Chroma

High Speed
DC Electronic Load
6330 Series
Programming Manual

Version 1.4
March 2007
P/N A11 000204

Legal Notices

The information in this document is subject to change without notice.

Chroma ATE INC. makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Chroma ATE INC. shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

CHROMA ATE INC.
66 Hwa-Ya 1st Rd., Hwa-Ya Technical Park, Kuei-Shan Hsiang, Taoyuan Hsien, Taiwan

Warranty

All Chroma instruments are warranted against defects in material and workmanship for a period of one year after date of shipment. Chroma agrees to repair or replace any assembly or component found to be defective, under normal use during this period. Chroma's obligation under this warranty is limited solely to repairing any such instrument, which in Chroma's sole opinion proves to be defective within the scope of the warranty when returned to the factory or to an authorized service center. Transportation to the factory or service center is to be prepaid by purchaser. Shipment should not be made without prior authorization by Chroma.

This warranty does not apply to any products repaired or altered by persons not authorized by Chroma, or not in accordance with instructions furnished by Chroma. If the instrument is defective as a result of misuse, improper repair, or abnormal conditions or operations, repairs will be billed at cost.

Chroma assumes no responsibility for its product being used in a hazardous or dangerous manner either alone or in conjunction with other equipment. High voltage used in some instruments may be dangerous if misused. Special disclaimers apply to these instruments. Chroma assumes no liability for secondary charges or consequential damages and in any event, Chroma's liability for breach of warranty under any contract or otherwise, shall not exceed the purchase price of the specific instrument shipped and against which a claim is made.

Any recommendations made by Chroma for use of its products are based upon tests believed to be reliable, but Chroma makes no warranty of the results to be obtained. This warranty is in lieu of all other warranties, expressed or implied, and no representative or person is authorized to represent or assume for Chroma any liability in connection with the sale of our products other than set forth herein.

CHROMA ATE INC.
66 Hwa-Ya 1st Rd., Hwa-Ya Technical Park,
Kuei-Shan Hsiang, Taoyuan Hsien, Taiwan
Tel: 886-3-327-9999
Fax: 886-3-327-2886
http://www.chromaate.com

Material Contents Declaration

A regulatory requirement of The People's Republic of China defined by specification SJ/T 11364-2006 mandates that manufacturers provide material contents declaration of electronic products, and for Chroma products are as below:

Part Name	Hazardous Substances					
	Lead	Mercury	Cadmium	Hexavalent Chromium	Polybrominated Biphenyls	Polybromodiphenyl Ethers
	Pb	Hg	Cd	Cr^{6+}	PBB	PBDE
PCBA	×	O	O	O	O	O
CHASSIS	×	O	O	O	O	O
ACCESSORY	×	O	O	O	O	O
PACKAGE	O	O	O	O	O	O

"O" indicates that the level of the specified chemical substance is less than the threshold level specified in the standards of SJ/T-11363-2006 and EU 2005/618/EC.

"×" indicates that the level of the specified chemical substance exceeds the threshold level specified in the standards of SJ/T-11363-2006 and EU 2005/618/EC.

1. Chroma is not fully transitioned to lead-free solder assembly at this moment; however, most of the components used are RoHS compliant.
2. The environment-friendly usage period of the product is assumed under the operating environment specified in each product's specification.

Disposal

Do not dispose of electrical appliances as unsorted municipal waste, use separate collection facilities. Contact your local government for information regarding the collection systems available. If electrical appliances are disposed of in landfills or dumps, hazardous substances can leak into the groundwater and get into the food chain, damaging your health and well-being. When replacing old appliances with new one, the retailer is legally obligated to take back your old appliances for disposal at least for free of charge.

Revision History

The following lists the additions, deletions and modifications in this manual at each revision.

