEStionable:ENABLE?

Returns

< value >

STATus:QUEStionable:PTRansition

This command set the value of the Questionable PTR(Positive-Transition) register. The register serve as polarity filters between the Questionable Condition and Questionable Event registers to cause the following actions.

When a bit of Questionable PTR register is set to 1, then a 1-to-0 transition of the corresponding bit of the Questionable Condition register causes that bit in the Questionable Event register to be set.

Command syntax

STATus:QUEStionable:PTRansition <value>

Arguments

<value> 0~65535

Example

STATus:QUEStionable:PTRansition 128

Query syntax

STATus:QUEStionable:PTRansition?

Returns

<value>

STATus:QUEStionable:NTRansition

This command set the value of the Questionable NTR(Negative-Transition) register. The register serve as polarity filters between the Questionable Condition and Questionable Event registers to cause the following actions.

When a bit of Questionable NTR register is set to 1, then a 1-to-0 transition of the corresponding bit of the Questionable Condition register causes that bit in the Questionable Event register to be set.

Command syntax

STATus:QUEStionable:NTRansition <value>

Arguments

<value>0~65535

Example

STATus:QUEStionable:NTRansition 128

Query syntax

STATus:QUEStionable:NTRansition?

Returns

<value>

STATus:QUEStionable:CONDition?

This command is used to read the value of query condition register to get the power meter's status, such as ocp/peak/ ocrms/ ov/op/ot

Command syntax

STATus:QUEStionable:CONDition?

Arguments

None

Returns

<value>

STATus:OPERation[:EVENT]?

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

Description of the register.:

Bit 0	CAL, Calibration
Bit 1	ST, self-testing
Bit 2	SETUP mode
Bit 3	LIST mode
Bit 4	STEP mode
Bit 5	Meter mode
Bit 6	Harmonic mode
Bit 7	Scope mode
Bit 8	Vect mode

Command syntax

STATus:OPERation[:EVENT]?

Arguments

None

Returns

<value>

Related Command

STATus:OPERation:ENABLE

STATus:OPERation:ENABLE

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

Command syntax

STATus:OPERation:ENABLE <value>

Arguments

<value> 0~65536

Example

STATus:OPERation:ENABLE 128

Query syntax

STATus:OPERation:ENABLE?

Returns

<value>

STATus:OPERation:PTRansition

This command set the value of the Operation PTR(Positive-Transition) register. Setting parameter can determine which bit value of operation event register is 1 and the bit will cause OPER of status byte register to be 1.

Command syntax

STATus:OPERation:PTRansition <value>

Arguments

<value> 0~65535

Example

STATus:OPERation:PTRansition 128

Query syntax

STATus:OPERation:PTRansition?

Returns

<value>

STATus:OPERation:NTRansition

This command set the value of the Operation NTR(Negative-Transition) register. Setting parameter can determine which bit value of operation event register is 1 and the bit will cause OPER of status byte register to be 1.

Command syntax

STATus:OPERation:NTRansition <value>

Arguments

<value> 0~65535

Example

STATus:OPERation:NTRansition 128

Query syntax

STATus:OPERation:NTRansition?

Returns

<value>

STATus:OPERation:CONDition?

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

Command syntax

STATus:OPERation:CONDition?

Arguments

None

Returns

<value>

SYSTem:BEEPer:IMMediate

This command is used to test the beeper. After executing this command, the instrument will beeper immediately.

Command syntax

SYSTem:BEEPer:IMMediate

Arguments

None

Query syntax

None

SYSTem:BEEPer[:STATe]

This command is used to turn on/turn off the beeper, when the arguments is set to 1/ON, the beeper is enabled and press keyboard will beeper, otherwise, it is mute.

Command syntax

SYSTem:BEEPer < OFF|ON|0|1>

Arguments

<OFF|ON|0|1>

Example

SYSTem:BEEP 1

Query syntax

SYSTem:BEEPer:STATe?

Returns

<0|1>

SYSTem:VERSion?

This command is used to query the current SCPI version. The returned parameter is a string like "YYYY.V", in which the YYYY is the year of that version, V is the software version of that year.

Command syntax

SYSTem:VERSion?

Arguments

None

Returns

<string>

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.

- 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 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”.
- 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.

SYSTem:CLEar

This command is used to clear the error information.

Command syntax

SYSTem:CLEar

Arguments

None

Returns

None

SYSTem:REMOte

This command is used to switch to the remote control mode(PC control).

In this mode, the keys on front panel is invalid except [Loca] key.

Command syntax

SYST:REM

Arguments

None

Query syntax

None

SYSTem:LOCal

This command is used to switch the instrument to local control

mode(control from panel). In this mode ,all of keys on front panel is valid.

Command syntax

SYST:LOC

Arguments

None

Query syntax

None

SYSTem:RWLock

This command is transmit through interface, the same function as SYST:REM,but the difference is: this command can lock “LOCAL” button as well. When this command is executed, the “Local” button can’t switch to local control mode.

Command syntax

SYSTem:RWLock

Arguments

None

Returns

None

SYSTem:DATE

This command is used to set the system date.

Command syntax

SYSTem:DATE < value >,< value >,< value >

Arguments

<value> yy
<value> mm
<value> dd

Example

SYSTem:DATE 2015,5,22

Query syntax

SYSTem:DATE?

Returns

< value >,< value >,< value >

SYSTem:TIME

This command is used to set the system time.

Command syntax

SYSTem:TIME <value>,<value>,<value>

Arguments

<value> hh

<value> mm

<value> ss

Example

SYSTem:TIME 8,45,22

Query syntax

SYSTem:TIME?

Returns

<value>,<value>,<value>

SYSTem:COMMunicate:GPIB[:SELF]:ADDRess

This command is used to set the address of GPIB interface.

Command syntax

SYSTem:COMMunicate:GPIB[:SELF]:ADDRess <value>

Arguments

<value>

Arguments range

1~30

Query syntax

SYSTem:COMMunicate:GPIB[:SELF]:ADDRess?

Returns

<value>

SYSTem:COMMunicate:LAN:CURRent:ADDRess

This command is used to set the address of IP interface.

Command syntax

```
SYSTem:COMMunicate:LAN:CURRent:ADDRess <string>
```

Arguments

<string>

Unit

None

Example

```
SYSTem:COMMunicate:LAN:CURRent:ADDRess "192.168.0.10"
```

Query syntax

```
SYSTem:COMMunicate:LAN:CURRent:ADDRess?
```

Returns

<string>

SYSTem:COMMunicate:LAN:CURRent:DGATeway

This command is used to set gateway of power supply.

Command syntax

```
SYSTem:COMMunicate:LAN:CURRent:DGATeway <string>
```

Arguments

<string>

Unit

None

Query syntax

```
SYSTem:COMMunicate:LAN:CURRent:DGATeway?
```

Returns

<string>

SYSTem:COMMunicate:LAN:CURRent:SMASK

This command is used to set the subnet mask of power supply.

Command syntax

SYSTem:COMMunicate:LAN:CURRent:SMASk <string>

Arguments

<string>

Unit

None

Query syntax

SYSTem:COMMunicate:LAN:CURRent:SMASk?

Returns

<string>

SYSTem:COMMunicate:LAN:DHCP[:STATe]

This command is used to set Dynamic IP address of power supply.

