Chapter 1 Introduction

Purpose

The purpose of this manual is to help you remotely control your Array 372X series electronic load from a controller using SCPI programming language with SCPI commands. It is assumed you have completed the following:

1 The electronic load has been installed properly and is operated normally from its front panel.

2 The controller has been connected to the GPIB or RS-232 interface of the electronic load and the related parameters for the interface have been set.

Caution: Interface parameters such as GPIB address, RS-232 baud rate and data bit must be set from the front panel of the electronic load. Please refer to your 372X User's Guide for details.

Supplied Documentation

Every Array 372X series electronic load comes with the following electronic load documentation:

• User's Guide It instructs how to install and handle basic operations, including the local operation from the front panel. Be sure to read it first.

• SCPI Programming Manual It explains how to use SCPI commands to remotely control Array 372X series electronic load from a controller using SCPI programming language.

Reference Documentation

The following documents facilitate you to get a better understanding of GPIB interface and programming in SCPI:

• *ANSI/IEEE Std. 488.1-1987 IEEE Standard Digital Interface for Programmable Instrumentation.*

• *Standard Commands for Programmable Instruments VERSION 1999.0.*

About This Manual

This manual contains the information concerns programming Array 372X electronic load.

Chapter 1	Introduction to this manual.
Chapter 2	The basics about the message structure, syntax and data format for SCPI commands.
Chapter 3	Language dictionary
Chapter 4	Status Reporting
Appendix	Error Messages

What You Should Already Know

This manual does not assume that you have already known SCPI very well or you are a programmer. It is supposed that you have already known:

• the basics of GPIB interface.

• how to send and receive ASCII data between a computer and an instrument over GPIB or RS-232 interface.

• how to input and output the SCPI statements as ASCII strings with the programming language you are using.

• the basic operations of the electronic load introduced in the User's Guide.

Chapter 2 Introduction to Programming

2.1 GPIB Capabilities of the Electronic load

GPIB interface is optional for the electronic load and it must be set from the front panel. Set GPIB interface in "Interface" option after pressing "Menu" key to enter into setting menu. "Interface" option is saved in non-volatile memory. All electronic load functions except for setting GPIB address are programmable over GPIB interface. When GPIB interface is selected, other interfaces are closed. Table 1-1 lists the IEE488.2 capabilities of the electronic load.

Table 1-1 Capabilities of the Electronic load

GPIB Capabilities	Response	Interface Function
Talker/Listener	All electronic load functions except for setting GPIB address are programmable over GPIB interface. The electronic load can send and receive messages over GPIB. Status information is sent by a serial poll.	AH1, SH1, T6. L4
Service Request	The electronic load sets SRQ signal true if an enabled service request condition occurs.	SR1
Remote/Local	The electronic load is in local mode at power-up, and is controlled from the front panel. When the electronic load receives a command over GPIB, it enters into remote mode. In remote mode, the front panel REM annunciator is on and all front panel keys (except Local key) are disabled. Pressing 2nd+Local returns the electronic load to local mode.	RL1

Device Trigger	The electronic load will respond to device trigger function.	DT1
Group Execute Trigger	The electronic load will respond to group execute trigger function.	GET
Device Clear	The electronic load responds to the Device Clear (**DCL**) and Selected Device Clear (**SDC**) interface commands. They cause the electronic load to clear any operation that may prevent it from receiving and executing a new command (including ***WAI** and ***OPC?**). **DCL** and **SDC** do not change any programmed settings.	DCL, SDC

GPIB Address

The GPIB address is set from the front panel. Set GPIB address via “GPIB Address” option in “Menu”. GPIB address ranges from 0 to 30, and is saved in non-volatile memory.
Note: if GPIB interface is not selected, “GPIB Address” option will not be found in “Menu”.

2.2 RS-232 Capabilities of the Electronic load

The electronic load is equipped with RS-232 interface, which must be set from the front panel. Set RS-232 interface in “Interface” option by pressing “Menu” key to enter into setting menu. “Interface” option is saved in non-volatile memory. All SCPI commands are programmable over RS-232 interface. When RS-232 interface is selected, other interfaces are closed.
ELA RS-232 Standard defines how Data Terminal Equipment（DTE） and Data Communications Equipment (DCE) interconnects with each other. The electronic load, as a kind of DTE, can be connected to other DTE (e.g., a PC COM Port) with a null modem cable.
Array 372X series electronic load can program RS-232 interface in “Menu”. Please make sure the settings of the interlinked equipments are matched, or you will fail to connect them properly.
Note: if RS-232 interface is not selected, RS-232–related options will not be found in “Menu”.

RS-232 Data format

RS-232 data is composed of one start bit, one or two stop bits and seven or eight data bits. For party check, you can select among odd, even and none. All parameters are set in “Menu”.
Data Bit: Select seven or eight data bits.
Stop Bit: Select one or two stop bits.
Party Check: None
Even
Odd
The data format is saved in non-volatile memory.

Baud Rate

Baud rate can be set via “Baud Rate” option in “Menu”. Its parameters are saved in non-volatile memory. The electronic load supports the following baud rates: 2400, 4800, 9600, 19200 and 38400. The default value is 9600bps.

RS-232 Flow Control

The electronic load supports the following flow control options that are selected via “Flow Control” option in “Menu”. Selecting “On” enables DTR-DSR to conduct flow control and selecting “Off” disables flow control. Flow control parameters are saved in non-volatile memory.
DTR-DSR The electronic load sets DTR signal to “false” to indicate that the bus controller will hold on its transmission when its receive buffer is almost full. The load also monitors DSR line to determine if the bus controller is ready to receive data. The load sends data to the bus controller only when DSR line is “true”.

2.3 Introduction to SCPI

SCPI (Standard Commands for Programmable Instruments) is a programming language controlling instrument over GPIB or RS-232 interface. SCPI is layered on top of the hardware-portion of IEEE488.2. The same SCPI commands and parameters control the same functions for different categories of instruments.

Conventions for This Manual

For a convenient description, the subsequent symbols are defined as follows:

Angle brackets （<>）	Items within angle brackets are parameter type in abbreviations.
Square brackets ([])	items within square brackets are optional.
Braces（{}）	Parameters within braces can be repeated zero or more times.
Vertical bar（|）	Alternative parameters is separated by a vertical bar.

Types of SCPI Commands

SCPI has two types of commands: common ones and subsystem ones.
Common commands are the general term for a category of commands. They, defined by IEEE488.2 Standard, commonly not related to a specific operation but to controlling overall load functions, such as reset, synchronization, status setting and query. Every common command is composed of a three-letter mnemonic and an asterisk: *RST, *IDN?, *SAV.
Subsystem commands focuses on specific electronic load functions. They are organized into an inverted tree structure with the “root” at the top. Figure1-1 shows a part of a subsystem command tree, which can facilitate you to operate commands at various levels.

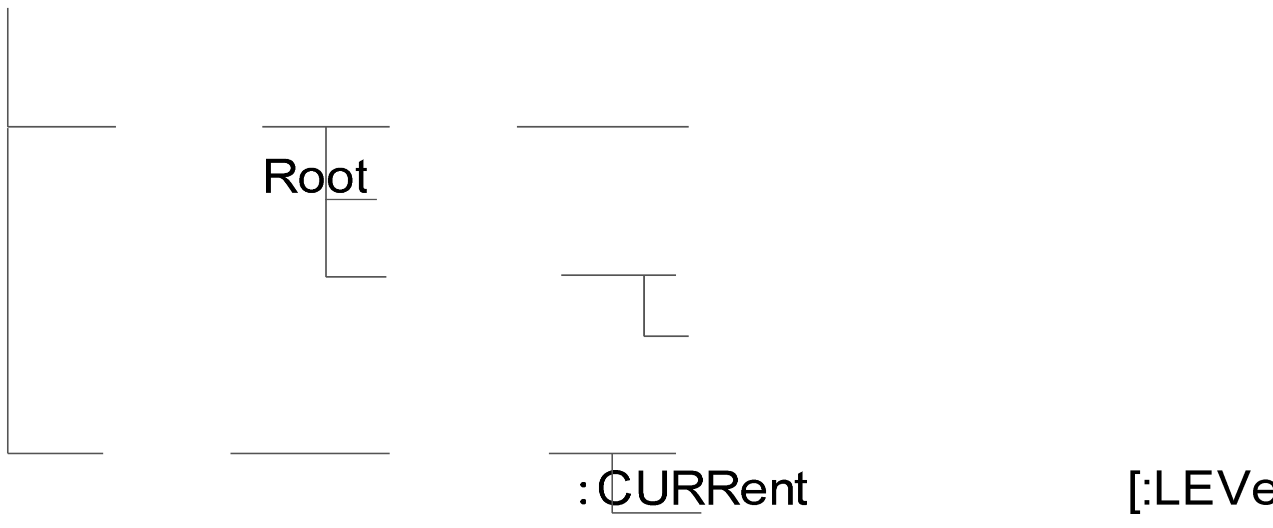


Figure1-1 Partial Command Tree

Multiple SCPI Commands in a Message

Multiple SCPI commands can be combined and sent as a single SCPI message with one message terminator. The following points should be considered when sending several messages within a single message:

- Use a semicolon (;) to separate multiple commands in a message.
- There is always an implied header path that affects the method in which the electronic load analyzes each command.

The header path can be thought of as a character string inserted before each command in a message. For the first command within a message, the header path is a null string. For the subsequent command, the header path is defined as a character string that makes up the headers of previous command in the message up to and including the last colon separator. This is an example of two commands within a message:

CURR:LEV 3;**PROT:STAT** OFF

It illustrates how to separate two commands using the semicolon and explains the header path concept as well. Note that for the second command, the first header "CURR" was omitted because the header path was defined as "CURR:" after the "**CURR:LEV** 3" command and thus the instrument interpreted the second command as :

CURR:PROT:STAT OFF

In fact, it would have generated a syntactic error to include "CURR:" improperly in the second command, since the command after being combined with the header path would become: **CURR:CURR:PROT:STAT** OFF

that is incorrect.

Moving among Subsystems

For the purpose of combining commands from different subsystems, it requires to reset the header path to a null string within a message by beginning the command with a root specifier (:) to

discard any preceding header path. For example, you can use a root specifier to clear the input protection status and check the status of Operation Condition register in a message as follows:
INPut**:PROT**ection**:CLE**ar;:**STAT**us**:OPER**ation**:COND**ition?
The following message shows how to combine commands from different subsystems and within the same subsystem:
INPut**:PROT**ection**:CLE**ar;:**STAT**us**:OPER**ation**:COND**ition?

Including Common Commands

Common commands can be combined with subsystem commands in a message. Treat common command as a message unit by using a semicolon to separate it with other commands. Since common commands do not affect header path, they can be inserted anywhere in a message.
VOLTage**:TRIG**gered 17.5;**:INIT**ialize;***TRG**
INPut OFF;***RCL** 2;**INP**ut ON

Using Queries

Using queries has the following concerns:
- Specify proper number of variables for the data returned by queries.
- Read all the returned data of a query before sending another command to the electronic load. Otherwise, there will be a query interrupted error and the unreturned data will be lost.

Types of SCPI Messages

There are two types of SCPI messages: program and response.
A program message consists of one or more properly formatted SCPI commands sent to the electronic load from the controller. A program message can be sent at any time to request the electronic load to perform some operation.
A response message consists of data in a specific SCPI format sent to the controller from the electronic load. The electronic load sends the response message only when receiving a program message called a “query’.
SCPI message structure is showed as follows:

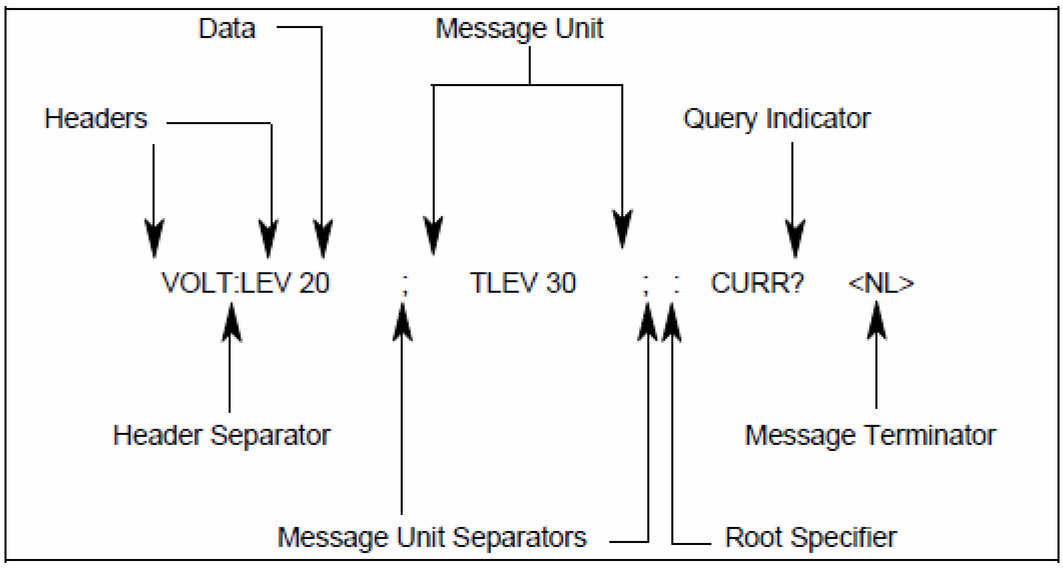


Figure 1-2 SCPI Message Structure

The Message Unit

The simplest SCPI command is a single message unit consists of a command header (or keyword) followed by a command terminator. There may be a parameter after the header in a message unit. The parameter can be numeric or a string.
ABORt <NL>
VOLTage 20<NL>

Headers

Headers, also known as keywords, are instructions can be analyzed and recognized by the electronic load. Header may be either in the long form or the short form. In the long form, the header is completely spelt out to identify its function, such as STATUS, RESISTANC and TRIGGER. In the short form, the header is represented by the first three or four letters of the long form, such as STAT, RES and TRIG.
The short format is constructed according to the following rules:
●For a keyword with four or less letters, all letters should be employed in the short format.
●For a keyword with five or more letters,
if the fourth one is a vowel (a, e, i, o, u), the first three letters are used;
if the fourth one is not a vowel, the first four letters are used.
In this manual, the short form part of each keyword is emphasized in boldface upper-class letters:
TRIGger
IMMediate
RESistance
SHORt
SCPI is case-insensitive and is able to receive keywords such as Trig, trig, trigger, TRIGGER. Whatever format you choose to use, you must spell out the boldface letters or all letters of a

keyword. For example, RES and TRI are not accepted as correct commands.