Date	Version	Revised Sections
May 2001	1.0	Complete this manual.
June 2003	1.1	Correct the errors in CONFigure:VOLTage:LATCh for "CONGIFURE Subsystem" under "Language Dictionary".
June 2005	1.2	Change the address and phone no. of Chroma ATE Inc.
Nov. 2006	1.3	Add the following: – "*SYNCHRONOUS Subsystem*" in "*Specific Commands*" section in the chapter of "*Language Dictionary*". – The description of "*CONFigure:VOLTage:LATCh:RESet*" in "*CONGIFURE Subsystem*" in the chapter of "*Language Dictionary*". Delete the duplicate "*PROGram:RUN*" command and the syntax listed in "*Query Syntax*" that is unable to query in "*PROGRAM Subsystem*" section in the chapter of "*Language Dictionary*".
Mar. 2007	1.4	Add "*Material Contents Declaration*".

Table of Contents

1. General Information

1.1 Introduction

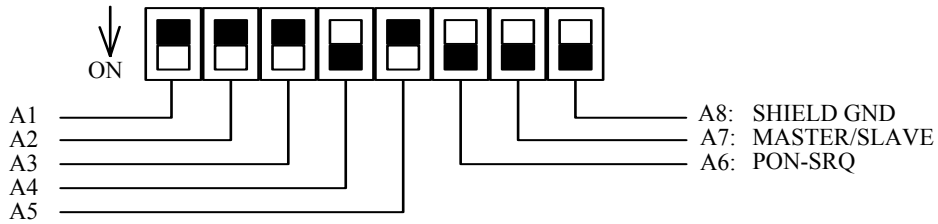
This Programming Manual describes how to program the 6330 Series high speed load remotely from a GPIB controller or RS232C. The command set introduced here can be applied to all electronic loads of 6330 series, including 63301, 63302, 63303, etc. equipped with optional GPIB cards or standard RS232C equipment.

Either GPIB or RS232C can be used one at a time. They cannot be used simultaneously. If GPIB is used first in remote control, RS232C will be disabled unless the machine is reset, and vice versa.

1.2 DIP Switches on the GPIB Card

1.2.1 GPIB Address

Before programming the electronic load remotely via a GPIB computer, you need to know the GPIB address. Each device connected to the GPIB interface has a unique address assigned to it. Such address allows the system controller to communicate with individual devices. To set the GPIB address of an individual mainframe, Chroma 6332 or 6334, it is done by an 8-bit DIP switch on a GPIB card at the mainframe rear panel. The five bits, from A1 to A5, are GPIB address bits, which offer address space from 0 to 30. For details please refer to the following illustration and table.



Address	A5	A4	A3	A2	A1	Address	A5	A4	A3	A2	A1
0	0	0	0	0	0	16	1	0	0	0	0
1	0	0	0	0	1	17	1	0	0	0	1
2	0	0	0	1	0	18	1	0	0	1	0
3	0	0	0	1	1	19	1	0	0	1	1
4	0	0	1	0	0	20	1	0	1	0	0
5	0	0	1	0	1	21	1	0	1	0	1
6	0	0	1	1	0	22	1	0	1	1	0
7	0	0	1	1	1	23	1	0	1	1	1

8	0	1	0	0	0	24	1	1	0	0	0
9	0	1	0	0	1	25	1	1	0	0	1
10	0	1	0	1	0	26	1	1	0	1	0
11	0	1	0	1	1	27	1	1	0	1	1
12	0	1	1	0	0	28	1	1	1	0	0
13	0	1	1	0	1	29	1	1	1	0	1
14	0	1	1	1	0	30	1	1	1	1	0
15	0	1	1	1	1						

Table 1-1 GPIB address

1.2.2 Other DIP Switches

The remaining bits on the DIP switch, A6-A8, preset the electronic load mainframe 6332/ 6334 to the following functions:

Bit	Meaning	Preset	Description
A6	Frame LOAD ON Link	OFF	When ON is set, two frames can act as LOAD Key ON/OFF through RS232C port.
A7		OFF	It must be “OFF”.
A8	SHIELD GND	OFF	It is the selection to enable shield ground.