Command syntax

SYSTem:COMMunicate:LAN:DHCP[:STATe] <0|1|OFF|ON>

Arguments

<0|1|OFF|ON>

Example

SYST:COMM:LAN:DHCP 1

Query syntax

SYSTem:COMMunicate:LAN:DHCP[:STATe]?

Returns

<0|1>

SYSTem:COMMunicate:LAN:SOCKetport

This command is used to set socket port of communication.

Command syntax

SYSTem:COMMunicate:LAN:SOCKetport <value>

Arguments

<value>

Unit

None

Query syntax

SYSTem:COMMunicate:LAN:SOCKetport?

Returns

<value>

SYSTem:COMMunicate:LAN:MACaddress?

This command is used to return the MAC address of LAN communication.

Command syntax

SYSTem:COMMunicate:LAN:MACaddress?

Arguments

None

Unit

None

Query syntax

SYSTem:COMMunicate:LAN:MACaddress?

Returns

<string>

SYSTem:COMMunicate:SElect

This command is used to select the communication interface.

Command syntax

SYSTem:COMMunicate:SElect <RS232|USB|GPIB|LAN>

Arguments

<RS232|USB|GPIB|LAN>

Example

SYST:COMM:SEL RS232

Query syntax

SYSTem:COMMunicate:SElect?

Returns

<RS232|USB|GPIB|LAN>

SYSTem:COMMunicate:SERial:BAUDrate

This command is used to set the baudrate of RS232.

Command syntax

SYSTem:COMMunicate:SERial:BAUDrate

Arguments

4800|9600|19200|38400|57600|115200

Query syntax

SYSTem:COMMunicate:SERial:BAUDrate?

Returns

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

SYSTem:BACKlight:BRIGhtness

This command is used to set the brightness of backlight.

Command syntax

SYSTem:BACKlight:BRIGhtness <value>

Arguments

<value>

Arguments range

1-9

Query syntax

SYSTem:BACKlight:BRIGhtness?

Returns

<value>

SYSTem:POSetup

This command is used to set the system power-on parameter.

Command syntax

SYSTem:POSetup RST|SAV0

Arguments

RST|SAV0

Reset Value

None

Query syntax

SYSTem:POSetup?

SYSTem:DEVice:NUM

This command is used to set the number of device.

Command syntax

SYSTem:DEVice:NUM <value>

Arguments

<value>

Reset Value

None

Query syntax

SYSTem:DEVice:NUM?

Returns

<value>

SYSTem:MEASure:MODE

This command is used to set the measure mode.

Command syntax

SYSTem:MEASure:MODE

Arguments

<METer|SCOPe|HARMonic|VECTor>

Query syntax

SYSTem:MEASure:MODE?

Returns

<METer|SCOPe|HARMonic|VECTor>

SYSTem:SOURce:MODE

This command is used to set the source mode.

Command syntax

SYSTem:SOURce:MODE

Arguments

<SETUp/STEP/LIST>

Query syntax

SYSTem:SOURce:MODE?

Returns

<SETUp/STEP/LIST>

SYSTem:TRIG:SOURce

This command is used to set the trigger source type of system.

Command syntax

SYSTem:TRIG:SOURce

Arguments

<Key|SOFT|BUS|EXTernal>

Query syntax

SYSTem:TRIG:SOURce?

Returns

<Key|SOFT|BUS|EXTernal>

SYSTem:RUN:NETWork?

This command is used to query the current network status.

Command syntax

SYSTem:RUN:NETWork?

Arguments

None

Query syntax

SYSTem:RUN:NETWork?

Returns

<ON|OFF>

SYSTem:RUN:DEVNum?

This command is used to query the number of current system devices.

Command syntax

SYSTem:RUN:DEVNum?

Arguments

None

Query syntax

SYSTem:RUN:DEVNum?

Returns

<value>

SYSTem:RUN:GRPNum?

This command is used to query the number of current system groups.

Command syntax

SYSTem:RUN:GRPNum?

Arguments

None

Query syntax

SYSTem:RUN:GRPNum?

Returns

<value>

Chapter3 Measurement Commands

FETCh[:SCALar]:VOLTage?

This command is used to read the A voltages.

Command syntax

```
FETCh[:SCALar]:VOLTage? <A>
```

Arguments

<A>

Returns

<float>

FETCh[:SCALar]:CURRent?

This command is used to read the A currents.

Command syntax

```
FETCh[:SCALar]:CURRent? <A>
```

Arguments

<A>

Returns

<float>

FETCh[:SCALar]:POWer[:REAL]?

This command is used to read the A powers.

Command syntax

```
FETCh[:SCALar]:POWer [:REAL]? <A>
```

Arguments

<A>

Returns

<float>

FETCh[:SCALar]:POWer:APParent?

This command is used to read the A apparent powers.

Command syntax

FETCh[:SCALar]:POWer:APParent? <A>

Arguments

<A>

Returns

<float>

FETCh[:SCALar]:POWer:PFACTOR?

This command is used to read the A power factors.

Command syntax

FETCh[:SCALar]:POWer:PFACTOR? <A>

Arguments

<A>

Returns

<float>

FETCh[:SCALar]:FREQuency?

This command is used to read the A frequencys.

Command syntax

FETCh[:SCALar]:FREQuency? <A>

Arguments

<A>

Returns

<float>

FETCh[:SCALar]:CFACTOR?

This command is used to read the A peak factors.

Command syntax

FETCh[:SCALar]:CFACTOR? <A>

Arguments

<A>

Returns

<float>

FETCh[:SCALar]:CURRent:PEAK:PLUS?

This command is used to read the A current readings at maximum positive peak.

Command syntax

FETCh[:SCALar]:CURRent:PEAK:PLUS? <A>

Arguments

<A>

Returns

<float>

FETCh[:SCALar]:CURRent:PEAK:MINUs?

This command is used to read the A current readings at maximum negative peak.

Command syntax

FETCh[:SCALar]:CURRent:PEAK:MINUs? <A>

Arguments

<A>

Returns

<float>

FETCh[:SCALar]:CURRent:ISURge?

This command is used to read the A surge currents.

Command syntax

FETCh[:SCALar]:CURRent:ISURge? <A>

Arguments

<A>

Returns

<float>

MEASure[:SCALar]:VOLTage?

This command is used to read the A real-time voltages.

Command syntax

```
MEASure [:SCALar]:VOLTage? <A>
```

Arguments

<A>

Returns

<float>

MEASure[:SCALar]:CURRent?

This command is used to read the A real-time currents.

Command syntax

```
MEASure[:SCALar]:CURRent? <A>
```

Arguments

<A>

Returns

<float>

MEASure[:SCALar]:POWer[:REAL]?

This command is used to read the A real-time powers.

Command syntax

```
MEASure[:SCALar]:POWer[:REAL]? <A>
```

Arguments

<A>

Returns

<float>

MEASure[:SCALar]:POWer:APParent?

This command is used to read the A real-time apparent powers.

Command syntax

```
MEASure[:SCALar]:POWer:APParent? <A>
```

Arguments

<A>

Returns

<float>

MEASure[:SCALar]:POWER:PFACtor?

This command is used to read the A real-time power factors.