Query Indicator

A header followed by a question mark forms a query (**VOLT**age?, **VOLT**age**:TRIG**ger?). If a query contains parameters, place the question mark after the last header（**VOLT**age**:TRIG**ger? MAX）.

Command Separator

When two or more commands are combined into a compound command, separate the commands with a semicolon.
STATus**:OPER**ation?;**QUES**tionable?

Root Specifier

When the colon precedes the first header of a message unit, it is referred to as a root specifier. It informs the command parser that this is the root or the top of a command tree.

Terminator

The SCPI messages sent to the electronic load must be terminated by a <newline> character. IEEE-488 EOI can function as a <newline> character to terminate a command string. The <Carriage Return> character followed by a <newline> character is also acceptable for a terminator. The termination of a message always resets the header path for the current SCPI statement to its root.

SCPI Data Formats

All programming data and the value returned from the load is ASCII. The data may be the numerical or character string.

Figure1-2 Numeric Data Format

Symbol	Data format
NRl	Figures without decimal point, namely, the decimal point is assumed at the right of the least significant digit. Example: 2730, 02730
NR2	Figures with a decimal point. Example: 2730., 27.30, 02730
NR3	Figure with a decimal point and an exponent. Example:2.730 E+2, 2.730E-2
NRf	A flexible data format, including NRl, NR2 or NR3. Example: 2730, 27.30, 2.730E+2
NRf+	An extensional data format, including NRf, and MIN, MAX. Example: 730, 27.30, 2.730E-2, MIN, MAX. MIN and MAX represent the minimum and maximum limit values, both within the parameter’s range.
Bool	Boolean data. Example: 0|1 or ON|OFF

Suffixes and Multipliers

Numeric data can be followed by a suffix or not. For the data without a suffix, it is assumed that it is measured by the standard unit of the command.

Table 1-3 Suffixes

Category	Preferred suffix	Alternative suffix	Referenced Unit
Current	A		Ampere
Resistance	OHM	MOHM	Ohm, Megohm
Time	S		Second
Amplitude	V		Volt
Power	W		Watt

Table 1-4 Commonly-used Suffix Multipliers

Multiplier	Mnemonic	Definition
1E6	MA	mega
1E3	K	kilo
1E-3	M	milli
1E-6	U	micro
1E-9	N	nano

Character String Format

For control and query commands, the character string may be in one of the forms showed in table 1-5.

Table1-5 Character String Format

Symbol	Character format
〈Bool〉	Boolean data. Example: ON|OFF
〈crd〉	Character Response Data. Example: CURR.
〈aard〉	Arbitrary ASCII Response Data. Undefined 7-bit ASCII is allowed to be returned. An implied terminator is contained in this data type.

SCPI Command Execution

SCPI commands sent to the electronic load are executed sequentially or in parallel. Sequential commands are completed before implementing subsequent commands. And parallel commands allow other commands are processed during the execution of a parallel command. Commands that affect trigger actions are parallel commands.

The *WAI, *OPC and *OPC？ common commands provide different methods to illustrate that all transmitted commands, including parallel commands have completed the operations. The following should be noted in practice:

*WAI It prevents the execution of any subsequent commands until all pending commands have completed.

*OPC？ It puts a 1 in the Output Queue when all pending commands have completed. Since the returned value should be read by your program, *OPC？ can be used to require the controller to continue its subsequent operations until all pending operations have finished.

*OPC　It sets OPC status bit when all pending operations have finished. As you program can read this status by interruption, *OPC permits subsequent commands to be implemented.

Device Clear

You can send a device clear which may be hanging up the GPIB interface to terminate a SCPI command. When the system receives a device clear command, the status registers, the error queue and all configuration states remain the same. Device clear executes the following operations:

・Clear the input and output buffers of the electronic load .

・The electronic load is ready to receive a new command string.

RS-232 Troubleshooting

If you encounter problems communicating over the RS-232 interface, please check the following:

● The computer and the electronic load must be configured for the same rate, number of data bits, number of stop bits, parity check and flow control options.

● Use correct interface cables or adapter. Please note that even though the cable has the suitable connector, the inner wiring may be incorrect.

● The interface cables must be connected to the correct serial port on your computer (COM1, COM2…).

SCPI Conformance Information

SCPI Conformed Commands

The electronic load conforms to SCPI Version 1999.0.

ABOR
INIT[:IMM]
INIT:CONT
TRIG [:IMM]
TRIG:SOUR
[SOUR:]POW[:LEV][:IMM][:AMPL]
[SOUR:]POW[:LEV]:TRIG[:AMPL]
[SOUR:]CURR[:LEV][:IMM][:AMPL]
[SOUR:]CURR[:LEV]:TRIG[:AMPL]
[SOUR:]VOLT[:LEV][:IMM][:AMPL]
[SOUR:]VOLT[:LEV]:TRIG[:AMPL]
[SOUR:]RES[:LEV][:IMM][: AMPL]
[SOUR:]RES[:LEV]:TRIG[:AMPL]
[SOUR:]CURR:PROT[:LEV]
[SOUR:]CURR:PROT:STAT
STAT:QUES[:EVEN]
STAT:QUES:COND
STAT:QUES:ENAB
SYST:ERR
SYST:VER

Non-SCPI Commands

Although the following commands are not standard SCPI commands, their command syntax and parameter form are defined on the SCPI Version 1999.0 basis.

[SOUR:]BATT[:STAT]	**[SOUR:]CURR[:LEV]:HIGH**
[SOUR:]BATT[:DISC]:TIME	**[SOUR:]VOLT[:LEV]:HIGH**
[SOUR:]BATT:TERM:VOLT	**[SOUR:]RES[:LEV]:HIGH**
[SOUR:]BATT[:DISC]:CAP	**[SOUR:]CURR[:LEV]:LOW**
[SOUR:]BATT[:DISC]:CURR	**[SOUR:]VOLT[:LEV]: LOW**
[SOUR:]BATT:CAP:CLE	**[SOUR:]RES[:LEV]: LOW**
[SOUR:]LIST[:STAT]	**[SOUR:]LIST:SAVE**
[SOUR:]LIST:MEMO	**[SOUR:]LIST:LENG**
[SOUR:]LIST:CLE	**[SOUR:]LIST[:STEP]:INS**
[SOUR:]LIST:CHA	**[SOUR:]LIST[:STEP]:EDIT**
[SOUR:]LIST:NUMB	**[SOUR:]LIST[:STEP]:ADD**
[SOUR:]LIST:COUN	**[SOUR:]LIST[:STEP]:DEL**
[SOUR:]CURR:PROT:DEL	**[SOUR:] TRAN:LTIM**
[SOUR:] CURR:RISE:RATE	**[SOUR:] TRAN:RTIM**
[SOUR:] CURR:FALL:RATE	**[SOUR:] TRAN:HTIM**
[SOUR:] TRAN:MODE	**[SOUR:] TRAN:FTIM**
[SOUR:] TRAN [:STAT]	
[SOUR:]MODE	**INP:LIM[:CV]:CURR**
MEAS[:SCAL]:VOLT[:DC]	**INP:PROT:CLE**
MEAS [:SCAL]:CURR[:DC]	**INP[:STAT]**
MEAS [:SCAL]:POW[:DC]	**INP:SHOR[:STAT]**
MEAS [:SCAL]:RES[:DC]	**INP:LATC[:STAT]**
INP:LATC:VOLT[:LEV]	**SYST:REM**
SYST:LOC	
TRIG:FUNC	

Chapter 3 Language Dictionary

3.1 Introduction

This section will give you a thorough introduction to the syntax and parameters for IEEE488.2 common commands and SCPI commands used by 372X series electronic load. Suppose you have got a good understanding of the material in Chapter Two and 372X User's Guide.

Syntax Forms

Long forms are used to introduce command syntax, but only short forms appear in all examples. Using the long form makes your program easy to understand.

Parameters

Most commands come with a parameter and most queries return a parameter. The parameter range is determined by the model of the electronic load. Since the parameters for the sample program in this manual are based on Array 3721A electronic load and the program itself is common for any 372X electronic load, the associated parameters should be reset for other models. Parameters for all models are listed in the following table.

Related Commands

Commands and queries related to original command, which are either directly related to the original command by function or facilitate you to further understand original command.

Presentation Order

This section contains all commands and queries for 372X electronic load and is presented as follows:

- IEEE488.2 common commands, listed in alphabetical order.
- Root Level Commands, A-Z listing, including
    - Single commands.
    - Subsystems. The single subsystem commands are arranged alphabetically under the subsystem.

Programming Parameters

The following table lists the programming parameters for 372X electronic load. Please refer to the User's Guide for more details.

Parameter	Code	Model and Value			
		3720A	3721A	3722A	3723A
CURR <Nrf+> **CURR:TRIG** <Nrf+> **CURR:LOW** <Nrf+> **CURR:HIGH** <Nrf+> **BATT:CURR** <Nrf+>	L H	0~3A 0~30A	0~4A 0~40A	0~2A 0~20A	0~3A 0~30A
CURR:RISE:RATE <Nrf+>	L H	0.1~300mA/us 0.001~3A/us	0.1~400mA/us 0.001~4A/us	0.1~200mA/us 0.001~4A/us	0.1~300mA/us 0.001~4A/us
CURR:FALL:RATE <Nrf+>	L H	0.1~300mA/us 0.001~3A/us	0.1~400mA/us 0.001~4A/us	0.1~200mA/us 0.001~4A/us	0.1~300mA/us 0.001~4A/us
RES <Nrf+> **RES:TRIG** <Nrf+> **RES:LOW** <Nrf+> **RES:HIGH** <Nrf+>	L M H	0.02~2Ω 2~200Ω 20~2000Ω	0.02~2Ω 2~200Ω 20~2000Ω	0.0666~6.66Ω 6.66~666Ω 66.6~6660Ω	0.0666~6.66Ω 6.66~666Ω 66.6~6660Ω
VOLT <Nrf+> **VOLT:TRIG** <Nrf+> **VOLT:LOW** <Nrf+> **VOLT:HIGH** <Nrf+> **BATT:TERM:VOLT**		0~80V	0~80V	0~200V	0~200V
POW <Nrf+> **POW:TRIG** <Nrf+>		0~250W	0~400W	0~200W	0~350W
MEAS:CURR <Nrf+>	L H	0~3A 0~30A	0~4A 0~40A	0~2A 0~20A	0~3A 0~30A
MEAS:RES <Nrf+>	L M H	0.02~2Ω 2~200Ω 20~2000Ω	0.02~2Ω 2~200Ω 20~2000Ω	0.0666~6.66Ω 6.66~666Ω 66.6~6660Ω	0.0666~6.66Ω 6.66~666Ω 66.6~6660Ω
MEAS:VOLT <Nrf+>		0~80V	0~80V	0~200V	0~200V
MEAS:POW <Nrf+>		0~250W	0~400W	0~200W	0~350W
BATT:CAP?		1mAh~3000Ah	1mAh~3000Ah	1mAh~3000Ah	1mAh~3000Ah
BATT:TIME?		1s ~100h	1s ~100h	1s ~100h	1s ~100h
CURR:PROT <Nrf+> **CURR:PROT:DEL** <Nrf+>		0~30A	0~40A	0~20A	0~30A
TRAN:LTIM <Nrf+> **TRAN:HTIM** <Nrf+> **TRAN:RTIM** <Nrf+> **TRAN:FTIM** <Nrf+>		0~655.35ms 0~655.35ms 10us~655.35ms 10us~655.35ms	0~655.35ms 0~655.35ms 10us~655.35ms 10us~655.35ms	0~655.35ms 0~655.35ms 10us~655.35ms 10us~655.35ms	0~655.35ms 0~655.35ms 10us~655.35ms 10us~655.35ms
LIST: NUMB < NR1 >		0~6	0~6	0~6	0~6
LIST:CHIAN < NR1 >		0~6	0~6	0~6	0~6
LIST:DEL < NR1 >		1~50	1~50	1~50	1~50
LIST:LENG?		1~50	1~50	1~50	1~50
LIST:COUNT < NR1 >		1~65535	1~65535	1~65535	1~65535

3.2 IEEE488.2 Common Commands

Common commands are defined by IEEE488.2 standard to perform the basic instrument functions such as recognition, reset, distinguishing how to read and clear a status and how to execute a command and a query. Common commands are accepted and executed when they are sent as separate commands and also as an inserted portion of the instruction sequences for other programs. Performing a common command does not change the parser’s position in the command tree, which still remains in its previous place when the common command is processed. However, this does not mean that common command does not affect subsequent instructions.
The electronic loads respond to 14 required common commands, which control internal operation, synchronization, status and event register, and system data. As 372X series electronic loads have full trigger capability, they all respond to*TRG command. What’s more, the electronic loads allow using six selectable common commands to set and query Status register. If needed, please refer to section 2.2.14 for details.