1.3 GPIB Capability of the Electronic Load

GPIB Capability	Response	Interface Functions
Talker/Listener	All electronic load functions except the setting for GPIB address are programmable via the GPIB. The electronic load can send and receive messages through the GPIB. Status information is sent using a serial pull.	AH1, SH1, T6, L4
Service Request	The electronic load will set the SRQ line true if there is an enabled service request condition.	SR1
Remote/Local	In local mode, the electronic load is controlled by the front panel and also executes commands sent to GPIB. The electronic load powers up in local mode and remains there until it receives a command from GPIB. Once the electronic load is in remote mode, *REMOTE* will appear on the front panel LCD. All front panel keys except LCL are disabled, and the load module display is in normal metering mode. Press **LCL** key on the front panel to return to local mode. Local can be disabled using local lockout, so only the controller or the	RL1

	power switch can return to local mode.	
Device Clear	The electronic load responds to the Device Clear (DCL) and Selected Device Clear (SDC) interface commands. These two actions cause the electronic load to clear the activity that may prevent it from receiving and executing a new command. DCL and SDC do not change any programmed settings.	DCL, SDC

1.4 RS232C in Remote Control

When you use RS232C in remote control, you have to send the remote command **CONFigure:REMote ON** first in order to let control procedure enter into remote state, and then execute other command set. When control comes to an end, you have to send out the command **CONFigure:REMote OFF** so as to let control procedure return to local mode operation.

The RS232C control commands are same as those of GPIB. When the string comes to an end for RS232C command sending, <nl> must be added. Its ASCII code is 0A hexadecimal (or 10 decimal).

2. Introduction to Programming

2.1 Basic Definition

GPIB statement includes instrument control and query commands. A command statement sends an instruction to the electronic load, and a query command to request information from the electronic load.

Simple Command
A simple command statement consists of a command or keyword usually followed by a parameter or data:

LOAD ON
or TRIG

Compound Command
When two or more keywords are connected by colons (:), it creates a compound command statement. The last keyword usually is followed by a parameter or data:

CURRent : STATic : L1 3
or CONFigure : VOLTage : RANGe H

Query Command
A simple query command consists of a keyword followed by a question mark:

MEASure : VOLTage?
MEASure : CURRent?
or CHAN?

Forms of Keywords
There are two forms for a keyword as described below.

Long-Form	The word is spelled out completely to identify its function. For instance, CURRENT, VOLTAGE, and MEASURE are long-form keywords.
Short-Form	The word contains only the first three or four letters of the long-form. For instance, CURR, VOLT, and MEAS are short-form keywords.

In keyword definitions and diagrams, the short-form part of each keyword is emphasized in UPPER CASE letters to help you remember it. However, the electronic load will accept Volt, volt, voltage, VOLTAGE, volTAGE, etc. regardless of what form you have applied. However, if the keyword is incomplete, for example, “VOL” or “curre”, it will not be recognized.

2.2 Numerical Data Formats

Chroma 6330 Electronic Load accepts the numerical data type listed in Table 2-1. Numeric data may be followed by a suffix to specify the dimension of the data. A suffix may be preceded by a multiplier. Chroma 6330 makes use of the suffixes listed in Table 2-2 and multipliers listed in Table 2-3.

Symbol	Description	Example
NR1	Digits without decimal point. The decimal point is assumed to be at the right of the least-significant digit.	123, 0123
NR2	Digits with a decimal point.	123., 12.3, 0.123, .123
NR3	Digit with a decimal point and an exponent.	1.23E+3, 1.23E-3
NRf	Flexible decimal form that includes NR1 or NR2 or NR3.	123, 12.3, 1.23E+3
NRf+	Expanded decimal form that includes NRf and MIN, MAX. MIN and MAX are the minimum and maximum limit values for the parameter.	123, 12.3, 1.23E+3, MIN, MAX

Table 2-1 Numerical Data Type

Mode	Class	Preferred Suffix	Secondary Suffix	Referenced Unit
CC	Current	A		Ampere
CR	Resistance	OHM		Ohm
CV	Amplitude	V		Volt
All	Time	S		Second
			MS	Millisecond
All	Slew Rate	A/μS		Amperes/micro Second

Table 2-2 Suffix Elements

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

Table 2-3 Suffix Multipliers

2.3 Character Data Formats

For command statements, the <NRf+> data format permits entry of required characters. For query statements, character strings may be returned in either of the forms shown in the following table. It depends on the length of the returned string.

Symbol	Character Form
crd	Character Response Data. They permit the return up to 12 characters.
aard	Arbitrary ASCII Response Data. They permit the return of undelimited 7-bit ASCII. This data type is an implied message terminator (refer to *Separators and Terminators*).