Command syntax

MEASure[:SCALar]:POWER:PFACtor? <A>

Arguments

<A>

Returns

< float >

MEASure[:SCALar]:FREQuency?

This command is used to read the A real-time frequencies.

Command syntax

MEASure[:SCALar]:FREQuency? <A>

Arguments

<A>

Returns

<float>

MEASure[:SCALar]:CFACtor?

This command is used to read the A real-time peak factors.

Command syntax

MEASure[:SCALar]:CFACtor? <A>

Arguments

<A>

Returns

< float >

MEASure[:SCALar]:CURRent:PEAK:PLUS?

This command is used to read the A real-time current readings at maximum positive peak.

Command syntax

```
MEASure[:SCALar]:CURRent:PEAK:PLUS? <A>
```

Arguments

<A>

Returns

< float >

MEASure[:SCALar]:CURRent:PEAK:MINUs?

This command is used to real-time measure the A maximum negative peak current.

Command syntax

```
FETCh[:SCALar]:CURRent:PEAK:MINUs? <A>
```

Arguments

<A>

Returns

< float >

MEASure[:SCALar]:CURRent:ISURge?

This command is used to read the A real-time surge currents.

Command syntax

```
MEASure[:SCALar]:CURRent:ISURge? <A>
```

Arguments

<A>

Returns

<float>

FETCh?

This command is used to obtain all METER data of an item.

Command syntax

```
FETCh? <A>
```

Arguments

<A>

Returns

<float>, ..., <float> (Measured values of 16 floats)

MEASure?

This command is used to obtain all real-time METER data of an item.

Command syntax

MEASure? <A>

Arguments

<A>

Returns

<float>, ..., <float> (Measured values of 16 floats)

Chapter4 Vector Commands

Under the three-phase mode, use vector commands. However, IT7626 programmable AC power supply has no vector commands.

VECTor:ORDER

This command is used to set the vector order.

Command syntax

VECTor:ORDER <value>

Arguments

<value> order.

Query syntax

VECTor:ORDER?

Returns

<value>

VECTor:DATA?

This command is used to obtain the vector data to which the current order corresponds.

A: rms

A: phase

B:rms

B:phase

C:rms

C:phase

Command syntax

VECTor:DATA?

Arguments

None

Returns

<float>,...,<float> 6 floats

VECTor:TYPE

This command is used to set the vector type.

Command syntax

VECTor:TYPE <U||ALL>

Arguments

<U||ALL>

Query syntax

VECTor:TYPE?

Returns

<U||ALL>

Chapter5 Harmonic Commands

In commands of harmonics measurement, the range of values for parameter “value” is zero to fifty. 0 corresponds to DC, 1 corresponds to fundamental harmonic and 2-50 corresponds to harmonics. The instrument interface should be under the harmonics function while using the harmonics command to measure.

MEASure[:SCALar]:HARMonics?

This command is used to read the real-time measured harmonic of a phase.

Command syntax

```
MEASure[:SCALar]:HARMonics? <A|B|C>,<U||ALL>
```

Arguments

```
<A|B|C>  
<U||ALL>
```

Returns

```
<float>
```

MEASure[:SCALar]:VOLTage:HARMonics[:AMPLitude]?

This command is used to read the real-time measured voltage amplitude of a harmonic.

Command syntax

```
MEASure[:SCALar]:VOLTage:HARMonics[:AMPLitude]?  
<A|B|C>,<FUNDamental|TOTal| ALL|value>
```

Arguments

```
<A|B|C>,  
<FUNDamental|TOTal|ALL|value>
```

Returns

```
<float>
```

MEASure[:SCALar]:VOLTage:HARMonics:DISort?

This command is used to read the real-time measured distortion of a harmonic.

Command syntax

MEASure[:SCALar]:VOLTage:HARMonics:DISort? <A|B|C>,<TOTal|value>

Arguments

<A|B|C>,
<TOTal|ALL|value>

Returns

<float>

MEASure[:SCALar]:PHASe:HARMonics?

This command is used to read the real-time measured phase of a harmonic.

Command syntax

MEASure[:SCALar]:PHASe:HARMonics?<A|B|C>,<ALL|value>,<U|I>

Arguments

<A|B|C>,
<ALL|value>,
<U|I>

Returns

<float>

MEASure[:SCALar]:FREQuency:HARMonics:FUNDamental?

This command is used to read the real-time measured frequency of a harmonic.

Command syntax

MEASure[:SCALar]:FREQuency:HARMonics:FUNDamental? <A|B|C>,<U|I>

Arguments

<A|B|C>,
<U|I>

Returns

<float>

FETCH[:SCALar]:HARMonics?

This command is used to read the harmonic of a phase.

Command syntax

FETCh[:SCALar]:HARMonics? <A|B|C>,<U||ALL>

Arguments

A|B|C,
U||ALL

Returns

<float>

FETCh[:SCALar]:VOLTage:HARMonics[:AMPLitude]?

This command is used to read the voltage amplitude of a harmonic.

Command syntax

FETCh[:SCALar]:VOLTage:HARMonics[:AMPLitude]?<A|B|C>,FUNDamental|TOTal|ALL| order

Arguments

<A|B|C>,
FUNDamental|TOTal|ALL| order

Returns

<float>

FETCh[:SCALar]:VOLTage:HARMonics:DISTort?

This command is used to read the measured distortion of a harmonic.

Command syntax

FETCh[:SCALar]:VOLTage:HARMonics:DISTort?<A|B|C>,TOTal|ALL| order

Arguments

<A|B|C>,
TOTal|ALL| order

Returns

<float>

FETCh[:SCALar]:PHASe:HARMonics?

This command is used to read the phase of a harmonic.

Command syntax

FETCh[:SCALar]:PHASe:HARMonics?<A|B|C>,ALL| order,U||

Arguments

<A|B|C>,
ALL| order,U|I

Returns

<float>

**FETCh[:SCALAr]:FREQuency:HARMOnics:FUNDament
al?**

This command is used to read the frequency of a harmonic.

Command syntax

FETCh[:SCALAr]:FREQuency:HARMOnics:FUNDamental?A|B|C,U|I

Arguments

A|B|C,
U|I

Returns

<float>

Chapter6 Oscilloscope Commands

WAVE:TRIGger:MODE

This command is used to set the trigger mode when take waveform as trigger source.

Command syntax

WAVE:TRIGger:MODE <AUTO|NORMAl|SINGle>

Arguments

AUTO|NORMAl|SINGle

Query syntax

WAVE:TRIGger:MODE?

Returns

AUTO|NORMAl

WAVE:TRIGger:MODE?

This command is used to query the trigger mode when take waveform as trigger source.

Command syntax

WAVE:TRIGger:MODE?

Arguments

None

WAVE:TRIGger:SLOPe

This command is used to set trigger slope when takes waveform as trigger source.

Command syntax

WAVE:TRIGger:SLOPe <POSitive|NEGative|ANY>

Arguments

POSitive|NEGative|ANY

Query syntax

WAVE:TRIGger:SLOPe?

Returns

POS|NEG|ANY

WAVE:TRIGger:SLOPe?

This command is used to query trigger slope when takes waveform as trigger source.

Command syntax

WAVE:TRIGger:SLOPe?

Arguments

None

WAVE:TRIGger:SOURce

This command can set or read the device taking which kind waveform as trigger source.