*CLS

This command clears the following registers:
●Standard Event Register
●Questionable Status Register
●Operation Status Register
●Status Byte Register
●Error Queue.
Command Syntax: *CLS
Parameters: None

*ESE

This command sets the condition of the Standard Event Enable register, which determines which events of the Standard Event register are allowed to set ***ESB** of the Status Byte register. A “1” in the bit position enables the corresponding event of the Standard Event Register. All enabled events of Standard Event register are logically-ORed to set the ESB (BIT 5) of Status Byte register. See Chapter 4 for explanations of the three registers.
Command Syntax: *ESE <NRf>
Parameters: 0～255
Power-on Value: refer to *PSC command
Example: *ESE 100
Related Commands: *PSC, *STB?, *ESE?

*ESE?

This command reads the Standard Event Enable Register
Query Syntax: *ESE?
Parameters: None
Returned Parameters: <NRl>
Relevant Commands: *PSC, *STB?, ***ESE?**

*ESR?

This command reads Standard Event register. Reading Standard Event register clears it. The definition for internal bits of Standard Event register is the same as that for the internal bits of Standard Event Enable Register. Status reporting gives you more information about this register.
Query Syntax: *ESR?
Parameters: None
Returned Parameters: <NRl>
Relevant Commands: *CLS, *OPC

*IDN?

This command queries for the identification information of the instrument. The returned value consists of four strings separated by commas, including information such as manufacturer, product model, firmware version and so on.
Query Syntax: *IDN?
Parameters: None
Returned Parameters: <aard>
Example: ARRAY,3721A,0,1.43-0.0-0.0

String	Information
Array	Manufacturer
3721A	Product model represented by four digits with a letter suffix.
0	always returns 0
1.43-0.0-0.0	Firmware revision level**.** It is composed of three parts. The first filed indicates the firmware revision of the host processor, the second shows that of communication expansion cards (e.g., GPIB), and the third part, a reserved position, is always 0.

*OPC

This command sets Bit OPC (Bit 0) of the Standard Event Register when all pending operations have been completed. Pending operations are completed when:
・All commands sent before an *OPC have been accomplished.
・All trigger actions have been completed and trigger system has returned to the idle state.
* OPC does not prevent subsequent commands from performing. But Bit OPC will not be set until all pending operations are executed.

Command Syntax: *OPC
Parameters: None
Related Commands: *TRG, *WAI, *OPC?

*OPC?

This command makes the electronic load place an ASCII “1” in the output queue when all pending operations have been completed. Pending operations are completed when:
• All commands sent before an*OPC have been accomplished.
• All trigger actions are completed and trigger system has returned to the idle state.
Unlike *OPC, *OPC? stops the execution of all the subsequent commands. When all pending operations are completed, an ASCII “1” is put in the output queue. *OPC is commonly placed at the end of a command line to facilitate the program to monitor the bus data until it receives the “1”.
Note: Do not proceeds *OPC? with the trigger level setting command unless Ext is chosen as the trigger source. TRIG:IMM, *TRG and GPIB bus trigger follow *OPC? will be forbidden to process, stopping system operations. In this case, the only workable way to restore operation is to send a GPIB **DCL** (Device Clear) command to the electronic load.
Query Syntax: *OPC?
Returned Parameters: <NR1>
Related Commands: *OPC, TRIG:SOUR, *WAI

*PSC

This command controls an automatic clearing of the Service Request Enable and Standard Event Status Enable registers when the load is turned on.
1: Prevents the continents of the Service Request Enable and Standard Event Enable registers from being saved, causing them to be cleared automatically at turn-on. This prevents a PON event from clearing SRQ at turn-on.
0: Saves the contents of the Service Request Enable and Standard Event Enable registers in nonvolatile memory and automatically restore them at power on. This permits a PON event to generate SRQ at turn-on.
Command Syntax: *PSC <bool>
Parameters: 0 | 1
Examples: *PSC 0
Related Commands: *PSC?

*PSC?

This command queries if the contents of Service Request Enable and Standard Event Enable registers are stored.
Query Syntax: *PSC?
Returned Parameters: 0 | 1 0: power-on clear flag is false, related registers not cleared at starting

up.

1: power-on clear flag is true, related registers cleared at starting up.

Related Commands: *PSC

*RCL

This command causes the electronic load to recall a set of parameters saved previously by specifying parameters address. *RCL also performs the following operations:

1 Forces an ABORt command before the reset of any parameters. (This removes all pending trigger values.)

2 When all parameters have been recalled, implements an INP:PROT:CLE to clear the protection state of the electronic load.

3 Turns off calibration mode.

The load automatically executes a *****RCL 0 to recall the parameters stored in Location 0 at turn-on. The same parameters are recalled if no parameters have been saved to the address recalled by *RCL.

Command Syntax: *RCL <NR1>

Parameters: 0~9

Examples: *RCL 5

Relevant Commands: *RST, *SAV

*RST

This command causes the load restore to its factory state. *RST also does the following:

1 Forces an ABORt command before resetting any parameters. (This removes all pending trigger values.)

2 When all parameters have been reset, performs an INP:PROT:CLE to clear the protection state of the electronic load.

Command Syntax: *RST

Parameters: None

Related Commands: *RCL, *SAV

*SAV

This command stores the parameters for the current state of the electronic load in non-volatile memory, ten sets of parameters (0～9) saved in total. Please refer to the Table 2-1 in the User's Guide for details. The electronic load is set to the state in Location 0 automatically at turn on. If no state has been saved to Location 0, the factory default state is saved.

Command Syntax: *SAV <NR1>

Parameters: 0~9

Examples: *SAV 5

Related Commands: *RCL, *RST

*SRE

This command sets the condition of the Service Request Enable register, deciding which events of the Status Byte register are allowed to be used for requesting service. Status reporting will give you more explanations about the Service Request Enable register.
Command Syntax: *SRE <NR1>
Parameters: 0~255.
Examples: *SRE 20
Related Commands: *ESE, *ESR, *PSC, *SRE?

*SRE?

This command reads the value of Service Request Enable register.
Query Syntax: *SRE?
Returned Parameters: <NR1>
Related Commands: *PSC

*STB?

This command reads Status Byte register. *STB? is different from a serial query. When *STB? reads the Status Byte register, the MSS bit is returned in Bit 6 and it is not cleared; However, when a serial query reads the same register, RQS bit is returned in Bit 6 and is cleared. Status reporting will give you more explanations about the Status Byte register.
Query Syntax: *STB?
Parameters: None
Returned Parameters: <NR1>

*TRG

If "BUS" is set as the trigger source, this command generates a trigger. It is essentially equivalent to Group Execute Trigger (GET).
Command Syntax: *TRG
Parameters: None
Related Commands: ABOR, INIT, TRIG, TRIG:SOUR

*TST?

This command enables the load to conduct a self-test within a limited range and report the self-test structure, which does not change the mode and parameter settings of the electronic load.
Query Syntax: *TST?
Returned Parameters: <NR1> 0 =test passed
Nonzero indicates a test failure.

*WAI

This command requires the electronic load not to execute any subsequent commands until all the pending operations showed below have been completed.

- All commands sent before an *OPC have been executed.
- All trigger actions have been completed and the trigger system has returned to the idle state.

Only a GPIB DCL (Device Clear) command sent to the load can abort a ***WAI** command**.**
Parameters: None
Relevant Commands: *OPC, *OPC?

3.3 Subsystem Commands

Subsystem commands are applied to specific functions of the electronic load. They are arranged according to functions with a command tree structure. A subsystem is composed of the related function command, which may be a single command or several related ones.

3.3.1 SCPI Root-Level Commands

```
(:ROOT)
  [SOURce:]
    MODE
    CURRent
      [:LEVel]
        [:IMMediate]
          [:AMPLitude]
        :LOW
        :HIGH
        :TRIGgered
          [:AMPLitude]
      :RISE
      :RATE
      :FALL
      :RATE
      :PROTection
        [:LEVel]
        :STATe
        :DELay
    VOLTage
      [:LEVel]
        [:IMMediate]
          [:AMPLitude]
        :LOW
        :HIGH
        :TRIGgered
          [:AMPLitude]
    RESistance
      [:LEVel]
        [:IMMediate]
          [:AMPLitude]
        :LOW
        :HIGH
        :TRIGgered
          [:AMPLitude]
    POWer
      [:LEVel]
        [:IMMediate]
          [:AMPLitude]
        :TRIGgered
          [:AMPLitude]
    TRANsient
      [:STATe]
      :MODE
      :LTIMe
      :HTIMe
      :RTIMe
      :FTIMe
    LIST
      [:STATe]
      :MEMO
      :NUMBer
      :COUNt
      :CHAin
      :SAVE
      :CLEar
      :LENGth?
      [:STEP]
        :ADD
        :EDIT
        :INSert
        :DELete
    BATTery
      [:STATe]
      :TERMinate
        :VOLTage
      :CAPacity
        :CLEar
      [:DISCharge]
        :CURRent
        :CAPacity?
        :TIME?
  INPut
    [:STATe]
    :SHORt
      [:STATe]
    :LATCh
      [:STATe]
      :VOLTage
        [:LEVel]
    :LIMit
      [:CV]
        :CURRent
    :PROTection
      :CLEar
  MEASure
    [:SCALar]
      :CURRent
        [:DC]?
      :VOLTage
        [:DC]?
      :RESistance
        [:DC]?
      :POWer
        [:DC]?
  TRIGger
    [:IMMediate]
    :SOURce
    :FUNCtion
  ABORt
  INITiate
    [:IMMediate]
    :CONTinuous
  STATus
    :QUEStionable
      [:EVENt]?
      :CONDition?
      :ENABle
      :ENABle?
    :OPERation
      [:EVENt]?
      :CONDition?
      :ENABle
      :ENABle?
  SYSTem
    :ERRor
      [:NEXT]?
    :VERSion?
    :REMote
    :LOCal
    :REMote
    :UPDate
    :UPDate
      :CODE
    :GPIB
      :UPDate
    :GPIB
      :UPDate
        :CODE
  HARDware
    :VERSion?
```

ABORt

This command only affects trigger function. It clears all pending trigger settings, all pending trigger operations in transient test or sequence test as well as causes the trigger system return to the idle status. It also resets the WTG bit of the Operation Condition register.

Command Syntax: ABORt

Parameters: None

Examples:

MODE CCH	Set the electronic load to enter into CCH mode.
CURR:TRIG 4	Set the triggered current value to 4A.
INIT	Perform a trigger initialization.
CURR:TRIG?	Query the triggered current level.
4.000E+0	The returned value is 4A.
TRIG	Send a trigger signal to conduct a trigger operation, and the immediate current level is 4A.
CURR:TRIG 6	Set the triggered current level to 6A.
INIT	Perform a trigger initialization.
ABOR	Abort all pending trigger settings and return the trigger system to the idle state.
CURR:TRIG?	Query the triggered current level.
4.000E+0	The returned value is still 4A.
TRIG	Send a trigger signal to conduct a trigger operation. Triggered current value is not triggered and needs to be reset.

Query Syntax: None

Related Commands: CURR:TRIG, VOLT:TRIG, STAT:OPER:COND?

MODE

[SOURce:]MODE

This command selects the operating mode of the electronic load. 372X series electronic load is designed to be operated in the following modes:

Constant Current Mode: CCL CCH

Constant Voltage Mode: CV

Constant Power Mode: CPC CPV

Constant Resistance Mode: CRL CRM CRH

Caution: if the input is at turn-on, the input is cut off temporarily for 5ms to avoid current surge when the electronic load switches its operating mode. If the electronic load is in transient or sequence operation, sending this command interrupts its present operation, shuts off its input and switches it to the corresponding operating mode.

Command	Function
[**SOUR**ce:]**MODE** CCL	Set electronic load to constant current low range mode.

[**SOUR**ce:]**MODE** CCH	Set electronic load to constant current high range mode.
[**SOUR**ce:]**MODE** CV	Set electronic load to constant voltage mode.
[**SOUR**ce:]**MODE** CRL	Set electronic load to constant resistance low range mode.
[**SOUR**ce:]**MODE** CRM	Set electronic load to constant resistance medium range mode.
[**SOUR**ce:]**MODE** CRH	Set electronic load to constant resistance high range mode.
[**SOUR**ce:]**MODE** CPV	Set electronic load to constant power-voltage source mode.
[**SOUR**ce:]**MODE** CPC	Set electronic load to constant power-current source mode.

Command Syntax: [**SOUR**ce:]**MODE** <AARD>
Parameters: CCL|CCH|CRL| CRM|CRH|CV|CPC|CPV
Take acronyms of each operating mode as parameters. The electronic load is in CCH mode by default at power-on.
Examples: MODE CCL Set electronic load to CCL.
Query Syntax: [**SOUR**ce:]**MODE?** Query the present operating mode.
Returned Parameters: <AARD> CCL|CCH|CRL| CRM|CRH|CV|CPC|CPV
Related Commands: None

3.3.2 Current Subsystem

This subsystem controls functions related to current mode.

Command	**Function**
[**SOUR**ce:]**CURR**ent[**:LEV**el][**:IMM**ediate][**:AMPL**itude]	Set the immediate current level for the CC mode.
[**SOUR**ce:]**CURR**ent[**:LEV**el]**:LOW**	Set transient current low level.
[**SOUR**ce:]**CURR**ent[**:LEV**el]**:HIGH**	Set transient current high level.
[**SOUR**ce:]**CURR**ent[**:LEV**el]**:TRIG**gered[**:AMPL**itude]	Set the triggered current level.
[**SOUR**ce:]**CURR**ent**:RISE:RATE**	Set current rise rate.
[**SOUR**ce:]**CURR**ent**:FALL:RATE**	Set current fall rate.
[**SOUR**ce:]**CURR**ent**:PROT**ection [**:LEV**el]	Set current limit at which protection occurs.
[**SOUR**ce:]**CURR**ent**:PROT**ection**:STAT**e ON|1	Enable protection function.
[**SOUR**ce:]**CURR**ent**:PROT**ection**:STAT**e OFF|0	Disable protection function.
[**SOUR**ce:]**CURR**ent**:PROT**ection**:DEL**ay	Set the delay before the current protection is activated.