2.4 Separators and Terminators

In addition to keywords and parameters, GPIB program statements require the following:

Data Separators:
Data must be separated from the previous command keyword by a space. This is shown in examples as a space (CURR 3) and on diagrams by the letters *SP* inside a circle.

Keyword Separators:
Keywords (or headers) are separated by a colon (:), a semicolon (;), or both. For example:

- **LOAD:SHOR ON**
- **MEAS:CURR?;VOLT?**
- **CURR:STAT:L1 3;:VOLT:L1 5**

Program Line Separators:
A terminator informs GPIB that it has reached the end of a statement. Normally, this is sent automatically by your GPIB programming statements.

The termination also occurs with other terminator codes, such as EOI. In this guide, the terminator is assumed at the end of each example line of code. If it needs to be indicated, it is shown by the symbol <nl>, which stands for "new line" and represents the ASCII code byte 0A hexadecimal (or 10 decimal).

Traversing the Command Tree:

- The colon ":" separates keywords from each other which represents changes in branch level to the next lower one. For example:

  CONF:VOLT:ON 5
  CONF is a root-level command, *VOLT* is the first branch, and *ON* is the second branch. Each ":" moves down command interpretation to the next branch.

- The semicolon ";" allows you to combine command statements into one line. It returns the command interpretation to the previous colon.

  For example: Combine the following two command statements:
  RES:RISE 100 <nl> and
  RES:L1 400 <nl>
  which can be formed into one command line as follows:
  RES:RISE 100;L1 400 <nl>

- To return to the root-level form you can
  1. Enter a new line character. This is symbolized as "<nl>" and can be linefeed "LF" or/and end of line "EOL". Or else,
  2. Enter a semicolon followed by a colon ";:".

  Please refer to the following figure.

  1. (root):VOLT:L1: 30<nl>
     Starting a New Line to return to the Root.

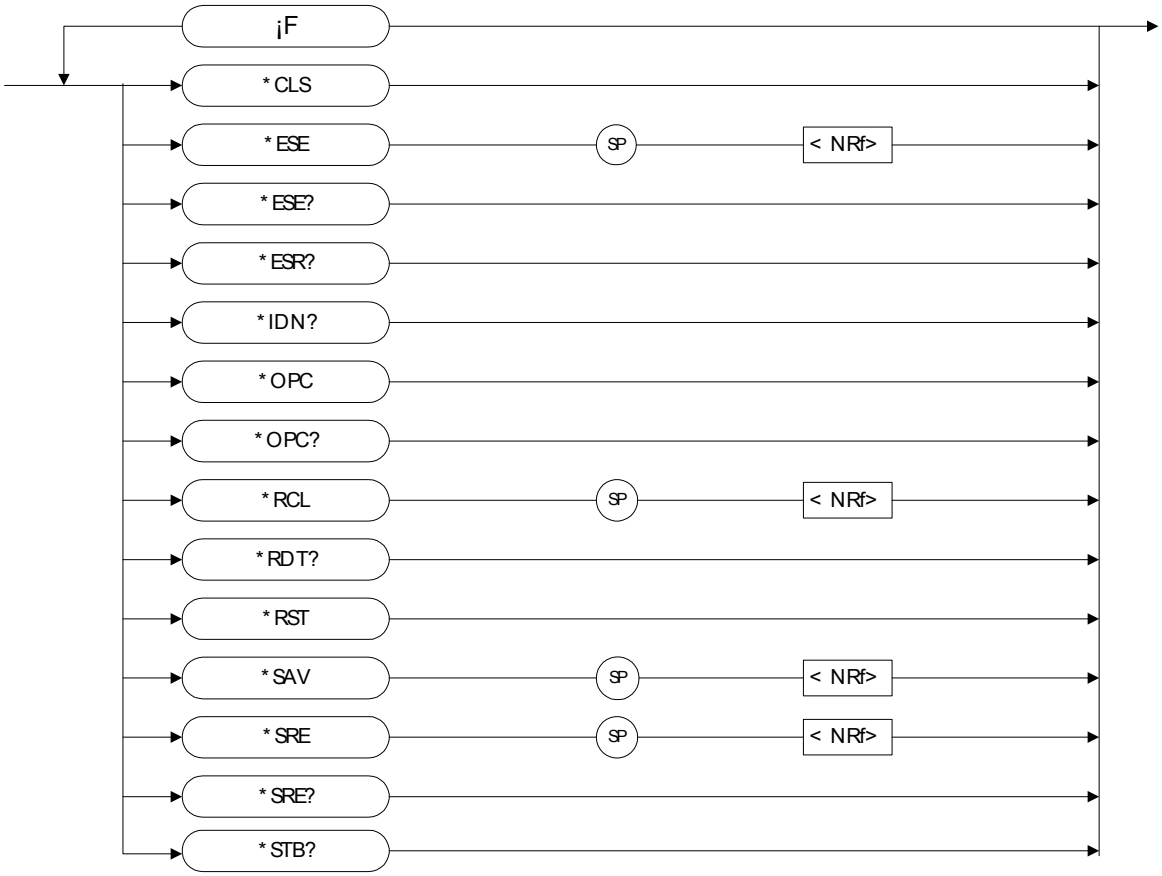
  2. (root):SPEC:VOLT:H 30;
     :L 5;:
     (root):RES:L1 400;
     :RISE 1000;:

3. Language Dictionary

Commands for operating the 6330 Electronic Load remotely are grouped into subsystems. Each command that belongs to the same subsystem is arranged in alphabetic order. A syntax chart of the subsystem that contains the commands in the same group is included. Subsystems are ordered alphabetically according to their names in the following sections.

3.1 Common Commands

The common commands defined by IEEE488.2 standard are generic commands and queries. The first part of the language dictionary covers the commands. Each of them has a leading "*".



CLS Clear Status Command

Type: Device Status

Description: The *CLS command executes the following actions:

1. Clear these registers
   - <1> Channel Status Event registers for all channels
   - <2> Channel Summary Event register
   - <3> Questionable Status Event register
   - <4> Standard Event Status Event register
   - <5> Operation Status Event register
2. Clear the Error Queue
3. If "Clear Status Command" immediately follows a program message terminator (<nl>), the "Output Queue" and the MAV bit are also cleared.

Syntax: *CLS

Parameters: nil

ESE Standard Event Status Enable Command/Query

Type: Device Status

Description: This command sets the condition of the Standard Event Status Enable register to determine which event (see *ESR?) is allowed to set the ESB (Event Summary Bit) for the Status Byte register. A "1" in the bit position enables the corresponding event. All of the events that enabled by Standard Event Status register are logically ORed to cause the Status Byte register ESB (bit 5) to be set. See descriptions of these three registers in Chapter 4 *Status Reporting*.

Syntax: *ESE <NRf>

Parameters: 0 to 255

Example: *ESE 48 This command enables the CME and EXE events for the Standard Event Status register.

Query Syntax: *ESE?

Return Parameters: <NR1>

Query Example: *ESE? This query returns the current setting for "Standard Event Status Enable".

ESR? Standard Event Status Register Query

Type: Device Status

Description: This query reads the Standard Event Status register. Reading the register clears it. See detailed explanation of this register in Chapter 4 *Status Reporting*.

Standard Event Status Event Register

Bit Position	7	6	5	4	3	2	1	0
Condition	0	0	CME	EXE	DDE	QYE	0	0
Bit Weight	128	64	32	16	8	4	2	1

Query Syntax: *ESR?

Return Parameters: <NR1>

Query Example: *ESR? Return the Standard Event Status register readings.

Return Example: 48

IDN? Identification Query

Type: System Interface
Description: This query requests the Electronic Frame (6334) to identify itself.
Query Syntax *IDN?
Return Parameters: <aard>
Query Example: *IDN?