Command syntax

WAVE:TRIGger:SOURce <VOLTage|CURRent>

Arguments

VOLTage|CURRent

Query syntax

WAVE:TRIGger:SOURce?

Returns

VOLT|CURR

WAVE:TRIGger:SOURce?

This command can query the device taking which kind waveform as trigger source.

Command syntax

WAVE:TRIGger:SOURce?

Arguments

None

WAVE:TRIGger:VOLTage[:LEVel]

This command is used to set the voltage trigger level.

Command syntax

WAVE:TRIGger:VOLTage[:LEVel]

Arguments

<value>

Arguments range

Refer to the specifications.

Unit

V

Query syntax

WAVE:TRIGger:VOLTage[:LEVel]?

Returns

<value>

WAVE:TRIGger:VOLTage[:LEVel]?

This command is used to query the voltage trigger level.

Command syntax

WAVE:TRIGger:VOLTage[:LEVel]?

Arguments

None

WAVE:TRIGger:CURRent[:LEVel]

This command is used to set the current trigger level

Command syntax

WAVE:TRIGger:CURRent[:LEVel]

Arguments

<value>

Arguments range

Refer to the specifications.

Unit

A

Query syntax

WAVE:TRIGger:CURRent[:LEVel]?

Returns

<value>

WAVE:TRIGger:CURRent[:LEVel]?

This command is used to query the current trigger level

Command syntax

WAVE:TRIGger:CURRent[:LEVel]?

Arguments

None

WAVE:TRIGger:DELay:TIME

This command is used to set the trigger delay time when take a waveform as trigger source.(Note:the delay time setting should be less than the time width of the whole screen).

Command syntax

WAVE:TRIGger:DELay:TIME <float>

Arguments

<float>

Unit

s

Query syntax

WAVE:TRIGger:DELay:TIME?

Returns

<float>

WAVE:TRIGger:DELay:TIME?

This command is used to query the trigger delay time when take a waveform as trigger source.(Note:the delay time setting should be less than the time width of the whole screen).

Command syntax

WAVE:TRIGger:DElay:TIME?

Arguments

None

WAVE:VOLTage:DATA[:NORMAlization]?

This command is used to obtain the voltage datas after normalization.

Command syntax

WAVE:VOLTage:DATA[:NORMAlization]?

Arguments

None

Returns

<float>

WAVE:CURRent:DATA[:NORMAlization]?

This command is used to obtain the current datas after normalization.

Command syntax

WAVE:CURRent:DATA[:NORMAlization]?

Arguments

None

Returns

<float>

WAVE:AUTO

This command is used to set the auto mode to start waveform capture.

Command syntax

WAVE:AUTO

Arguments

None

Query syntax

None

WAVE:SINGLe

This command is used to set the single mode to start waveform capture.

Command syntax

WAVE:SINGLe

Arguments

None

Query syntax

None

WAVE:RUN

This command is used to start waveform capture.

Command syntax

WAVE:RUN

Arguments

None

Query syntax

None

WAVE:STOP

This command can stop the waveform capture.

Command syntax

WAVE:STOP

Arguments

None

Query syntax

None

WAVE:TRIGger[:STATe]?

This command is used to query the trigger status.

Command syntax

WAVE:TRIGger[:STATe]?

Returns

Auto| Auto?|Trig|Trig?|Stop

WAVE:SCOPE:DIVTime

This command is used to set the resolution time interval.

Command syntax

WAVE:SCOPE:DIVTime

Arguments

0.0005|0.001|0.002|0.005|0.01|0.02|0.05|0.1|0.2

Unit

s

Query syntax

WAVE:SCOPE:DIVTime?

Returns

<float>

WAVE:SCOPE:DIVTime?

This command is used to query the resolution time interval.

Command syntax

WAVE:SCOPE:DIVTime?

Arguments

None

Unit

s

WAVE:SCOPE:SELection

This command is used to set waveform display options.

Command syntax

WAVE:SCOPE:SELection

Arguments

U|A|UA

Query syntax

WAVE:SCOPE:SELECTION?

Returns

U|A|UA

WAVE:SCOPE:SELECTION?

This command is used to query waveform display options.

Command syntax

WAVE:SCOPE:SELECTION?

Arguments

U|A|UA

WAVE:KNOB:SELECTION

This command is used to set the knob options.

Command syntax

WAVE:KNOB:SELECTION

Arguments

UR|AR|UB|AB|TL|TD|T/d

Query syntax

WAVE:KNOB:SELECTION?

Returns

U|A|TL|TD|T/d

WAVE:KNOB:SELECTION?

This command is used to query the knob options.

Command syntax

WAVE:KNOB:SELECTION?

Arguments

None

Returns

U|A|TL|TD|T/d

WAVE:VOLTage:BASE

This command is used to set the voltage base.

Command syntax

WAVE:VOLTage:BASE

Arguments

<float>

Unit

V

Query syntax

WAVE:VOLTage:BASE?

Returns

<float>

WAVE:VOLTage:BASE?

This command is used to query the voltage base.

Command syntax

WAVE:VOLTage:BASE?

Arguments

None

Returns

<float>

WAVE:CURREnt:BASE

This command is used to set the current base.

Command syntax

WAVE:CURREnt:BASE <float>

Arguments

<float>

Unit

A

Query syntax

WAVE:CURRent:BASE?

Returns

<float>

WAVE:CURRent:BASE?

This command is used to query the current base.

Command syntax

WAVE:CURRent:BASE <float>?

Arguments

None

Returns

<float>

WAVE:VOLTage:RANGe

This command is used to set the voltage range.

Command syntax

WAVE:VOLTage:RANGe

Arguments

<float>

Arguments range

1V-100V

Unit

V

Query syntax

WAVE:VOLTage:RANGe?

Returns

<float>

WAVE:VOLTage:RANGe?

This command is used to query the voltage range.

Command syntax

WAVE:VOLTage:RANGe?

Arguments

None

Returns

<float>

WAVE:CURREnt:RANGe

This command is used to set the current range.

Command syntax

WAVE:CURREnt:RANGe

Arguments

<float>

Arguments range

0.005A-100A

Unit

A

Query syntax

WAVE:CURREnt:RANGe?

Returns

<float>

WAVE:CURREnt:RANGe?

This command is used to query the current range.

Command syntax

WAVE:CURREnt:RANGe?

Arguments

None

Returns

<float>

WAVE:ADDR

This command is used to set the oscilloscope address.

Command syntax

WAVE:ADDR <value>

Arguments

<value> A|B|C

Query syntax

WAVE:ADDR?

Returns

<value>

WAVE:ADDR?

This command is used to query the oscilloscope address.

Command syntax

WAVE:ADDR?

Arguments

None

Returns

<value>

Chapter7 Self-defined wave Commands

SELFdefine:NAME

This command is used to set the current self-defined wave name.

Command syntax

SELFdefine:NAME

Arguments

<string>

Query syntax

SELFdefine:NAME?

Returns

<string>

SELFdefine:RECALL:NAME?

This command is used to query the saved self-defined wave file name.

Command syntax

SELFdefine:RECALL:NAME? <value>

Arguments

<value> self-defined wave file index (digital)

Returns

<string> file name

SELFdefine:SAVE

This command is used to save the current self-defined wave file.