Related Subsystem: VOLTage, RESistance

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

This command sets the immediate current level for CC mode. When the input is turned on, if the electronic load is in CC mode, this command transfers an immediate current level to the input immediately; if the electronic load is in other modes, the programmed values are saved for the time the load is operated in CC mode.

Command Syntax: [**SOUR**ce:]**CURR**ent[**:LEV**el][**:IMM**ediate][**:AMPL**itude] <NRf+>
Parameters: Figure |MIN|MAX
Unit: A | mA
Examples: CURR 25A Set the immediate current level to 25A.
Query Syntax: [**SOUR**ce:]**CURR**ent[**:LEV**el][**:IMM**ediate][**:AMPL**itude]?
Parameters: None|MIN|MAX

Examples: CURR?	Query immediate current level.
CURR? MIN	Query the minimum immediate current level.
CURR? MAX	Query the maximum immediate current level.
Returned Parameters: <NR3>	Return immediate current level.

Related Commands: CURR:LOW, CURR:TRIG , CURR:RISE:RATE

[SOURce:]CURRent[:LEVel]:LOW

This command sets the transient current low level for transient operation. In the transient operation, the input current switches between the high and the low level, in the method of continuous, pulsed or toggled transient operation. The high level must be set to a value greater than the low level, or the load fails to operate normally in transient operation. If programmed current low level exceeds the range of present operating mode, an error occurs.
Command Syntax: [**SOUR**ce:]**CURR**ent[**:LEV**el]**:LOW** <NRf+>
Parameters: Figure |MIN|MAX
Unit: A| mA
Examples: CURR:LOW 3A Set the transient current low level to 3A.
Query Syntax: [**SOUR**ce:]**CURR**ent[**:LEV**el]**:LOW**?
Parameters: None|MIN|MAX

Examples: CURR: LOW?	Query transient current low level.
CURR: LOW? MIN	Query the minimum transient current low level.
CURR: LOW? MAX	Query the maximum transient current high level

Returned Parameters: <NR3>
Related Commands: CURR:HIGH

[SOURce:]CURRent[:LEVel]:HIGH

This command sets transient current high level for the transient operation. In transient operation, input current switches between the high and the low level. The high level must be set to a value greater than the low level, or the electronic load fails to operate normally in transient operation. If programmed current high level exceeds the range of present operating mode, an error occurs.
Command Syntax: [**SOUR**ce:]**CURR**ent[**:LEV**el]**:HIGH** <NRf+>
Parameters: Figure|MIN|MAX
Unit: A| mA
Examples: CURR:HIGH 5A Set the transient current high level to 5A.
Query Syntax: [**SOUR**ce:]**CURR**ent[**:LEV**el]**:HIGH**?
Parameters: None|MIN|MAX

Examples: CURR: HIGH? Query transient current high level.

CURR: HIGH? MIN Query the maximum transient current low level.

CURR: HIGH? MAX Query the maximum transient current low level.

Returned Parameters: <NR3>

Related Commands: CURR:LOW

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]

This command specifies triggered current value. When the trigger system is initialized, the load automatically sets triggered current level as immediate current value as soon as a trigger signal is received. When the input is turned on, if the load is in CC mode this command changes input current immediately; if the load is in other modes, the programmed values are saved for the time the load is placed in CC mode. The subsequent trigger signal does not change the input if the triggered current level remains the same.

Before a trigger occurs, the trigger system must be initialized by executing **INIT**iate[**:IMM**ediate] or **INIT**iate**:CONT**inuous. Otherwise, it is impossible to activate a trigger.

Command Syntax: [**SOUR**ce:]**CURR**ent[**:LEV**el]**:TRIG**gered[**:AMPL**itude] <NRf+>

Parameters: Figure |MIN|MAX

Unit: A |mA

Examples: CURR:TRIG 5A Set triggered current to 5A.

CURR:TRIG 50mA Set triggered current to 50mA.

Query Syntax: [**SOUR**ce:]**CURR**ent[**:LEV**el]**:TRIG**gered[**:AMPL**itude]?

Parameters: None|MIN|MAX

Examples: CURR:TRIG? Query triggered current value.

CURR:TRIG? MIN Query the minimum triggered current value.

CURR:TRIG? MAX Query the maximum triggered current value.

Returned Parameters: <NR3>

Related Commands: INIT, INIT:CONT

[SOURce:]CURRent:RISE:RATE

This command specifies the current rise rate for CCH and CCL modes. And in CCL mode, the actual current rise rate is one-tenth of the set value.

Command Syntax: [**SOUR**ce:]**CURR**ent**:RISE:RATE** <NRf+>

Parameters: Figure|MIN|MAX

Unit: A | mA/us

Examples: CURR:RISE:RATE 3A Set current rise rate to 3A/us.

Query Syntax: [**SOUR**ce:]**CURR**ent**:RISE:RATE**？

Parameters: None |MIN|MAX

Examples: CURR:RISE:RATE? Query current rise rate.

CURR:RISE:RATE? MIN Query the minimum current rise rate.

CURR:RISE:RATE? MAX Query the maximum current rise rate.

Returned Parameters: <NR3>

Related Commands: CURR:FALL:RATE

[SOURce:]CURRent:FALL:RATE

This command specifies the current fall rate for CCH and CCL modes. And in CCL mode, the actual current fall rate is one-tenth of the set value.

Command Syntax: [**SOUR**ce:]**CURR**ent**:FALL:RATE** <NRf+>

Parameters: Figure|MIN|MAX

Unit: A |mA/us

Examples:	**CURR:FALL:RATE** 3A	Set the current fall rise to 3A/us.

Query Syntax: [**SOUR**ce:]**CURR**ent**:FALL:RATE**？

Parameters: None|MIN|MAX

Examples:	**CURR:FALL:RATE?**	Query current fall rate.
	CURR:FALL:RATE? MIN	Query the minimum current fall rate.
	CURR:FALL:RATE? MAX	Query the maximum current fall rate.

Returned Parameters: <NR3>

Related Commands: CURR:RISE:RATE

[SOURce:]CURRent:PROTection [:LEVel]

The command sets the protection level for the input current. If the input current exceeds the current limit, timers begins to work, PT, which indicates the load is in protection state, is showed on the front panel, but the input is not immediately turned off. When the specified delay time is reached, the overcurrent protection is triggered and the electronic load is cut off with OC displayed. In the meanwhile, OC and PS in the Questionable Status register are set. When the overcurrent condition is removed, OC and PS are reset.

Command Syntax: [**SOUR**ce:]**CURR**ent**:PROT**ection [**:LEV**el] <NRf+>

Parameters: Figure|MIN|MAX

Unit: A|mA

Examples:	**CURR:PROT** 15A	Set the current protection value to 15A.

Query Syntax: [**SOUR**ce:]**CURR**ent**:PROT**ection [**:LEV**el]？

Parameters: None |MIN|MAX

Examples:	**CURR:PROT?**	Query the current limit.
	CURR:PROT? MIN	Query the minimum current protection value.
	CURR:PROT? MAX	Query the maximum current protection value.

Returned Parameters: <NR3>

Related Commands: CURR:PROT:STAT, CURR:PROT:DEL, INP:PROT:CLE

[SOURce:]CURRent:PROTection:STATe

This command enables or disables current protection function.

Command Syntax: [**SOUR**ce:]**CURR**ent**:PROT**ection**:STAT**e <bool>
Parameters: ON（1）|OFF（0） 1=ON, 0=OFF
Examples: CURR:PROT:STAT ON|1 Enable current protection function.
CURR:PROT:STAT OFF|0 Disable current protection function.
Query Syntax: [**SOUR**ce:]**CURR**ent**:PROT**ection**:STAT**e？
Parameters: None
Examples: CURR:PROT:STAT? Check if current protection is on.
Returned Parameters: <NRl>
Related Commands: CURR:PROT:DEL, CURR:PROT

[SOURce:]CURRent:PROTection:DELay

This command sets the delay time before the current protection is activated. When input current reaches or exceeds current limit, the timer begin to work. When the specified delay time is reached, overcurrent protection is triggered and the load input is cut off.

Command Syntax: [**SOUR**ce:]**CURR**ent**:PROT**ection**:DEL**ay <NRf+>
Parameters: Figure |MIN|MAX
Unit: s| ms
Examples: CURR:PROT:DEL 0.5 Set the delay time to 0.5S.
Query Syntax: [**SOUR**ce:]**CURR**ent**:PROT**ection**:DEL**ay？
Parameters: None|MIN|MAX
Examples: CURR:PROT:DEL? Query the delay time.
CURR:PROT:DEL? MIN Query the minimum delay time.
CURR:PROT:DEL? MAX Query the maximum delay time.
Returned Parameters: <NR3>
Related Commands: CURR:PROT, INP:PROT:CLE

3.3.3 Voltage Subsystem

This command controls functions related to voltage mode.

Command	**Function**
[**SOUR**ce:]**VOLT**age[**:LEV**el][**:IMM**ediate][**:AMPL**itude]	Set immediate voltage for CV mode.
[**SOUR**ce:]**VOLT**age[**:LEV**el]**:LOW**	Set transient voltage low level.
[**SOUR**ce:]**VOLT**age[**:LEV**el]**:HIGH**	Set transient voltage high level.
[**SOUR**ce:]**VOLT**age[**:LEV**el]**:TRIG**gered[**:AMPL**itude]	Set triggered voltage level.

Related Subsystems: CURRent, RESistance

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

This command sets the immediate voltage level in CV mode. When the input is turned on, if the electronic load is in CV mode, this command changes the input voltage immediately; if the electronic load is in other modes, the programmed levels are saved for the time the load is placed CV mode.

Command Syntax: [**SOUR**ce:]**VOLT**age[**:LEV**el]**[:IMM**ediate][**:AMPL**itude] <NRf+>
Parameters: Figure |MIN|MAX
Unit: V|mV
Examples: VOLT 5V Set the immediate voltage to 5V.
Query Syntax: [**SOUR**ce:]**VOLT**age[**:LEV**el]**[:IMM**ediate][**:AMPL**itude]?
Parameters: None|MIN|MAX
Examples: VOLT? Query the immediate voltage.
VOLT? MIN Query the minimum immediate voltage.
VOLT? MAX Query the maximum immediate voltage.
Returned Parameters: <NR3>
Related Commands: CURRent, **RES**istance

[SOURce:]VOLTage[:LEVel]:LOW

This command sets transient voltage low level for the transient operation. In transient operation, input voltage switches between the high and the low level, in the method of continuous, pulsed or toggled transient operation. The high level is set to a value greater than low level, or the electronic load fails to operate normally in transient operation. If the programmable voltage low level exceeds range of present operating mode, an error occurs.

Command Syntax: [**SOUR**ce:]**VOLT**age[**:LEV**el]**:LOW** <NRf+>
Parameters: Figure|MIN|MAX
Unit: V|mV-
Examples: VOLT:LOW 5V Set transient voltage low level to 5V.
Query Syntax: [**SOUR**ce:]**VOLT**age[**:LEV**el]**:LOW?**
Parameters: None|MIN|MAX
Examples: VOLT: LOW? Query transient voltage low level.
VOLT: LOW? MIN Query minimum transient voltage low level.
VOLT: LOW? MAX Query maximum transient voltage low level.
Returned Parameters: <NR3>
Related Parameters: VOLT:HIGH, TRAN:LTIM

[SOURce:]VOLTage[:LEVel]:HIGH

This command sets transient voltage high level for the transient operation. In transient operation, input voltage switches between the high and the low level. The high level is set to a value greater than the low level, or the electronic load fails to operate normally in transient operation. If programmable voltage high level exceeds the range specification for the present operating mode, an error occurs.

Command Syntax: [**SOUR**ce:]**VOLT**age[**:LEV**el]**:HIGH** <NRf+>

Parameters: Figure|MIN|MAX

Unit: V|mV

Examples: VOLT:HIGH 10V Set transient voltage high level to10V.

Query Syntax: [**SOUR**ce:]**VOLT**age[**:LEV**el]**:HIGH?**

Parameters: None|MIN|MAX

Examples: VOLT:HIGH? Query transient voltage high level.

VOLT:HIGH? MIN Query the minimum transient voltage low level.

VOLT:HIGH? MAX Query the maximum transient voltage high level.

Returned Parameters: <NR3>

Related Commands: VOLT:LOW

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]

This command specifies triggered voltage value. After the trigger system initialization, the electronic load automatically sets triggered voltage level as immediate voltage value as soon as a trigger signal is received. When the input is turned on, if the load is in CV mode, this command changes input voltage immediately; if the load is in other modes, the programmed levels are saved for the time the load is placed in CV mode. The subsequent trigger signal does not change the input if the triggered current level remains the same.

Before a trigger occurs, the trigger system must be initialized by executing **INIT**iate[**:IMM**ediate] or **INIT**iate**:CONT**inuous. Otherwise, it is impossible to activate a trigger.

.**Command Syntax:** [**SOUR**ce:]**VOLT**age[**:LEV**el]**:TRIG**gered[**:AMPL**itude] <NRf+>

Parameters: Figure|MIN|MAX

Unit: V|mV

Examples: VOLT:TRIG 5V Set triggered voltage level to 5 V.

Query Syntax: [**SOUR**ce:]**VOLT**age[**:LEV**el]**:TRIG**gered[**:AMPL**itude]?

Parameters: None|MIN|MAX

Examples: VOLT:TRIG? Query triggered voltage level.

VOLT:TRIG? MIN Query the minimum voltage level.

VOLT:TRIG? MAX Query the maximum voltage level.

Returned Parameters: <NR3>

Related Commands: INIT, INIT:CONT, TRIG

3.3.4 Resistance Subsystem

This subsystem controls resistance-mode functions.