String	Information
CHROMA	Manufacture
6334	Model
0	Always return zero
01.00	Revision level of the primary interference firmware
0	Customer's version

Return Example: CHROMA 6334,0,01.00,0

OPC Operation Complete Command

Type: Device Status
Description: This command causes the interface to set the OPC bit (bit 0) of the Standard Event Status register when the Electronic Frame (6334) has completed all pending operations.
Syntax: *OPC
Parameters: nil

OPC? Operation Complete Query

Type: Device Status
Description: This query returns an ASCII "1" when all pending operations are completed.
Query Syntax: *OPC?
Return Parameters: <NR1>
Query Example: 1

RCL Recall Instrument State Command

Type: Device Status
Description: This command restores the electronic load to a state that was previously stored in memory with the *SAV command to the specified location (see *SAV).
Syntax: *RCL <NRf>
Parameters: 1 to 101
Example: *RCL 50

RDT? Resource Description Transfer Query

Type: System Interface
Description: This command returns the types of Electronic Frame (6334). If channel does not exist, it returns 0. If channel exists, it returns the types like 63303, 63302, 63307R, 63307L...
Query Syntax: *RDT?
Return Parameters: <aard>
Query Example: 63307L, 63307R, 63303, 0, 63302, 63302, 0, 0.

RST Reset Command

Type: Device State
Description: This command forces an ABORt, *CLS, LOAD=PROT=CLE command.
Syntax: *RST
Parameters: nil

SAV Save Command

Type: Device Status
Description: This command stores the present state of the single electronic load and all channel states of multiple loads in a specified memory location.
Syntax: *SAV <NRf>
Parameters: 1 to 100
Example: *SAV 50

SRE Service Request Enable Command/Query

Type: Device Status
Description: This command sets the condition of the Service Request Enable register to determine which event of the Status Byte register (see *STB) is allowed to set the MSS (Master Status Summary) bit. A "1" in the bit position is logically ORed to cause the Status Byte register Bit 6 (the Master Summary Status Bit) to be set. See details regarding the Status Byte register in Chapter 4 *Status Reporting*.
Syntax *SRE <NRf>
Parameters: 0 to 255
Example: *SRE 20 Enable the CSUM and MAV bit for Service Request.
Query Syntax: *SRE?
Return Parameters: <NR1>
Query Example: *SRE? Return current setting for "Service Request Enable".

STB? Read Status Byte Query

Type: Device Status
Description: This query reads the Status Byte register. Note that the MSS (Master Summary Status) bit instead of RQS bit is returned in Bit 6. This bit indicates if the electronic load has at least one reason for requesting service. *STB? does not clear the Status Byte register, which is cleared only when subsequent action has cleared all its set bits. Refer to Chapter 4 *Status Reporting* for more information about this register.

Status Byte Register

Bit Position	7	6	5	4	3	2	1	0
Condition	0	MSS	ESB	MAV	QUES	CSUM	0	0
Bit Weight	128	64	32	16	8	4	2	1

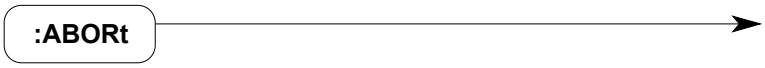
Query Syntax: *STB?

Return Parameters: <NR1>
Query Example: *STB? Return the contents of "Status Byte".
Return Example: 20