Command syntax

SELFdefine:SAVE

Arguments

None

SELFdefine:DATA

This command is used to set self-defined wave data.

Command syntax

SELFdefine:DATA <value>,<float>

Arguments

<value> Index (0-1023)

<float> value (-1.0-1.0)

Query syntax

SELFdefine:DATA? <value>

Returns

<float>

SELFdefine:RECALL

This command is used to import the self-defined file.

Command syntax

SELFdefine:RECALL <string>

Arguments

<string> file name

Returns

None

SELFdefine:RECALL:NUM?

This command is used to query the valid number of self-defined files.

Command syntax

SELFdefine:RECALL:NUM?

Arguments

None

Returns

<value>

Chapter8 Output Commands

[SOURce:]OUTPut[:STATe]

This command is used to control the output switch state.

Command syntax

```
[SOURce:]OUTPut[:STATe] <A|ALL>,<0|1|OFF|ON>
```

Arguments

<A|ALL> phase

<0|1|OFF|ON> Output state

Example

```
OUTP A,1
```

Query syntax

```
[SOURce:]OUTPut[:STATe]? < A|ALL >
```

Returns

```
<0|1|OFF|ON>
```

[SOURce:]OUTPut[:STATe]?

This command is used to query the output switch state.

Command syntax

```
[SOURce:]OUTPut[:STATe]?<A|ALL>
```

Arguments

<A|B|C|ALL>

Example

```
OUTP? A
```

Returns

```
0|1|OFF|ON
```

[SOURce:]RELAY:MODE

This command is used to set the relay mode of a phase (Normally Open, Normally Close).

Command syntax

[SOURce:]RELAY:MODE <A>,<CLOSE|OPEN>

Arguments

<A>

<CLOSE|OPEN>

Query syntax

[SOURce:]RELAY:MODE? < A|B|C|ALL >

Returns

<CLOSE|OPEN>

[SOURce:]RELAY:MODE?

This command is used to query the relay mode of a phase (Normally Open, Normally Close).

Command syntax

[SOURce:]RELAY:MODE? <A/B/C/ALL>

Arguments

<A/B/C/ALL>

Returns

<CLOSE|OPEN>

[SOURce:]RANGe

This command is used to set the voltage and current range.

Command syntax

[SOURce:]RANGe <A>,<AUTO|HIGH>

Arguments

<A>

<AUTO|HIGH>

Query syntax

[SOURce:]RANGe? <A/B/C/ALL>

[SOURce:]RANGe?

This command is used to query the voltage and current range.

Command syntax

[SOURce:]RANGe? <A/B/C/ALL>

Arguments

<A/B/C/ALL>

Returns

<AUTO|HIGH>

[SOURce:]PROTection:CLEar

This command is used to clear the latch. This latch disables input in case of protection, e.g., OV or OC protection. All error conditions must be removed before the latch is cleared. Input and then system recovers to the status before any error happens.

Command syntax

[SOURce:]PROTection:CLEar < A/B/C/ALL >,< OT|OV|OC|OP|ALL >

Arguments

<A/B/C/ALL>,
< OT|OV|OC|OP|ALL>

Example

PROTection:CLEar A,ALL

Returns

None

[SOURce:]PROTection?

This command is used to query the mark for obtaining protection status.

Command syntax

[SOURce:]PROTection? < A>

Arguments

< A>

Returns

<value> Current protection value of device

[SOURce:]LOOPspeed

This command is used to set the loop speed of a phase.

Command syntax

[SOURce:]LOOPspeed < A>,<LOW|HIGH>

Arguments

< A>

<LOW|HIGH>

Loop speed of phase (Low speed,High speed)

Query syntax

[SOURce:]LOOPspeed? < A/B/C/ALL >

Returns

None

[SOURce:]LOOPspeed?

This command is used to query the loop speed of a phase.

Command syntax

[SOURce:]LOOPspeed? < A/B/C/ALL >

Arguments

< A/B/C/ALL >

Returns

<LOW|HIGH>

[SOURce:]DEVICE:TYPE

This command is used to set the device type,single/paralle/threephase

Command syntax

[SOURce:]DEVICE:TYPE <SINGLE|PARAllel|THREephase>

Arguments

<SINGLE|PARAllel|THREephase>

Query syntax

[SOURce:]DEVICE:TYPE?

Returns

<SINGLE|PARAllel|THREephase>

[SOURce:]DEVICE:TYPE?

This command is used to query device type.

Command syntax

[SOURce:]DEVice:TYPE?

Arguments

None

Returns

<SINGle|PARAllel|THREephase>

[SOURce:]NORMAl:VOLTage:AC[:LEVEl][:IMMediate][:AMPLitude]

This command is used to set the AC voltage of a phase.

Command syntax

[SOURce:]NORMAl:VOLTage:AC[:LEVEl][:IMMediate][:AMPLitude]
<A>,<float>

Arguments

<A>
<float> AC Voltage value

Query syntax

[SOURce:]NORMAl:VOLTage:AC[:LEVEl][:IMMediate][:AMPLitude]?
<A|B|C|ALL>

Returns

<float>

[SOURce:]NORMAl:VOLTage:AC:LIMit[:LEVEl]

This command is used to set the AC maximum and minimum voltages.

Command syntax

[SOURce:]NORMAl:VOLTage:AC:LIMit[:LEVEl]
<A>,<MINimum|MAXimum>,<float>

Arguments

<A>
<MINimum|MAXimum>
<float>

Query syntax

[SOURce:]NORMAl:VOLTage:AC:LIMit[:LEVEl]?

<A|B|C|ALL>,<MINimum|MAXimum>

Returns

<float>

[SOURce:]NORMal:VOLTage:DC[:LEVel][:IMMediate]

This command is used to set the DC voltage of a phase.

Command syntax

[SOURce:]NORMal:VOLTage:DC[:LEVel][:IMMediate]
<A>,<float>

Arguments

<A>
<float> DC Voltage value

Query syntax

[SOURce:]NORMal:VOLTage:DC[:LEVel][:IMMediate]? <A|B|C|ALL>

Query syntax

<float>

[SOURce:]NORMal:VOLTage:DC:LIMit[:LEVel]

This command is used to set the DC maximum and minimum voltage.

Command syntax

[SOURce:]NORMal:VOLTage:DC:LIMit[:LEVel]
<A|B|C|ALL>,<MINimum|MAXimum>,<float>

Arguments

<A|B|C|ALL>
<MINimum|MAXimum>
<float>

Query syntax

[SOURce:]NORMal:VOLTage:DC:LIMit[:LEVel]?
<A|B|C|ALL>,<MINimum|MAXimum>

Returns

<float>

[SOURce:]NORMal:FREQuency[:LEVel][:IMMediate]

This command is used to set the frequency of a phase.

Command syntax

```
[SOURce:]NORMal:FREQuency[:LEVel][:IMMediate]
<A>,<float>
```

Arguments

```
<A>
<float> frequency
```

Query syntax

```
[SOURce:]NORMal:FREQuency[:LEVel][:IMMediate]? <A>
```

Returns

```
<float>
```

[SOURce:]NORMal:FREQuency:LIMit[:LEVel]

This command is used to set the maximum and minimum frequency.