Command	**Function**

[**SOUR**ce:]**RES**istance[**:LEV**el][**:IMM**ediate][**:AMPL**itude] Set immediate resistance value for CR mode.
[**SOUR**ce:]**RES**istance[**:LEV**el]**:LOW** Set transient resistance low level.
[**SOUR**ce:]**RES**istance[**:LEV**el]**:HIGH** Set transient resistance high level
[**SOUR**ce:]**RES**istance[**:LEV**el]**:TRIG**gered[**:AMPL**itude] Set triggered resistance value.
Related Subsystems: CURRent, VOLTage

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

This command sets the immediate resistance value is CR mode. When the input is turned on, if the electronic load is in CR mode, this command change the load resistance immediately; if the load is in other modes, the programmed levels are saved for the time the load is operated in CR mode.
Command Syntax: [**SOUR**ce:]**RES**istance[**:LEV**el][**:IMM**ediate][**:AMPL**itude] <NRf+>
Parameters: Figure|MIN|MAX
Unit: Ω | mΩ | kΩ
Examples: RES 10Ω Set immediate resistance level to 10Ω.
Query Syntax: [**SOUR**ce:]**RES**istance[**:LEV**el][**:IMM**ediate][**:AMPL**itude]?
Parameters: None|MIN|MAX
Examples: RES?; Query immediate resistance level.
RES? MIN Query the minimum immediate resistance level.

RES? MAX Query the maximum immediate resistance level.

Returned Parameters: <NR3>
Related Commands: CURR, **VOLT**

[SOURce:]RESistance[:LEVel]:LOW

This command sets transient resistance low level for the transient operation. In transient operation, the resistance switches between the high and the low level. The high level is set to a value greater than the low level, or the electronic load fails to operate normally in transient operation. If programmable resistance low level exceeds the range specification for the present operating mode, an error occurs.
Command Syntax: [**SOUR**ce:]**RES**istance[**:LEV**el]**:LOW** <NRf+>
Parameters: Figure|MIN|MAX
Unit: Ω | mΩ | kΩ
Examples: RES:LOW 3Ω Set transient resistance low level.
Query Syntax: [**SOUR**ce:]**RES**istance[**:LEV**el]**:LOW**?
Parameters: None|MIN|MAX
Examples: RES: LOW? Query transient resistance low level.
RES: LOW? MIN Query the minimum transient resistance low level

RES: LOW? MAX Query the maximum transient resistance high level.

Returned Parameters: <NR3>
Related Command: RES:HIGH, TRAN:HTIM

[SOURce:]RESistance[:LEVel]:HIGH

This command sets transient resistance high level for the transient operation. In transient operation, the resistance switches between the high and the low level. The high level is set to a value greater than the low level, or the electronic load fails to operate normally in transient operation. If programmable resistance high level exceeds the range specification for the present operating mode, an error occurs.
Command Syntax: [**SOUR**ce:]**RES**istance[**:LEV**el]**:HIGH** <NRf+>
Parameters: Figure|MIN|MAX
Unit: Ω | mΩ | kΩ
Examples: RES:HIGH 3Ω Set transient resistance high level.

Query Syntax: [**SOUR**ce:]**RES**istance[**:LEV**el]**:HIGH**?
Parameters: None|MIN|MAX
Examples: RES:HIGH? Query transient resistance high level.
RES:HIGH? MIN Query the minimum transient resistance high level.
RES:HIGH? MAX Query the maximum transient resistance high level.

Returned Parameters: <NR3>
Related Commands: RES:HIGH, TRAN:HTIM

[SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude]

This command specifies triggered resistance value. After the trigger system initialization, the electronic load automatically sets triggered resistance level as immediate resistance value as soon as a trigger signal is received. When the input is turned on, if the electronic load is in CR mode, this command changes input resistance immediately; if the load is in other modes, the programmed levels are saved for the time the load is placed in CR mode. The subsequent trigger signal does not change the input if the triggered resistance remains the same.
Before a trigger occurs, the trigger system must be initialized by executing **INIT**iate[**:IMM**ediate] or **INIT**iate**:CONT**inuous. Otherwise, it is impossible to activate a trigger.
Command Syntax: [**SOUR**ce:]**RES**istance[**:LEV**el]**:TRIG**gered[**:AMPL**itude] <NRf+>
Parameters: Figure|MIN|MAX
Unit: Ω | mΩ | kΩ
Examples: RES:TRIG 3Ω Set triggered resistance value.
Query Syntax: [**SOUR**ce:]**RES**istance[**:LEV**el]**:TRIG**gered[**:AMPL**itude]?
Parameters: None|MIN|MAX

Examples: RES:TRIG?	Query triggered resistance value.
RES:TRIG? MIN	Query the minimum triggered resistance value.
RES:TRIG? MAX	Query the maximum triggered resistance value.

Returned Parameters: <NR3>
Related commands: INIT, INIT:CONT

3.3.5 Power Subsystem

This subsystem controls functions related to power mode.

Command	Function
[**SOUR**ce:]**POW**er[**:LEV**el] [**:IMM**ediate][**:AMPL**itude]	Set immediate power level.
[**SOUR**ce:]**POW**er[**:LEV**el]**:TRIG**gered[**:AMPL**itude]	Set triggered power level.

Related Subsystems: CURRent, VOLTage, RESistance

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

This command sets the immediate power level for CP mode.
Command Syntax: [**SOUR**ce:]**POW**er[**:LEV**el] [**:IMM**ediate][**:AMPL**itude] <NRf+>
Parameters: Figure|MIN|MAX
Unit: W|mW
Examples: POW 10W Set immediate power level to 10W.
Query Syntax: [**SOUR**ce:]**POW**er[**:LEV**el] [**:IMM**ediate][**:AMPL**itude]?
Parameters: None|MIN|MAX

Examples: POW?	Query immediate power level.
POW? MIN	Query the minimum immediate power level.
POW? MAX	Query the maximum immediate power level.

Returned Parameters: <NR3>
Related Commands: POW:TRIG

[SOURce:]POWer[:LEVel]:TRIGgered[:AMPLitude]

This command specifies triggered power value. After the trigger system initialization, the electronic load automatically sets triggered power level as immediate power value as soon as a trigger signal is received. When the input is turned off, if the electronic load is in CP mode, this command changes input power immediately; if the load is in other modes, the programmed values are saved for the time the load is placed in CP mode. The subsequent trigger signal does not change the input if the triggered power remains the same.

Before a trigger occurs, the trigger system must be initialized by executing **INIT**iate[**:IMM**ediate] or **INIT**iate**:CONT**inuous. Otherwise, it is impossible to activate a trigger.

Command Syntax: [**SOUR**ce:]**POW**er[**:LEV**el]**:TRIG**gered[**:AMPL**itude] <NRf+>

Parameters: Figure|MIN|MAX

Unit: W|mW

Examples: POW:TRIG 10w	Set the triggered power level to 10W.

Query Syntax: [**SOUR**ce:]**POW**er[**:LEV**el]**:TRIG**gered[**:AMPL**itude]?

Parameter: None|MIN|MAX

Examples: POW:TRIG?	Query triggered power level.
POW:TRIG? MIN	Query the minimum triggered power level.
POW:TRIG? MAX	Query the maximum triggered power level.

Returned Parameters: <NR3>

Related Commands: INIT, INIT:CONT

3.3.6 List Subsystem

This subsystem controls functions related to list test. List test operations guarantee that the load operates in accordance with the preset test steps and the operating mode, load values and duration time for a single test step can be specified. 372X series electronic load can store up to 7 test lists and each one can contain 50 test steps at most.

Different lists can be chained so that when the present list has been executed, the load can automatically execute the next chained list. Lists allow to be processed cyclically and the cycle time, ranging from 1 to 65535, is set by the user. A list can be chained to itself to achieve the endless cycle of executing.

As for the list test, if the adjacent steps differ in operating mode, the load will automatically has a 5ms–delay after the previous step is over to avoid probable current surge. The input will be turned off during the 5ms-delay.

The operations related to list test are completed by carrying out following commands.

Command	**Function**
[**SOUR**ce:]**LIST**[**:STAT**e	Enable or disable the present list.
[**SOUR**ce:]**LIST:NUM**Ber	Specify the number for the list that is to edit or execute.
[**SOUR**ce:]**LIST:MEMO**	Set the memo for the present list.
[**SOUR**ce:]**LIST**[**:STEP**]**:ADD**	Add a list to the end of present list.
[**SOUR**ce:]**LIST**[**:STEP**]**:DEL**ete	Delete the specified steps for the present list.
[**SOUR**ce:]**LIST**[**:STEP**]**:INS**ert	Insert a step to the specified position of present list.
[**SOUR**ce:]**LIST**[**:STEP**]**:EDIT**	Edit the specified steps for the present list.
[**SOUR**ce:]**LIST**[**:STEP**]**:EDIT?**	Query one step for the present list.
[**SOUR**ce:]**LIST:LENG**th?	Query the total number of steps for the current list.

[**SOUR**ce:]**LIST:COUN**t	Set the cycle time for the present list.
[**SOUR**ce:]**LIST:CHA**in	Specify the chain list for the present list.
[**SOUR**ce:]**LIST:CLE**ar	Clear all steps for the present list.
[**SOUR**ce:]**LIST:SAVE**	Save the settings for the present list.

Related Subsystems: TRAN

[SOURce:]LIST[:STATe]

This command enables or disables the present list.
Command Syntax: [**SOUR**ce:]**LIST**[**:STAT**e] <bool>
Parameters: ON| 1;
OFF| 0

Examples: LIST ON;	Enable the present list.
LIST OFF	Disable the present list.

Query Syntax: [**SOUR**ce:]**LIST**[**:STAT**e]?
Returned Parameters: <NR1>

0|OFF	The present list is disabled.
1|ON	The present list is enabled..

Examples: LIST?
Related Commands: LIST:NUMB, LIST:CONT

[SOURce:]LIST:NUMBer

This command specifies the number for the list that is to edit or execute. The electronic load accepts parameters in the range from 0 to 6 and it returns an error for a parameter is outside of the scope.
Command Syntax: [**SOUR**ce:]**LIST:NUMB**er <NR1>
Parameters: 0~6
Examples: LIST:NUMB 2 Specify List 2 to edit or execute.
Query Syntax: [**SOUR**ce:]**LIST:NUMB**er?
Returned Parameters: <NR1>, 0~6
Examples: LIST:NUMB? Query the number for the list that is editing or executing.

Related Commands: LIST:ADD, LIST:CONT

[SOURce:]LIST:MEMO

This command sets the memo for the present list, which consists of upper and lower case letters, digits and a variety of symbols.
Command Syntax: [**SOUR**ce:]**LIST:MEMO** “<aard>”

Parameters: “0x20-0x7f”

Examples: LIST:MEMO “ARRAY” Set the memo for the present list as ARRAY.

Query Syntax: [**SOUR**ce:]**LIST:MEMO?**

Returned Parameters: <AARD>

Examples: LIST: MEMO? Query the memo for the present list.

Related Commands: LIST:ADD, LIST:CONT

[SOURce:]LIST[:STEP]:ADD

This command adds a step to the end of present list. Parameters such as operating mode, set value and duration time are included in this command.

Command Syntax: [**SOUR**ce:]**LIST**[**:STEP**]**:ADD** <AARD>,<NRf>,<NRf>

Parameters: CCL|CCH|CRL|CRM |CRH|CV, Figure|MIN|MAX, Figure|MIN|MAX

Examples: None, Current Unit|Votage Unit|Resistance Unit, mS|uS

Examples: LIST:ADD CCL,1A,1S Add the followings to the end of present list:
CCL, 1A, 1S

Related Commands: LIST:DEL, LIST:INS

[SOURce:]LIST[:STEP]:DELete

This command deletes the specified step for the present list. The number of the step to be deleted functions as the command parameter.

Command Syntax: [**SOUR**ce:]**LIST**[**:STEP**]**:DEL**ete <NR1>

Parameters: 1~50

Examples: LIST:DEL 2 Delete the second step of present list.

Related Commands: LIST:ADD, LIST:INS

[SOURce:]LIST[:STEP]:INSert

This command inserts a step to the specified position of present list. Parameters such as the step number, operating mode, set value and duration are included in the command.

Command Syntax: [**SOUR**ce:]**LIST**[**:STEP**]**:INS**ert <NR1>,<aard>,<NRf>,<NRf>

Parameters: Figure 1~50, CCL|CCH|CRL|CRM |CRH|CV, Figure|MIN|MAX, Figure|MIN|MAX

Examples: LIST:INS 2,CCH,10A,5S Insert to the second step of present list
CCH, 10A, 5S

Related Commands: LIST:DEL, LIST:ADD

[SOURce:]LIST[:STEP]:EDIT

This command edits the specified step for the present list. Parameters such as step number, operating mode, set value and duration are contained in the command.
Command Syntax: [**SOUR**ce:]**LIST**[**:STEP**]**:EDIT** <NR1>,<aard>,<NRf>,<NRf>
Parameters: Figure 1~50,CCL|CCH|CRL|CRM |CRH|CV, Figure|MIN|MAX, Figure|MIN|MAX
Examples: LIST:EDIT 2,CV,10V,10S Edit the second step of present list: CV, 10V, 10S
Query Syntax: [**SOUR**ce:]**LIST**[**:STEP**]**:EDIT?** <NR1>
Parameters: Figure 1~50
Examples: LIST:EDIT? 2 Query the parameters for the second step of present list.
Returned Parameters: CCL|CCH|CRL|CRM |CRH|CV, Figure|MIN|MAX, Figure|MIN|MAX
Related Parameters: LIST:DEL, LIST:ADD

[SOURce:]LIST:LENGth?