3.2 Specific Commands

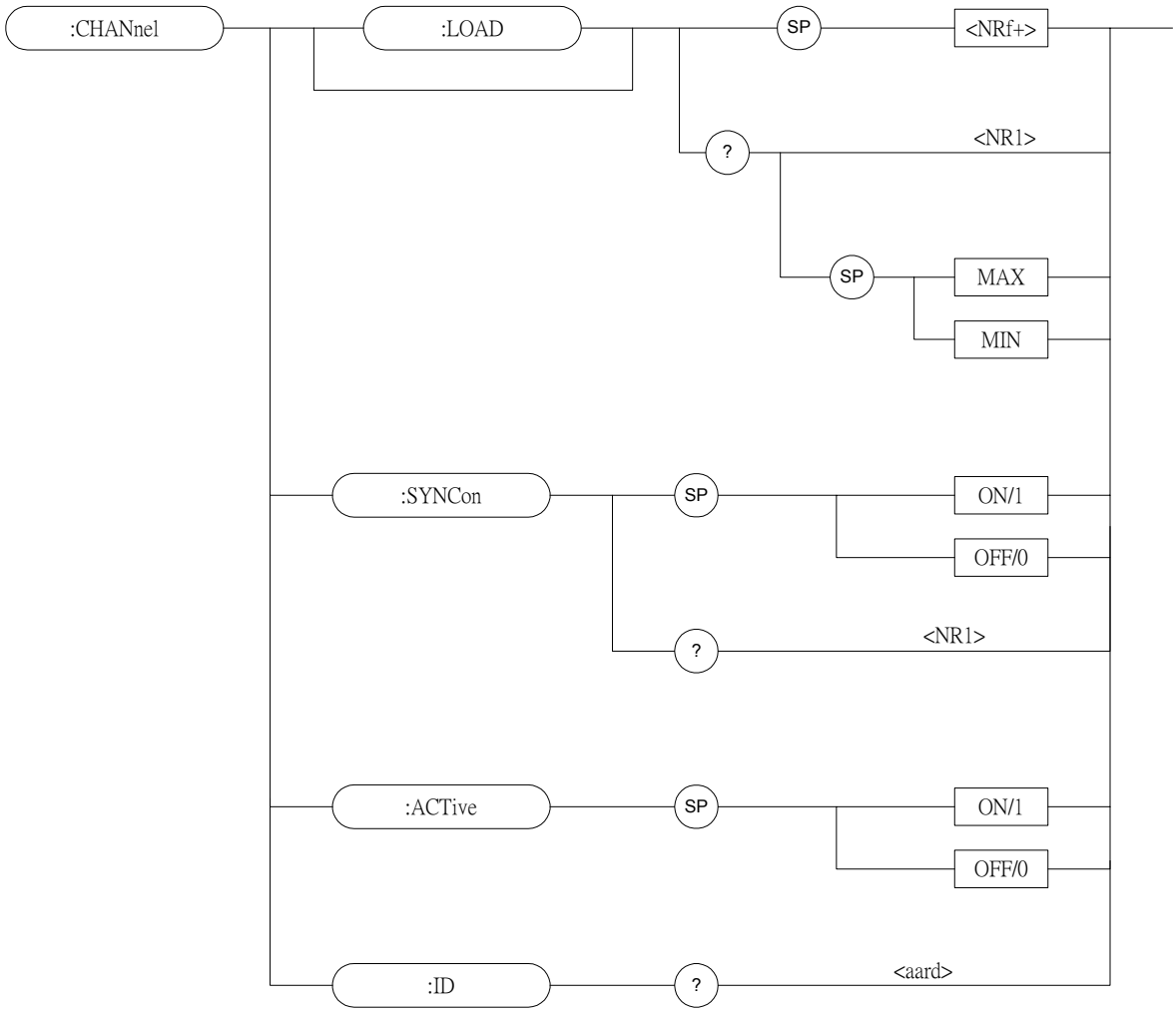
The 6330 series products are equipped with the following specific GPIB commands.

3.2.1 ABORT Subsystem



ABORt
Type: All Channel
Description: Set all electronic loads as "OFF".
Syntax: ABORt

3.2.2 CHANNEL Subsystem



CHANnel:[LOAD]

Type:	Channel Specific	
Description:	Select a channel of which the coming channel-specific command will be received and executed.	
Syntax:	CHANnel <NRf+>	
Parameters:	1 ~ 8	
Example:	CHAN 1	Set the channel to "1".
	CHAN MAX	Set the channel to "8".
	CHAN MIN	Set the channel to "1".
Query Syntax:	CHAN?	
	CHAN? MAX	
	CHAN? MIN	
Return Parameters:	<NR1>	
Query Example:	CHAN?	Return current specified channel.
Return Example:	1	

CHANnel:ACTive

Type: Channel Specific
Description: Enable or disable the load module.
Syntax: CHANnel: ACTive ON. Enable the load module. The front panel displays the measurement of voltage and current. CHANnel: ACTive OFF. Disable the load module. LCD on the front panel appears OFF.
Parameter: ON/1, OFF/0
Example: CHAN: ACT ON

CHANnel:SYNCon

Type: Channel Specific
Description: Set the load module to receive synchronized command action to RUN ABORT or not.
Syntax: CHANnel: SYNCon ON
CHANnel: SYNCon OFF
Parameters: ON/1, OFF/0
Example: CHAN: SYNC ON. Set the load module to receive synchronized command action.
CHAN: SYNC OFF. Set the load module not to receive synchronized command action.
Query Syntax: CHAN: SYNC?
Return Parameters: <NR1>
Query Example: CHAN: SYNC? Return to the load module and make it receive synchronized command status.
Return Example: 0 The load module does not receive synchronized command status.
1 The load module receives synchronized command status.