Command syntax

```
[SOURce:]NORMal:FREQuency:LIMit[:LEVel]
<A>,<MINimum|MAXimum>,<float>
```

Arguments

```
<A>
<MINimum|MAXimum>
<float>
```

Query syntax

```
[SOURce:]NORMal:FREQuency:LIMit[:LEVel]?
<A|B|C|ALL>,<MINimum|MAXimum>
```

Returns

```
<float>
```

[SOURce:]NORMal:PHASe:STARt[:LEVel][:IMMediate]

This command is used to set the starting angle of a phase.

Command syntax

```
[SOURce:]NORMal:PHASe:STARt[:LEVel][:IMMediate]
<A>,<float>
```


Arguments

<A>
<float> the starting angle

Query syntax

[SOURce:]NORMAl:PHASe:STARt[:LEVeI][:IMMediate]? <A>

Returns

<float>

[SOURce:]NORMAl:PHASe:STARt:LIMit[:LEVeI]

This command is used to set the maximum and minimum starting angle of a phase.

Command syntax

[SOURce:]NORMAl:PHASe:STARt:LIMit[:LEVeI]
<A>,<MINimum|MAXimum>,<float>

Arguments

<A>
<MINimum|MAXimum>
<float>

Query syntax

[SOURce:]NORMAl:PHASe:STARt:LIMit[:LEVeI]?
<A>,< MINimum|MAXimum >

Returns

<float>

[SOURce:]NORMAl:PHASe:STOP[:LEVeI][:IMMediate]

This command is used to set the stopping angle of a phase.

Command syntax

[SOURce:]NORMAl:PHASe:STOP[:LEVeI][:IMMediate]
<A>,<float>

Arguments

<A>
<float> Stopping angle (float value)

Query syntax

[SOURce:]NORMal:PHASe:STOP[:LEVel][:IMMEDIATE]? <A>

Returns

<float>

[SOURce:]NORMal:PHASe:STOP:LIMit[:LEVel]

This command is used to set the maximum and minimum stopping angle of a phase.

Command syntax

[SOURce:]NORMal:PHASe:STOP:LIMit[:LEVel]
<A>,<MINimum|MAXimum>,<float>

Arguments

<A>
<MINimum|MAXimum>
<float>

Query syntax

[SOURce:]NORMal:PHASe:STOP:LIMit[:LEVel]?
<A>,<MINimum|MAXimum>

Returns

<float>

[SOURce:]NORMal:MODE

This command is used to set the working mode of power supply.

Command syntax

[SOURce:]NORMal:MODE <A>,<AC|DC|ACDC>

Arguments

<A>
<AC|DC|ACDC> working mode

Query syntax

[SOURce:]NORMal:MODE? <A>

Returns

<AC|DC|ACDC>

[SOURce:]NORMal:WAVE

This command is used to set the wave of a phase.

Command syntax

```
[SOURce:]NORMal:WAVE <A>,<value>
```

Arguments

<A>

<value> wave index (0-sine,1-square,2-Sawtooth,3-Triangle,4-THDWave)

Query syntax

```
[SOURce:]NORMal: WAVE? < A>
```

Returns

<value>

[SOURce:]NORMal:THD:ORDer:DATA

This command is used to set the THD order data.

Command syntax

```
[SOURce:]NORMal: THD:ORDer:DATA <A>,<string>
```

Arguments

<A>

<string> String ("order number, thd float value, angle value")

Example

Example: NORMal:THD:ORDer:DATA A,"5,0.1,350" Set the thd of 5th harmonic to 0.1, and angle to 350 degrees.

Query syntax

```
[SOURce:]NORMal: THD:ORDer:DATA? <A>,<value>
```

Returns

<string>

[SOURce:]NORMal:THD:DOWNload

This command is used to download the THD configuration.

Command syntax

```
[SOURce:]NORMal:THD:DOWNload <A|B|C|ALL>
```

Arguments

<A|B|C|ALL>

Query syntax

None

Returns

None

Chapter9 STEP Commands

STEP:VOLTage:STARt

This command is used to set the starting voltage value.

Command syntax

STEP:VOLTage:STARt <float>

Arguments

<float> float number of the starting voltage value

Query syntax

STEP:VOLTage:STARt?

Returns

<float>

STEP:VOLTage:STARt?

This command is used to query the starting voltage value.

Command syntax

STEP:VOLTage:STARt?

Arguments

None

Returns

<float>

STEP:VOLTage:STOP

This command is used to set the end voltage value.

Command syntax

STEP:VOLTage:STOP <float>

Arguments

<float> float number of the end voltage value

Query syntax

STEP:VOLTage:STOP?

Returns

<float>

STEP:VOLTage:STOP?

This command is used to query the end voltage value.

Command syntax

STEP:VOLTage:STOP?

Arguments

None

Returns

< float>

STEP:VOLTage:INTERval

This command is used to set the voltage step interval.

Command syntax

STEP:VOLTage:INTERval <float>

Arguments

<float> float number of voltage step

Query syntax

STEP:VOLTage:INTERval?

Returns

<float>

STEP:VOLTage:INTERval?

This command is used to query the voltage step interval.

Command syntax

STEP:VOLTage:INTERval?

Arguments

None

Returns

<float>

STEP:MODE

This command is used to set the STEP working mode.

Command syntax

```
STEP:MODE <TIME|TRIG >
```

Arguments

<TIME|TRIG >

TIME: Time sweep mode

TRIG: Trigger sweep mode

Query syntax

```
STEP:MODE?
```

Returns

<TIME|TRIG >

STEP:MODE?

This command is used to query the STEP working mode.

Command syntax

```
STEP:MODE?
```

Arguments

None

Returns

<TIME|TRIG >

STEP:FREQ:START

This command is used to set the starting frequency value.

Command syntax

```
STEP:FREQ:START <float>
```

Arguments

<float> float number of the starting frequency value

Query syntax

```
STEP:FREQ:START?
```

Returns

<float>

STEP:FREQ:START?

This command is used to query the starting frequency value.

Command syntax

STEP:FREQ:START?

Arguments

None

Returns

<float>

STEP:FREQ:STOP

This command is used to set the end frequency value.

Command syntax

STEP:FREQ:STOP <float>

Arguments

<float> float number of the end frequency value

Query syntax

STEP: FREQ:STOP?

Returns

<float>

STEP:FREQ:STOP?

This command is used to query the end frequency value.

Command syntax

STEP:FREQ:STOP?

Arguments

None

Returns

<float>

STEP:FREQ:INTERval

This command is used to set the STEP frequency step interval.

Command syntax

STEP:FREQ:INTERval <float>

Arguments

<float> float number of frequency step

Query syntax

STEP:FREQ:INTERval?

Returns

<float>

STEP:FREQ:INTERval?

This command is used to query the STEP frequency step interval.

Command syntax

STEP:FREQ:INTERval?

Arguments

None

Returns

<float>

STEP:TIME:INTERval

This command is used to set the STEP time step interval.

Command syntax

STEP:TIME:INTERval <float>

Arguments

<float> float number of time step

Query syntax

STEP:TIME:INTERval?

Returns

<float>

STEP:TIME:INTERval?

This command is used to query the STEP time step interval.

Command syntax

STEP:TIME:INTERval?

Arguments

None

Returns

<float>

STEP:PRiority<VOLT|FREQ>

This command is used to set STEP sweep priority (voltage priority or frequency priority).