This command queries the total number of steps for the current list.
Command Syntax: [**SOUR**ce:]**LIST:LENG**th?
Parameters: None
Returned Parameters: <NR1>
Examples: LIST:LENG? Query the total number of steps for current list.
Related Commands: LIST:NUMB, LIST:EDIT

[SOURce:]LIST:COUNt

The command sets the number of times the list is executed before it is completed. The parameter ranges from 1 to 65535. The electronic load returns an error for a parameter exceeds the scope. If it requires to execute the list infinitely, please use CHAin function to link to the current list itself.
Command Syntax: [**SOUR**ce:]**LIST:COUN**t <NR1>
Parameters: 1~65535
Examples: LIST:COUN 10 Set the load to execute the present list ten times before it follows the next link or stops its operation.
Query Syntax: [**SOUR**ce:]**LIST:COUN**t?
Returned Parameters: <NR1>
Related Commands: LIST:NUMB, LIST:EDIT

[SOURce:]LIST:CHAin

This command specifies the chain list for the current list.

Command Syntax: [**SOUR**ce:]**LIST:CHA**in <NR1>
Parameters: 0~6
Examples: LIST:CHA 2 Set the chain list for the current list as List 2.
Query Syntax: [**SOUR**ce:]**LIST:CHA**in?
Returned Parameters: <NR1>
Examples: LIST:CHA? Query the chain list for the current list.
Related Commands: LIST:NUMB, LIST:MEMO

[SOURce:]LIST:CLEar

This command clears all steps for the current list.
Command Syntax: [**SOUR**ce:]**LIST:CLE**ar
Parameters: None
Examples: LIST:CLE Clear all steps for the current list.
Relative Commands: LIST:SAVE

[SOURce:]LIST:SAVE

This command saves the settings for the current list, including its memo, test steps, cycle times and chain.
Command Syntax: [**SOUR**ce:]**LIST:SAVE**
Parameters: None
Examples: LIST:SAVE Save the settings for the current list.
Related Commands: LIST:CLE

3.3.7 Transient Subsystem

Transient operation allows the electronic load to switch between the high level (LevelH) and the low level (LevelL), which facilitates you to test the dynamic characteristics of the power supply. Transient operation can be executed in CC, CV and CR mode, and has three operating modes: Continuous, Pulse and Toggle.
●Continuous The load periodically switches between LevelH and LevelL.
●Pulse Before a trigger occurs, the load remains at LevelL. While a trigger is received, the load switches to LevelH. And after the input has remained at LevelH for a certain time, the load returns to LevelL again.
●Toggle When a trigger occurs, the load switches to LevelH from LevelL. And when another trigger is received, the load switches to LevelL from LevelH.
The related parameters such as transient low level (LevelL), transient high level (LevelH), time for transient low level (TimeL), time for transient high level (TimeH), time for rising edge (TimeR) and time for falling edge (TimeF) need to be set for the transient operation.

Command	Function
[**SOUR**ce:]**TRAN**sient**:MODE**	Set the operating mode for transient operation.
[**SOUR**ce:]**TRAN**sient [**:STAT**e]	Enable or disable transient operation.
[**SOUR**ce:]**TRAN**sient**:LTIM**e	Set the time for transient low level.
[**SOUR**ce:]**TRAN**sient**:HTIM**e	Set the time for transient high level.
[**SOUR**ce:]**TRAN**sient**:RTIM**e	Set the time for rising edge. 设
[**SOUR**ce:]**TRAN**sient**:FTIM**e	Set the time for falling edge.

Related Subsystem: TRIGger subsytem

The high/low level time ranges from 0 to 655.35ms; the time for rising/falling edge ranges from 10us to 655.35ms. And the resolution is 10us. For a transient operation, parameters such as Von Point and protection current value should be considered in advance. As improper settings may cause the load input to shut off and consequently interrupt the transient operation.

[SOURce:]TRANsient:MODE

This command selects the operating mode for transient operation. The programmable operating modes are Continuous, Pulse and Toggle.

Command Syntax: [**SOUR**ce:]**TRAN**sient**:MODE** <aard>

Parameters: CONTinuous|PULSe|TOGGle

Examples: TRAN:MODE PULS Select pulsed transient operation.

Query Syntax: [**SOUR**ce:]**TRAN**sient**:MODE?**

Returned Parameters: <aard>CONT, PULS, or TOGG

Related Commands: TRIG

[SOURce:]TRANsient [:STATe]

This command enables or disables the transient operation. To conduct a transient test, you should enable transient operation before turning on the load input.

Command Syntax: [**SOUR**ce:]**TRAN**sient [**:STAT**e] <bool>

Parameters: ON|OFF

Examples: TRAN ON Enable the transient operation.

Query Syntax: [**SOUR**ce:]**TRAN**sient [**:STAT**e]**?**

Returned Parameters: <NR1> Value: 0 for OFF, l for ON.

Related Commands: None

[SOURce:]TRANsient:LTIMe

This command set the low level time for the continuous transient operation. This command is invalid for pulsed and toggled transient operation.

Command Syntax: [**SOUR**ce:]**TRAN**sient**:LTIM**e <NRf+>
Parameters: Figure|MIN|MAX
Unit: mS|uS
Examples: **TRAN:LTIM** 500ms Set the time for transient low level to 500ms.
Query Syntax: [**SOUR**ce:]**TRAN**sient**:LTIM**e?
Parameters: None|MIN|MAX
Examples: **TRAN:LTIM?** Query the time for transient low level.
TRAN:LTIM? MIN
TRAN:LTIM? MAX
Returned Parameters: <NR3>
Related Commands: TRAN:HTIM

[SOURce:]TRANsient:HTIMe

This command set the high level time for continuous and pulsed transient operation. This command is invalid for toggled transient operation.

Command Syntax: [**SOUR**ce:]**TRAN**sient**:HTIM**e <NRf+>
Parameters: Figure|MIN|MAX
Unit: mS|uS
Examples: **TRAN:HTIM** 500ms Set the time for transient high level to 500ms.
Query Syntax: [**SOUR**ce:]**TRAN**sient**:HTIM**e?
Parameters: None|MIN|MAX
Examples: **TRAN:HTIM?** Query the time for transient high level.
TRAN:HTIM? MIN
TRAN:HTIM? MAX
Returned Parameters: <NR3>
Related Commands: TRAN:LTIM

[SOURce:]TRANsient:RTIMe

This command sets the time for rising edge in transient operation, namely, the time for the input to rise from transient low level to transient high level.
Command Syntax: [**SOUR**ce:]**TRAN**sient**:RTIM**e <NRf+>
Parameters: Figure|MIN|MAX
Unit: mS|uS
Examples: **TRAN:RTIM** 100 us Set the time for transient rising edge to 100us.
Query Syntax: [**SOUR**ce:]**TRAN**sient**:RTIM**e?
Parameters: None|MIN|MAX
Examples: **TRAN:RTIM?** Query the time for transient rising edge.
TRAN:RTIM? MIN
TRAN:RTIM? MAX

Returned Parameters: <NR3>
Related Commands: TRAN:FTIM

[SOURce:]TRANsient:FTIMe

This command sets the time for falling edge, namely, the time for the input to fall from transient high level to transient low level.
Command Syntax: [**SOUR**ce:]**TRAN**sient**:FTIM**e <NRf+>
Parameters: Figgure|MIN|MAX
Unit: mS|uS
Examples: TRAN:FTIM 200ms Set the time for transient falling edge to 200us.
Query Syntax: [**SOUR**ce:]**TRAN**sient**:FTIM**e?
Parameters: None|MIN|MAX
Examples: TRAN:FTIM? Query the time for falling edge.
TRAN:FTIM? MIN
TRAN:FTIM? MAX
Returned Parameters: <NR3>
Related Commands: TRAN:RTIM

3.3.8 Battery Subsystem

This subsystem is used to test battery discharge capacity. The load adopts constant current discharge to test battery capacity. When the battery voltage decreases to the user-programmed termination voltage, the electronic load will stop discharging. The associated parameters are set according to following commands.

Command	Function
[**SOUR**ce:]**BATT**ery[**:STAT**e]	Enable or disable discharge capacity test.
[**SOUR**ce:]**BATT**ery**:TERM**inate**:VOLT**age	Set discharge termination voltage level.
[**SOUR**ce:]**BATT**ery[**:DISC**harge]**:CURR**ent	Set discharge current level.
[**SOUR**ce:]**BATT**ery[**:DISC**harge]**:TIME?**	Query discharge time.
[**SOUR**ce:]**BATT**ery[**:DISC**harge]**:CAP**acity?	Query discharge capacity.
[**SOUR**ce:]**BATT**ery**:CAP**acity**:CLE**a	Clear discharge time and discharge capacity for the present battery.

[SOURce:]BATTery[:STATe]

This command enables or disables battery discharge operation. In order to test battery discharge capacity, you should enable discharge capacity test before turning the input on.
Command Syntax: [**SOUR**ce:]**BATT**ery[**:STAT**e] <bool>

Parameters: ON|OFF
Examples: BATT ON Enable battery discharge capacity test.
Query Syntax: [**SOUR**ce:]**BATT**ery[**:STAT**e]? Query if battery discharge capacity test in on.
Examples: BATT?
Returned Parameters: <NRI> Value: 0 for OFF, l for ON
Related Commands: INP ON

[SOURce:]BATTery:TERMinate:VOLTage

This command sets the termination voltage for battery discharge operation. The electronic load operates normally with an input voltage higher than the termination voltage. During battery discharge test, battery output voltage decreases with the increase of discharge time. Once battery output voltage falls to a value lower than the termination voltage, the electronic load is turned off and discharge is automatically stopped.
Command Syntax: [**SOUR**ce:]**BATT**ery**:TERM**inate**:VOLT**age <NRf+>
Parameters: Figure|MIN|MAX
Unit: V|mV
Examples: BATT:TERM:VOLT 2V Set termination voltage for battery discharge operation to 2V.
Query Syntax: [**SOUR**ce:]**BATT**ery**:TERM**inate**:VOLT**age? Query the termination voltage for battery discharge operation.
Parameters: None|MIN|MAX
Examples: BATT:TERM:VOLT? Query the termination voltage for battery capacity test.
BATT:TERM:VOLT? MIN Query the minimum termination voltage for battery capacity test.
BATT:TERM:VOLT? MAX Query the maximum termination voltage for battery capacity test.
Returned Parameters: <NR3>
Related Commands: VOLT

[SOURce:]BATTery[:DISCharge]:CURRent

This command sets discharge current level. The electronic load adopts current specified by this command to discharge the battery.
Command Syntax: [**SOUR**ce:]**BATT**ery[**:DISC**harge]**:CURR**ent <NRf+>
Parameters: Figure|MIN|MAX
Unit: A|mA
Examples: BATT:CURR 1A Set the discharge current to 1A.
Query Syntax: [**SOUR**ce:]**BATT**ery[**:DISC**harge]**:CURR**ent?
Parameters: None|MIN|MAX
Examples: BATT:CURR? Query discharge current level.
BATT:CURR? MIN Query the minimum discharge current level.

BATT:CURR? MAX Query the maximum discharge current level

Returned Parameters: <NR3>

Related Commands: BATT:TERM:VOLT

[SOURce:]BATTery[:DISCharge]:TIME?

This command queries total discharge time. The electronic load automatically calculates the discharge time, and the discharge timing stops when discharge terminates.

Command Syntax: [**SOUR**ce:]**BATT**ery[**:DISC**harge]**:TIME?**

Examples: BATT:TIME? Query total discharge time.

Returned Parameters: <NR1>:<NR1>:<NR1>(Hour:Minute:Second)

Related Commands: BATT:TERM:VOLT

[SOURce:]BATTery[:DISCharge]:CAPacity?

This command queried battery discharge capacity.

Command Syntax: [**SOUR**ce:]**BATT**ery[**:DISC**harge]**:CAP**acity?

Parameters: None

Examples: BATT:CAP? Query battery discharge capacity.

Returned Parameters: <NR3>

Related Commands: BATT:CURR?, BATT:TIME?

[SOURce:]BATTery:CAPacity:CLEar

This command clears the discharge time and discharge capacity for current battery. It requires to clear the previous discharge time and discharge capacity recorded by the load before the next discharge test.

Command Syntax: [**SOUR**ce:]**BATT**ery**:CAP**acity**:CLE**ar

Paarameters: None

Examples: BATT:CAP:CLE Clear the discharge time and discharge capacity.

Related Commands: None

3.3.9 Input Subsystem

This subsystem controls functions related to the load input.

Command	**Function**
INPut[**:STAT**e]	Enable or disable the input.

INPut**:SHOR**t[**:STAT**e]	Enable or disable short-circuit test function.
INPut**:LATC**h[**:STAT**e]	Enable or disable Von Latch function.
INPut**:LATC**h**:VOLT**age[**:LEV**el]	Set the Von Point.
INPut**:LIM**it[**:CV**]**:CURR**ent	Set the current limit in CV mode.
INPut**:PROT**ection**:CLE**ar	Clear protection status.

INPut[:STATe]

This command enables or disables the input.
Command Syntax: INPut[**:STAT**e] <bool>
Parameters: ON|OFF
Examples: INP ON Enable the input.
Query Syntax: INPut[**:STAT**e]?
Parameters: None
Examples: INP? Query the input state.
Returned Parameters: <NR1> Value: 0 for OFF), l for ON
Related Command: INP:SHOR

INPut:SHORt[:STATe]

This command enables or disables short-circuit operation. The previous operating mode and settings of the electronic load are not changed when the short-circuit operation is on.
Command Syntax: INPut**:SHOR**t[**:STAT**e] <bool>
Parameters: ON|OFF
Examples: INP:SHOR ON Enable short-circuit operation.
Query Command: INPut**:SHOR**t[**:STAT**e]?
Parameters: None
Examples: INP:SHOR?
Returned Parameters: <NR1> Value: 0 for OFF, l for ON
Related Commands: INP

INPut:LATCh[:STATe]

This command enables or disables Von Latch function. Von Latch latches the active status of the load. If the Von Latch function is disabled and the load receives the command of turning on the input, once the input voltage reaches the Von Point, the load starts to work automatically; and once the input voltage is lower than the Von Point, the load is turned off automatically. If he Von Latch function is enabled, once the input voltage reaches Von Point, the load starts to work and will keep working no matter how the input voltage changes, even though the input voltage is less than the Von Point. The automatically turning on /off of the input can be implemented via setting the Von

Point and Von Latch, which simplifies test operation effectively.
Command Syntax: INPut**:LATC**h[**:STAT**e] <bool>
Parameters: ON|OFF
Examples: INP:LATC ON Enable Von Latch.
Query Syntax: INPut**:LATC**h[**:STAT**e]?
Parameters: None
Examples: INP:LATC? Query if Von Latch is on.
Returned Parameters: <NR1> Value:0 for OFF, l for ON
Related Commands: INP:LATC:VOLT

INPut:LATCh:VOLTage[:LEVel]

This command sets the Von Point. When the input voltage is lower than the Von Point, the load will not be activated even though the input is on. And the load will be activated till the input voltage reaches or exceeds the Von Point. The default Von Point is 0V.
Command Syntax: INPut**:LATC**h**:VOLT**age[**:LEV**el] <NRf+>
Parameters: Figure|MIN|MAX
Unit: V | mV
Examples: INP:LATC:VOLT 2 Set the Von Point to 2V.
Query Syntax: INPut**:LATC**h**:VOLT**age[**:LEV**el]?
Parameters: None|MIN|MAX
Examples: INP:LATC:VOLT? Query the Von Point.
INP:LATC:VOLT? MIN Query the minimum Von Point.
INP:LATC:VOLT? MAX Query the maximum Von Point.
Returned Parameters: <NR3>
Related Commands: INP:LATC

INPut:LIMit[:CV]:CURRent

This command sets the current limit in CV mode. This value specifies the maximum input current in CV mode. When the input current reaches the limit value, and the input voltage is sill higher than the set value, the input current will not rise higher, so that the electronic load enters into CC mode.
Command Syntax: INPut**:LIM**it[**:CV**]**:CURR**ent <NRf+>
Parameters: Figure|MIN|MAX
Unit: A | mA
Examples: INP:LIM:CURR 20 Set the current limit to 20A.
Query Syntax: INPut**:LIM**it[**:CV**]**:CURR**ent?
Parameters: None|MIN|MAX
Examples: INP:LIM:CURR? Query the current limit.
INP:LIM:CURR? MIN Query the minimum current limit.
INP:LIM:CURR? MAX Query the maximum current limit.

Returned Parameters: <NR3>
Related Commands: CURR

INPut:PROTection:CLEar

This command clears the protection status for the electronic load: OC, OV, OP, OT and RV. Once an exceptional condition occurs, the input is turned off immediately and enters into protection status. Besides specific operations, the electronic load fails to respond to other instructions. And normal state is restored by clearing protection status,
Command Syntax: INPut**:PROT**ection**:CLE**ar
Parameters: None
Examples: INP:PROT:CLE Clear protection status.
Related Commands: INP

3.3.10 MEASure

This subsystem queries the measured value of input voltage, current, resistance and power.

Command	Function
MEASure[**:SCAL**ar]**:CURR**ent [**:DC**]?	Query the measured value of input current.
MEASure[**:SCAL**ar]:**VOLT**age [**:DC**]?	Query the measured value of input voltage.
MEASure[**:SCAL**ar]:**RES**istance[**:DC**]?	Query the measured value for the resistance.
MEASure[**:SCAL**ar]**:POW**er[**:DC**]?	Query the measured value of input power.

Related Subsystems: CURRent, VOLTage, RESistance, POWer

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

This command queries the measured value of input current.
Command Syntax: MEASure[**:SCAL**ar]**:CURR**ent [**:DC**]?
Parameters: None
Returned Parameters: <NR3>
Examples: MEAS:CURR?
Related Commands: MEAS:VOLT?, MEAS:RES?, MEAS:POW?

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

This command queries the measured value of input current.
Command Syntax: MEASure[**:SCAL**ar]:**VOLT**age [**:DC**]?

Parameters: None
Returned Parameters: <NR3>
Examples: MEAS:VOLT?
Related Commands: MEAS:CURR?, MEAS:RES?, MEAS:POW?

MEASure[:SCALar]:RESistance[:DC]?

This command queries the measured value of the resistance.
Command Syntax: EASure[**:SCAL**ar]:**RES**istance[**:DC**]?
Parameters: None
Returned Parameters: <NR3>
Examples: MEAS:RES?
Related Commands: MEAS:CURR?, MEAS: VOLT?, MEAS:POW?

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

This command sets the measured value of input power.
Command Syntax: MEASure[**:SCAL**ar]**:POW**er[**:DC**]?
Parameters: None
Returned Parameters: <NR3>
Examples: MEAS:POW?
Related Commands: MEAS:CURR?, MEAS: VOLT?, MEAS: RES?

3.3.11 Trigger Subsystem

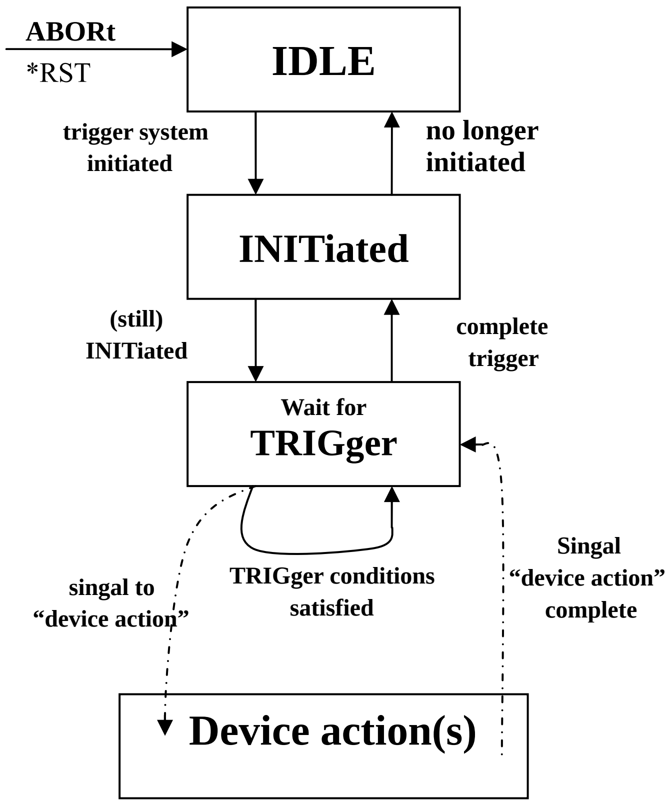
This subsystem sets trigger-related functions. Trigger operation is applied in the following cases:

- Trigger a trigger value.
- Trigger a transient pulse.
- Trigger a transient toggle.
- Trigger a test list.

Command	**Function**
TRIGger[**:IMM**ediate]	Generate a trigger signal.
TRIGger**:SOUR**ce	Set the trigger source.
TRIGger**:FUNC**tion	Set the trigger object.
INITiate[**:IMM**ediate]	Initialize the trigger system.
INITiate**:CONT**inuous	Initialize the trigger system continuously.

Related Subsystems: LIST, TRAN

Trigger Subsystem operates in the pattern showed below:



TRIGger[:IMMediate]

This command generates a trigger signal for any trigger source. There are three methods to trigger for the remote control: GPIB <GET>, ***TRG** and **TRIG**ger command.

Command Syntax: TRIGger[**:IMM**ediate]

Parameters: None

Examples: TRIG

Commands: *TRG

TRIGger:SOURce

This command sets the trigger source.

The electronic load has three trigger sources:

●**BUS:** GPIB<GET> or***TRG** functions as the trigger source.

●**EXTernal:** Select the external trigger input terminal or 2nd key + Trigger key as the trigger source. The input signal at the external trigger input terminal is TTL, the falling edge is triggered

●**HOLD:** Only TRIGger[:IMMediate] command can work as the trigger source. All other trigger methods including *TRG and GPIB<GET> are invalid.

Command Syntax: TRIGger**:SOUR**ce <aard>
Parameters: BUS| EXTernal| HOLD
Examples: TRIG:SOUR EXT Set external trigger as trigger source.
Query Syntax: TRIGger**:SOUR**ce?
Returned Parameters: <aard> BUS| EXTernal| HOLD
Examples: TRIG:SOUR?
Related Commands: TRIG, *TRG

TRIGger:FUNCtion

This command selects the trigger object between LIST and TRAN.
Command Syntax: TRIGger**:FUNC**tion <aard>
Parameters: TRAN| LIST
Examples: TRIG:FUNC LIST Set the trigger object as LIST.
Query Syntax: TRIGger**:FUNC**tion? Query trigger object.
Returned Parameters: <aard> TRAN| LIST
Examples: TRIG:FUNC?
Related Commands: TRIG, *TRG

INITiate[:IMMediate]

This command initializes a trigger operation. Trigger system initialization must be conducted before sending a trigger signal.
Command Syntax: INITiate[**:IMM**ediate]
Parameters: None
Examples: INIT
Related Commands: *TRG

INITiate:CONTinuous

This command turns on/off the continuous initialization function. If this function is enabled, the subsequent trigger operations do not need to initialize the trigger system.
Command Syntax: INITiate**:CONT**inuous 1
Parameters: ON(1)|OFF(0)

Examples: INIT:CONT 1
Query Syntax: INITiate**:CONT**inuous?
Returned Parameters: <NR1> Value: 0 for OFF, 1 for ON.
Examples: INIT:CONT?
Related Commands: *TRG

3.3.12 Status Subsystem

The electronic load uses the following four groups of status registers to record different device statuses: Status Byte register, Standard Event register, Questionable Status register and Operation status register. Status Byte register records the information of other registers.

Command	Function
STATus**:QUES**tionable[**:EVEN**t]?	Query Questionable Status Event register.
STATus**:QUES**tionable**:ENAB**le	Set Questionable Status Enable register.
STATus**:QUES**tionable**:ENAB**le?	Query Questionable Status Enable register.
STATus**:QUES**tionable**:COND**ition?	Query Questionable Status Condition register.
STATus**:OPER**ation[**:EVEN**t]?	Query Operation Status Event register.
STATus**:OPER**ation**:ENAB**le	Set Operation Status Enable register.
STATus**:OPER**ation**:ENAB**le?	Query Operation Status Enable register.
STATus**:OPER**ation**:COND**ition?	Query Operation Status Condition register.

STATus:QUEStionable[:EVENt]?

This command queries Questionable Status Event register. The electronic load returns a decimal value which corresponds to the binary-weighted sum of all bits in the register. After the execution of this command, the value of the Questionable Status Event register is reset.
Query Syntax: STATus**:QUES**tionable[**:EVEN**t]?
Parameters: None
Examples: STAT:QUES?
Returned Parameters: <NR1>

STATus:QUEStionable:ENABle

This command sets Questionable Status Enable register. Select the corresponding bits in Questionable Status Event register by causing the related bits in Questionable Status Enable register to be set to 1. And QUES bit in Status Byte register is set provided any of the selected bits is 1.
Command Syntax: STATus**:QUES**tionable**:ENAB**le <NRf>
Parameters: 0～65535

Power-on Value: Refer to ***PSC** Command
Examples: STAT:QUES:ENAB 64

STATus:QUEStionable:ENABle?

This command queries Questionable Status Enable register. The electronic load returns a decimal value which corresponds to the binary-weighted sum of all bits in the register.
Query Syntax: STATus**:QUES**tionable**:ENAB**le?
Returned Parameters: <NR1>
Examples: STAT:QUES:ENAB?
Related Commands: *PSC

STATus:QUEStionable:CONDition?

This command queries Questionable Status Condition register.
Query Syntax: STATus**:QUES**tionable**:COND**ition?
Parameters: None
Examples: STAT:QUES:COND?
Returned Parameters: <NR1>

STATus:OPERation[:EVENt]?

This command queries Operation Status Event register. The execution of this command resets the value in Operation Status Event register to zero.
Query Syntax: STATus**:OPER**ation[**:EVEN**t]?
Parameters: None
Examples: STAT:OPER?
Returned Parameters: <NR1>

STATus:OPERation:ENABle

This command sets Operation Status Enable register. Select corresponding bits in Operation Status Event register by setting the related bits in Operation Status Enable register to 1. And OPER bit in Status Byte register is set provided any of the selected bits is 1.
Command Syntax: STATus**:OPER**ation**:ENAB**le <NRf>
Parameters: 0～255
Power-on Value: Refer to ***PSC** Command
Examples: STAT:OPER:ENAB 128
Query Syntax: STATus**:OPER**ation**:ENAB**le?

Returned Parameters: <NR1>
Examples: STAT:OPER:ENAB?
Related Commands: *PSC

STATus:OPERation:CONDition?

This command queries Operation Status Condition register.
Query Syntax: STATus:OPERation:CONDition?
Parameters: None
Examples: STAT:OPER:COND?
Returned Parameters: <NR1>

3.3.13 System Subsystem

This subsystem sets system-related functions.

Command	**Function**
SYSTem**:ERR**or[**:NEXT**]?	Query error messages
SYSTem**:VER**Sion?	Query the version number for the current SCPI.
SYSTem:**LOC**al	Set the electronic load to the local mode.
SYSTem**:REM**ote	Set the electronic load to the remote mode.

SYSTem:ERRor[:NEXT]?

This command reads one error from the error queue and the returned value consists of error numbers and error messages. This load can store up to 20 error messages. If there are more than 20 errors, the load replaces the last error stored in the queue with -350, "Too many errors". Error storage memory reads detected errors according to the principle of "first-in first-out". The first returned error is the first saved. When an error is read, it is removed from the error queue. When all errors are read, the electronic load returns "+0, No Error".
*CLS (Clear status command) clears error queue, while *RST does not. All error records are lost when the electronic load is off.
Command Syntax: SYSTem:ERRor[:NEXT]?
Parameters: None
Examples: SYST:ERR? Return the errors in error queue.
Returned Parameters: <NR1>, <aard> Return error number and error explanation.
Related Commands: None

SYSTem:VERSion?