Command syntax

STEP:PRiority <VOLT|FREQ>

Arguments

<VOLT|FREQ>

VOLT: voltage priority

FREQ: frequency priority

Query syntax

STEP:PRiority? <VOLT|FREQ>

Returns

<VOLT|FREQ>

STEP:PRiority?

This command is used to query the STEP sweep priority (voltage priority or frequency priority).

Command syntax

STEP:PRiority?

Arguments

None

Returns

<VOLT|FREQ>

STEP:REPetition

This command is used to set the STEP sweep repetitions.

Command syntax

STEP:REPetition <value>

Arguments

<value>: Repetitions

Query syntax

STEP:REPetition?

Returns

<value>

STEP:REPetition?

This command is used to query the STEP sweep repetitions.

Command syntax

STEP:REPetition?

Arguments

None

Returns

<value>

Chapter10 Protection commands

PROTECT:RMS:CURRENT

This command is used to set the RMS protection current.

Command syntax

```
PROTECT:RMS:CURRENT <PARALLEL|THREEphase>,<float>
```

Arguments

Parameter 1<PARALLEL|THREEphase>: Device working mode

PARALLEL,THREEphase

Parameter 2<float>: Current rms

Query syntax

```
PROTECT:RMS:CURRENT? <PARALLEL|THREEphase>
```

Returns

<float>

PROTECT:RMS:CURRENT?

This command is used to query the RMS protection current.

Command syntax

```
PROTECT:RMS:CURRENT? <PARALLEL|THREEphase>
```

Arguments

Parameter 1<PARALLEL|THREEphase>: Device working mode

PARALLEL,THREEphase

Returns

<float>

PROTECT:RMS:TIME

This command is used to set the delay time of RMS protection current.

Command syntax

```
PROTECT:RMS:TIME <PARALLEL|THREEphase>,<float>
```

Arguments

Parameter 1: Device working mode

PARA,THREephase

Parameter 2: rms current time.

Query syntax

PROTect:RMS:TIME? <PARAlle|THREephase>

Returns

<float>

PROTect:RMS:TIME?

This command is used to query the delay time of RMS protection current.

Command syntax

PROTect:RMS:TIME? <PARAlle|THREephase>

Arguments

Parameter 1: Device working mode

PARA,THREephase

Returns

<float>

PROTect:RMS:CTR

This command is used to set the control mode of RMS protection current.

Command syntax

PROTect:RMS:CTR <PARAlle|THREephase>,<IMME|TIME>

Arguments

Parameter 1: Device working mode

PARA,THREephase

Parameter 2: RMS Current control mode

IMME

TIME

Query syntax

PROTect:RMS:CTR? <PARAlle|THREephase>

Returns

<IMME|TIME>

PROTECT:RMS:CTR?

This command is used to query the control mode of RMS protection current.

Command syntax

```
PROTECT:RMS:CTR? <PARALLEL|THREEphase>
```

Arguments

Parameter 1: Device working mode

PARA, THREEphase

Returns

```
<IMME|TIME>
```

PROTECT:PEAK:CURRENT

This command is used to set the peak protection current.

Command syntax

```
PROTECT:PEAK:CURRENT <PARALLEL|THREEphase>,<float>
```

Arguments

Parameter 1: Device working mode

PARA, THREEphase

Parameter 2: Current peak

Query syntax

```
PROTECT:PEAK:CURRENT? <PARALLEL|THREEphase>
```

Returns

```
<float>
```

PROTECT:PEAK:CURRENT?

This command is used to query the peak protection current.

Command syntax

```
PROTECT:PEAK:CURRENT? <PARALLEL|THREEphase>
```

Arguments

Parameter 1: Device working mode

PARA, THREEphase

Returns

<float>

PROTECT:PEAK:TIME

This command is used to set the delay time of peak protection current.

Command syntax

PROTECT:PEAK:TIME <PARALLEL|THREEphase>,<float>

Arguments

Parameter 1: Device working mode

PARA,THREEphase

Parameter 2: PEAK Current time

Query syntax

PROTECT:PEAK:TIME? <PARALLEL|THREEphase>

Returns

<NRf>

PROTECT:PEAK:TIME?

This command is used to query the delay time of peak protection current.

Command syntax

PROTECT:PEAK:TIME? <PARALLEL|THREEphase>

Arguments

Parameter 1: Device working mode

PARA,THREEphase

Returns

< float>

PROTECT:PEAK:CTR

This command is used to set the control mode of peak protection current.

Command syntax

PROTECT:PEAK:CTR<PARALLEL|THREEphase>,<IMME|TIME>

Arguments

Parameter 1: Device working mode

PARA,THREephase

Parameter 2: PEAK Current control mode

IMME

TIME

Query syntax

PROTect:PEAK:CTR? <PARAllel|THREephase>

Returns

<IMME|TIME>

PROTect:PEAK:CTR?

This command is used to query the control mode of peak protection current.

Command syntax

PROTect:PEAK:CTR? <PARAllel|THREephase>

Arguments

Parameter 1: Device working mode

PARA,HREephase

Returns

<IMME|TIME>

PROTect:RMS:TIME:MAX?

This command is used to query the maximum RMS time.

Command syntax

PROTect:RMS:TIME:MAX? <PARAllel|THREephase>

Arguments

Parameter 1: Device working mode

PARA,HREephase ,

Returns

<float>

PROTect:RMS:TIME:MIN?

This command is used to query the minimum RMS time.

Command syntax

PROTECT:RMS:TIME:MIN? <PARALLEL|THREephase>

Arguments

Parameter 1: Device working mode

PARA,HREephase ,

Returns

<float>

PROTECT:PEAK:TIME:MAX?

This command is used to query the maximum PEAK time.

Command syntax

PROTECT:PEAK:TIME:MAX? <PARALLEL|THREephase>

Arguments

Parameter 1: Device working mode

PARA,HREephase

Returns

<float>

PROTECT:PEAK:TIME:MIN?

This command is used to query the minimum PEAK time.

Command syntax

PROTECT:PEAK:TIME:MIN? <PARALLEL|THREephase>

Arguments

Parameter 1: Device working mode

PARA,HREephase

Returns

<float>

Chapter11 List Operation Commands

LIST:STATe

This command is used to set the state of list mode.

Command syntax

LIST:STATe <DISable|ENABle>

Arguments

<DISable|ENABle>

Query syntax

LIST:STATe?

Returns

<DISable|ENABle>

LIST:STATe?

This command is used to query the state of list mode.

Command syntax

LIST:STATe ?

Arguments

None

Returns

<DISable|ENABle>

LIST:REPeat

This command is used to set the running times of list file.

Command syntax

LIST:REPeat <value>

Arguments

<value> : Running repetitions

Query syntax

LIST:REPeat?

Returns

<vaule>

LIST:REPeat?

This command is used to query the running times of list file.

Command syntax

LIST:REPeat?

Arguments

None

Returns

<value>

LIST:ENDState

This command is used to set end of LIST

Command syntax

LIST:ENDState <0|1>

Arguments

<0|1> : 0 Stop output, 1 Keep last state

Query syntax

LIST:ENDState?

Returns

<0|1>

LIST:ENDState?

This command is used to query end of LIST

Command syntax

LIST:ENDState?