This command queries the SCPI version number for which the load conforms to. The returned value is a character string in the format of YYYY.V. Y represents the year of the release and V represents the revision number for that year. For example: 1995.0.
Command Syntax: SYSTem**:VER**Sion?
Parameters: None
Examples: SYST:VERS?
Returned Parameters: <aard>, <NR2>
Related Commands: *IDN?

SYSTem:LOCal

This command sets the electronic load to stay in the local mode.
Command Syntax: SYSTem:**LOC**al
Parameters: None
Examples: SYST:LOC
Related Commands: SYST:REM

SYSTem:REMote

This command sets the electronic load to the remote mode when the operation is conducted over RS-232 interface. In remote mode, except 2nd+Local, other keys on the front panel are disabled.
Command Syntax: SYSTem:**REM**ote
Parameters: None
Examples: SYST:REM
Related Commands: SYST:LOC

Chapter 4 Status Reporting

This chapter discusses the status registers of 372X series electronic load. You can always decide the operating condition of the electronic load using the status register. For example, you can set the electronic load to generate an interruption (a request service) for the emergencies, such as an overvoltage protection. In consequence, your program can take appropriate measures for such interruptions.
The Status register groups of the electronic load are illustrated in Figure 4-1. Standard Event register, Output Queue, Status Byte register and Service Request Enable register perform the

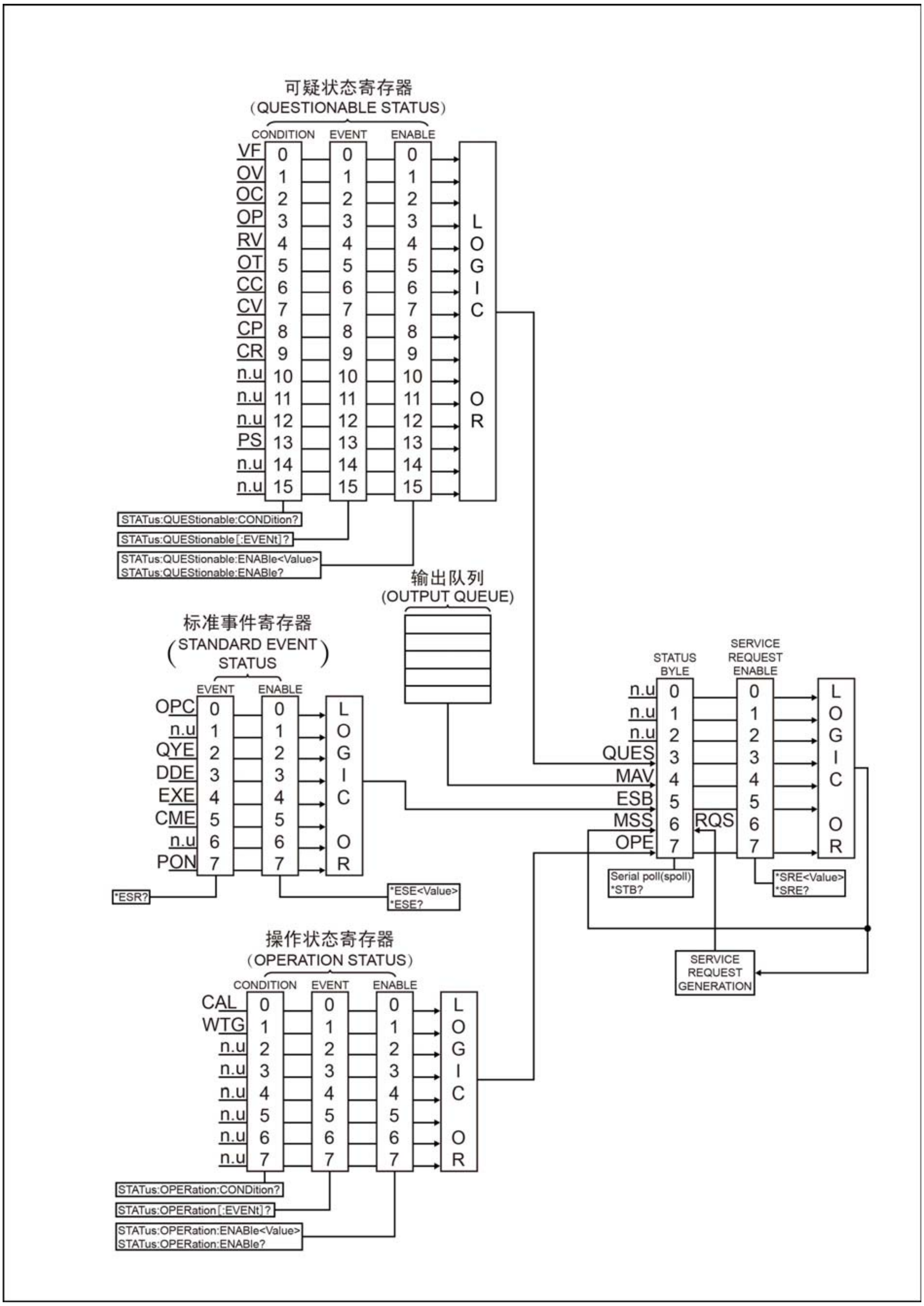
standard GPIB function and defined in IEEE488.2 Standard Digital Interface for Programmable Instrumentation. While Operation Status register and Questionable Status register execute the status reporting function.

The following figure defines each status register:

BIT	Signal	Meaning
		Operation Status register
0	CAL	Calibrating.
1	WTG	Waiting for trigger signal.
		Questionable Status register
0	VF	Voltage fault caused by the reverse voltage on input or overvoltage. VF is set to 1 and remains set until “**INP:PROT:CLE**” is received.
1	OV	Overvoltage. In this condition, the electronic load turns off. OV and VF are set to 1 simultaneously and remain set until overvoltage condition is removed and “**INP:PROT:CLE**” is received.
2	OC	Overcurrent. When the current exceeds the current limit programmed by the user, OC is set to 1 and remains set until overcurrent condition is removed. However, if the overcurrent condition lasts beyond the protection delay time programmed by the user, PS is also set to 1 and the load is cut off. In such circumstances, PS and OC will not be restored until the overcurrent condition is removed and “**INP:PROT:CLE**” is received.
3	OP	Overpower. In this case, the load turns off and OP and PS are set to 1 and remain set until the overpower condition is removed and **INP:PROT:CLE** is received.
4	RV	Reverse Voltage on input. When this occurs, RV and VF are set to 1. And when reverse voltage on input is removed, RV is cleared. But VF remains set until “**INP:PROT:CLE**” is received.
5	OT	Overtemperature. In this case, the electronic load turns off , OP and PS are set to 1 and remain set until the load is cooled down by the electronic load fan and “**INP:PROT:CLE**” is received.
6	CC	Constant Current mode
7	CV	Constant Voltage mode
8	CP	Constant Power mode
9	CR	Constant Resistance mode
13	PS	Protection shutdown The load turns off for overcurrent, overpower or overtemperature, and PS is set to 1 and remains set until “**INP:PROT:CLE**” is received.
		Standard Event Register.
0	OPC	Operation complete. When all parallel operations are complemented, executing “*OPC” command cause OPC to be set to 1.
2	QYE	Query error. When buffer area is read, no data can be found there. Errors from -400 to -499 can set this bit to 1.
3	DDE	Device Dependent Error. The data stored in the instrument memory is

		lost. Errors from -300 to -399 can set this bit to 1.
4	EXE	Execution error. The command parameters exceed the load's limit, or disaccords with the load's operation or some commands fail to be implemented in some conditions. Errors from -200 to -299 can set this bit to 1.
5	CME	Command error. It indicates there is syntax or semantic errors in command messages received. Errors from -100 to -199 can set this bit to 1.
7	PON	Power-on bit. When the load is powered on, this bit is always set to 1.
		Status Byte Register
3	QUES	Questionable. If an enabled questionable event has occurred, QUES bit is 1.
4	MAV	Message Available. If the output buffer contains data, MAV bit is 1.
5	ESB	Event Status Bit. If an enabled standard event has occurred, ESB bit is 1.
6	MSS/ RQS	During a serial poll, RQS bit is returned and cleared. For an *STB query, MSS is returned without being cleared.
7	OPER	Operation. If an enabled operation event has occurred, OPER bit is 1.

The following figure describes the status register groups apply to 372X electronic load.


可疑状态寄存器
(QUESTIONABLE STATUS)
CONDITION
EVENT
ENABLE
VF
OV
OC
OP
RV
OT
CC
CV
CP
CR
n.u
n.u
n.u
PS
n.u
n.u
LOGIC OR
STATus:QUEStionable:CONDition?
STATus:QUEStionable[:EVENt]?
STATus:QUEStionable:ENABle<Value>
STATus:QUEStionable:ENABle?
输出队列
(OUTPUT QUEUE)
标准事件寄存器
(STANDARD EVENT STATUS)
EVENT
ENABLE
OPC
n.u
QYE
DDE
EXE
CME
n.u
PON
LOGIC OR
*ESR?
*ESE<Value>
*ESE?
STATUS BYLE
SERVICE REQUEST ENABLE
n.u
n.u
n.u
QUES
MAV
ESB
MSS
RQS
OPE
LOGIC OR
Serial poll(spoll)
*STB?
*SRE<Value>
*SRE?
SERVICE REQUEST GENERATION
操作状态寄存器
(OPERATION STATUS)
CONDITION
EVENT
ENABLE
CAL
WTG
n.u
n.u
n.u
n.u
n.u
n.u
LOGIC OR
STATus:OPERation:CONDition?
STATus:OPERation[:EVENt]?
STATus:OPERation:ENABle<Value>
STATus:OPERation:ENABle?

Common Register Model

The Condition register reflects the current or live state of various electronic load signals. Reading this register does not change its bits. Only changes in the electronic load changes the bits of this register. Not all status register groups include a Condition register.

The Event register catches the changes in conditions. Each bit in an Event register either corresponds to a responsive bit in a Condition register or to a specific condition in the electronic load. An event turns to be true if the corresponding condition in the electronic load makes one of the following transitions:

●Positive Transition (0 to 1)

●Negative Transition (1 to 1)

●Positive or Negative Transition (0 to 1 or 1 to 0)

The Enable register selects which bits in the Event register are logically-ORed into a summary bit. The Enable register is reset to zero at turn-on. However, Standard Event Enable register and Service Request Enable register are restored to the state before turn-on if ***PSC** 0 command is programmed.

Questionable Status register

The Questionable Status register reports one or more errors have occurred to the device.

- Setting a bit in Questionable Status Condition register indicates the presence of corresponding errors or unusual conditions.
- The Questionable Status Event register represents all questionable events that have happened since the last time this register was read. A condition transition from 0 to 1 on a bit in the Questionable Status Condition register will set the corresponding bit in the Questionable Status Event register. Reading the Questionable Status Event register sets it to zero.
- The Questionable Status Enable register determines which questionable status event bits are logically–ORed to set QUES bit in Status Byte register.

Output Queue

The output queue is a data structure based on a FIFO（first-in, first-out）principle and it stores output messages of the electronic load until they are read. Once there is data in this queue, MAV bit in the Status Byte register is set.

Standard Event Register

Any IEEE488.2 device is typically equipped with a Standard Event register. Any programming errors will set one or more bits in the Standard Event register.

- The Standard Event Register represents all standard events that have happened since the last time this register was read. Reading this register will set it to zero.
- The Standard Event Enable register sets which standard event bits are logically–ORed to set ESB bit in Status Byte register.

Operation Status Register

The Operation Status register records the operation status of the electronic load.

- The Operation Status Condition register reflects the current status of the electronic load.
- The Operation Status Event register represents all selected conditions since the last time the register was read. Reading this register will set it to zero.
- The Operation Status Enable register sets which operation event bits are logically–ORed to set OPER bit in the Status Byte register.

Status Byte Register

The Status Byte register collects all status events from other status registers. It can be read by a serial poll or a ***STB?** command.
When this register is read by a ***STB?** command, Bit 6 of the returned value is the MSS bit. Setting MSS bit manifests that there is at least one reason for the electronic load to request services. It is the result of Inclusive-OR after the bits in the Status Byte register have been screened by Service Request Enable register.
When a serial poll is sent to respond to a service request, Bit 6 of the returned value is the RQS bit. RQS bit is the latched MSS bit. Once the load needs to request services, it sets SQR signal to true and latch RQS bit simultaneously. The RQS bit is automatically cleared after a serial poll. And other bits in the Status Byte register are not influenced by a serial poll.
*STB? does not affect Status Byte register. *CLS clears all related Status registers including Status Byte register.

Service Request Enable Register

The Service Request Enable register specifies setting which bits in the Status Byte register will generate service requests. All bits except Bit 6 (RQS/MSS) can be set to generate service requests.

The enabled bits in the Service Request Enable register are logically-ORed to become the MSS bit in the Status Byte register.

Appendix Error Messages

The following table lists the error numbers returned by the electronic load and the corresponding messages they indicate respectively.

Error Messages

Error Number	Error String (Description/Explanation/Examples)
-104	Data type error [e.g., numeric or string expected, got block data]
-108	Missing parameter[too few parameters]
-113	Undefined header[operation not allowed]
-123	Exponent too large[numeric overflow; exponent magnitude >32 k]
-131	Invalid suffix [unrecognized units, or units not appropriate]
-170	Expression error
-210	Command not allowed with GPIB
-220	Cal sequence wrong
-221	LIST out of range
-222	Data out of range
-223	Output buffer overflow
-230	Cal mode not open
-260	Cal failed, Numeric data out of range
-270	Err cause data not saved
-313	Readback cal data out of range
-350	Too many errors
-410	Query INTERRUPTED[query followed by DAB or GET before response complete]
-440	Query UNTERMINATED after indefinite response