Arguments

None

Returns

<0|1>

LIST:JUMPto

This command is used to set the jump when single cycle of list ends.

Command syntax

LIST:JUMPto <value>

Arguments

<value>: list Start position of next list when previous running ends.

Query syntax

LIST:JUMPto?

Returns

<value>

LIST:JUMPto?

This command is used to query the jump when single cycle of list ends.

Command syntax

LIST:JUMPto?

Arguments

None

Returns

<value>

LIST:RUN:RECOrd?

This command is used to query the record index during LIST running.

Command syntax

LIST:RUN:RECOrd?

Arguments

None

Returns

<value>: LIST Index number of the record in running

LIST:RUN:REPEat?

This command is used to query the repeat number during LIST running.

Command syntax

LIST:RUN:REPEat?

Arguments

None

Returns

<value>

LIST:RECOOrder

This command is used to set one record of the LIST in the format of:

Wave index

Frequency

Amplitude

Time

KAC (rising slope)

Starting phase angle

Stopping phase angle

Offset

Kferq

TYPE (0: time, 1: trigger).

Command syntax

LIST:RECOOrder <value>,<string>

Arguments

<value> : Edit index number of RECORD

<string> : Record description

Example

LIST:RECOOrder 1,"1,50.0,100.0,1,0,90,0,0"

Query syntax

LIST:RECOOrder? <value>

Returns

<string>

LIST:RECOOrder?

This command is used to query one record of the LIST.

Command syntax

LIST:RECOOrder? <value>

Arguments

<value>: Edit index number of RECORD

Returns

<string>: Record description

LIST:CONFigure

This command is used to configure the list, where 0 means invalid and 1 means valid.

Command syntax

LIST:CONFigure

Arguments

None

Query syntax

LIST:CONFigure?

Returns

<0|1>

LIST: CONFigure?

This command is used to query the configuration state of list program.

Command syntax

LIST:CONFigure?

Arguments

None

Returns

<0|1>

LIST:NAME

This command is used to set the file name.

Command syntax

LIST:NAME <string>

Arguments

<string>

Query syntax

LIST:NAME?

Returns

<string>

LIST:NAME?

This command is used to query the file name.

Command syntax

LIST:NAME?

Arguments

None

Returns

<string>

LIST:RECall

This command is used to recall a list file.

Command syntax

LIST:RECall <string>

Arguments

<string>: LIST file name.

Returns

None

LIST:RECOrd:NUM

This command is used to set the LIST step number

Command syntax

LIST:RECOrd:NUM <value>

Arguments

<value>: Step number

Query syntax

LIST:RECOrd:NUM?

Returns

<value>

LIST:RECOrd:NUM?

This command is used to query the LIST step number

Command syntax

LIST:RECOrd:NUM?

Arguments

None

Returns

<value>

LIST:SAVE

This command is used to save the list file in a nonvolatile memory.

Command syntax

LIST:SAVE

Arguments

None

Returns

None

LIST:RECAII:NAME?

This command is used to query the file name of the LIST specified by the index number.

Command syntax

LIST:RECAII:NAME? <value>

Arguments

<value>:LIST index number

Returns

<string>

LIST:RECAI:NUM?

This command is used to query the number of valid LIST files saved in current device.

Command syntax

LIST:RECAI:NUM?

Arguments

None

Returns

<value>

LIST:RUN

This command is used to run the list file.

Command syntax

LIST:RUN

Arguments

None

Returns

None

LIST:STOP

This command is used to stop the list file.

Command syntax

LIST:STOP

Arguments

None

Returns

None

LIST:RUN:STATe?

This command is used to set the running state of LIST: ON/OFF.

Command syntax

LIST:RUN:STATe?

Arguments

None

Returns

<0|1>

Chapter12 Common Commands

*CLS

This command clears the following registers:

- Standard event register
- Query event register
- Status byte register

Command syntax

*CLS

Arguments

None

*ESE

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

Register description:

Standard Event Status Register

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PON	NU2	CME	EXE	DDE	QYE	NU1	OPC

OPC: Operation Complete

NU1: Retain 1

QYE: Queryerror

DDE: Device-dependent error

EXE: Execution error

CME: Command error

PON: Power-on has occurred

Command syntax

*ESE <value>

Arguments

0~255

Default set

Refer to *PSC command

Example

*ESE 128

Query syntax

*ESE?

Returns

<value>

Related Command

*ESR? *PSC *STB?

***ESE?**

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

Command syntax

*ESE ?

Arguments

0~255

Default set

Refer to *PSC command

Example

*ESE?

Returns

<vaule>

Related Command

*ESR? *PSC *STB?

***ESR?**

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

Query syntax`*ESR?`**Arguments**

None

Returns

<value>

Related Command`*CLS *ESE *ESE? *OPC`***IDN?**

This command is used to query related information of AC power.

Query syntax`*IDN?`**Arguments**

None

Returns

<string>

Example`*IDN?`***OPC**

When all commands before this command are executed, bit OPC in standard event register will be set to 1.

Command syntax`*OPC`**Arguments**

None

Query syntax`*OPC?`**Returns**

None

*OPC?

When all commands before this command are executed, bit OPC in standard event register will be set to 1 and return to 1.

Command syntax

*OPC?

Arguments

None

Returns

1

*RST

This command resets the frequency to 50 and other parameters to 0 of AC power supply in the setup mode and meter mode.

Command syntax

*RST

Arguments

None

*SRE

This command sets the condition of the Status Request Enable Register. After executing this command, AC power supply will return back a decimal value which is a binary weighted sum of all bits from the enable register.

Command syntax

*SRE <value>

Arguments

0~255

Default set

Refer to *PSC command

Example

*SRE 128

Query syntax

*SRE?

Returns

<value>

Related Command

*ESE *ESR? *PSC *STB?

***SRE?**

This command query the condition of the Status Request Enable Register. After executing this command, AC power supply will return back a decimal value which is a binary weighted sum of all bits from the enable register.

Command syntax

*SRE ?

Arguments

0~255

Default set

Refer to *PSC command

Example

*SRE?

Returns

<value>

Related Command

*ESE *ESR? *PSC *STB?

***STB?**

This command can read the data from status byte register. After execute this command, the value of bit6 in status byte register will be clean.

Status Byte Register

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPER	RQS	ESB	MAV	QUES	EAV	ALARM	NU1

NU1: Retain 1

ALARM: Alarm summary tag

EAV: Error queue not empty

QUES: Questionable status summary

MAV: Message available

ESB: Event status byte summary

RQS: Request for service

OPER:Operation status summary

Query syntax

*STB?

Arguments

None

Returns

<value>

Related Command

*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[:TRANsient][:IMMediate]**

Command syntax

*TRG

Arguments

None

Related Command

TRIG TRIG:SOUR

***TST?**

This command is used to query the self-test by the instrument. 0 indicates that self-test succeeds; other Arguments indicate self-test fails. In addition, in case of failure, an error information will be generated to describe failure reason.

Query syntax

*TST?

Arguments

None

Returns

<NR1>

***WAI**

This command instructs the power supply not to process any further commands until all pending operations are completed. Pending operations are as defined under the *OPC command. *WAI can be aborted only by sending the power supply a device clear command.

Command syntax

*WAI

Arguments

None

Related Command

*OPC

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.