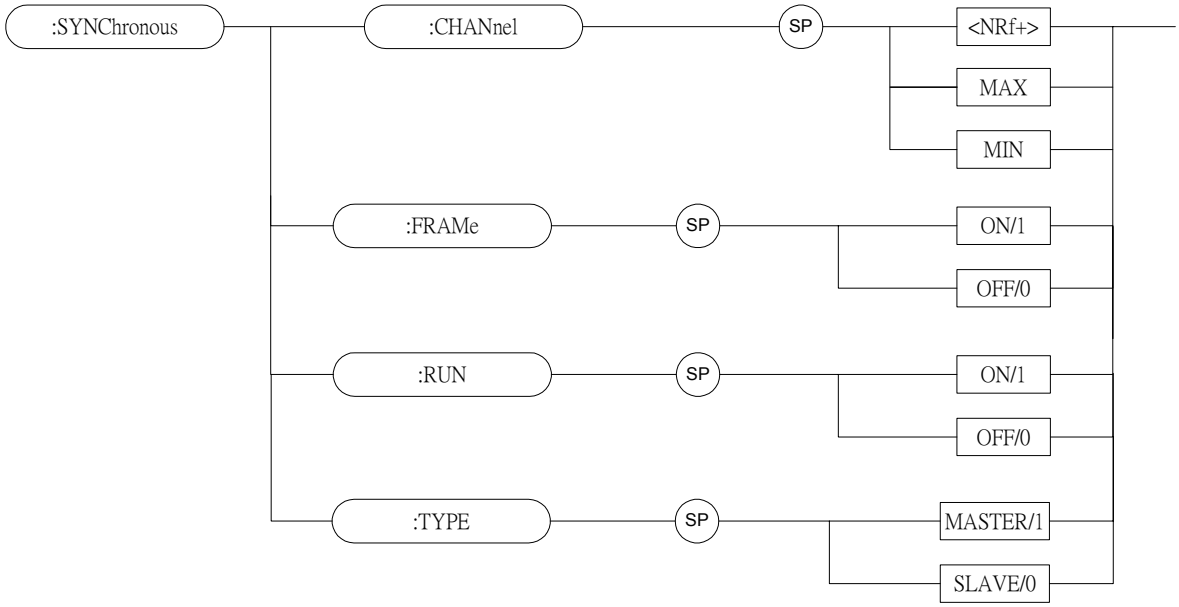
CHAN:ID?

Type: Channel-Specific
Description: This query requests the module to identify itself.
Query Syntax: ID?
Return Parameters: <aard>
Query Example: ID?

String	Information
CHROMA	Manufacturer
6330X	Model
0	Always return zero
xx.xx	Revision of the primary interface firmware
0	Customer's Version

Return Example: CHROMA,63302,0,01.00,0

3.2.3 SYNCHRONOUS Subsystem



SYNChronous:CHANnel

Type:	All Channels	
Description:	Set the specified channel to T1 & T2 in sync dynamic mode for parallel loading.	
Syntax:	SYNChronous:CHANnel <NRf+>	
Parameters:	1 ~ 8	
Example:	SYNC:CHAN 1	Set the specified channel to "1".
	SYNC:CHAN MAX	Set the specified channel to "8".
	SYNC:CHAN MIN	Set the specified channel to "1".

SYNChronous:FRAMe

Type:	All Channels	
Description:	Set the mainframe if to sync. in parallel run. The 6330 series have a master/slave paralleling control mode that allows synchronous load control in static and dynamic loading mode.	
Syntax:	SYNChronous: FRAMe ON.	Enable the mainframe to sync. in parallel run.
	SYNChronous: FRAMe OFF.	Disable the mainframe to sync. in parallel run.

Parameter: ON/1, OFF/0
Example: SYNC: FRAM ON

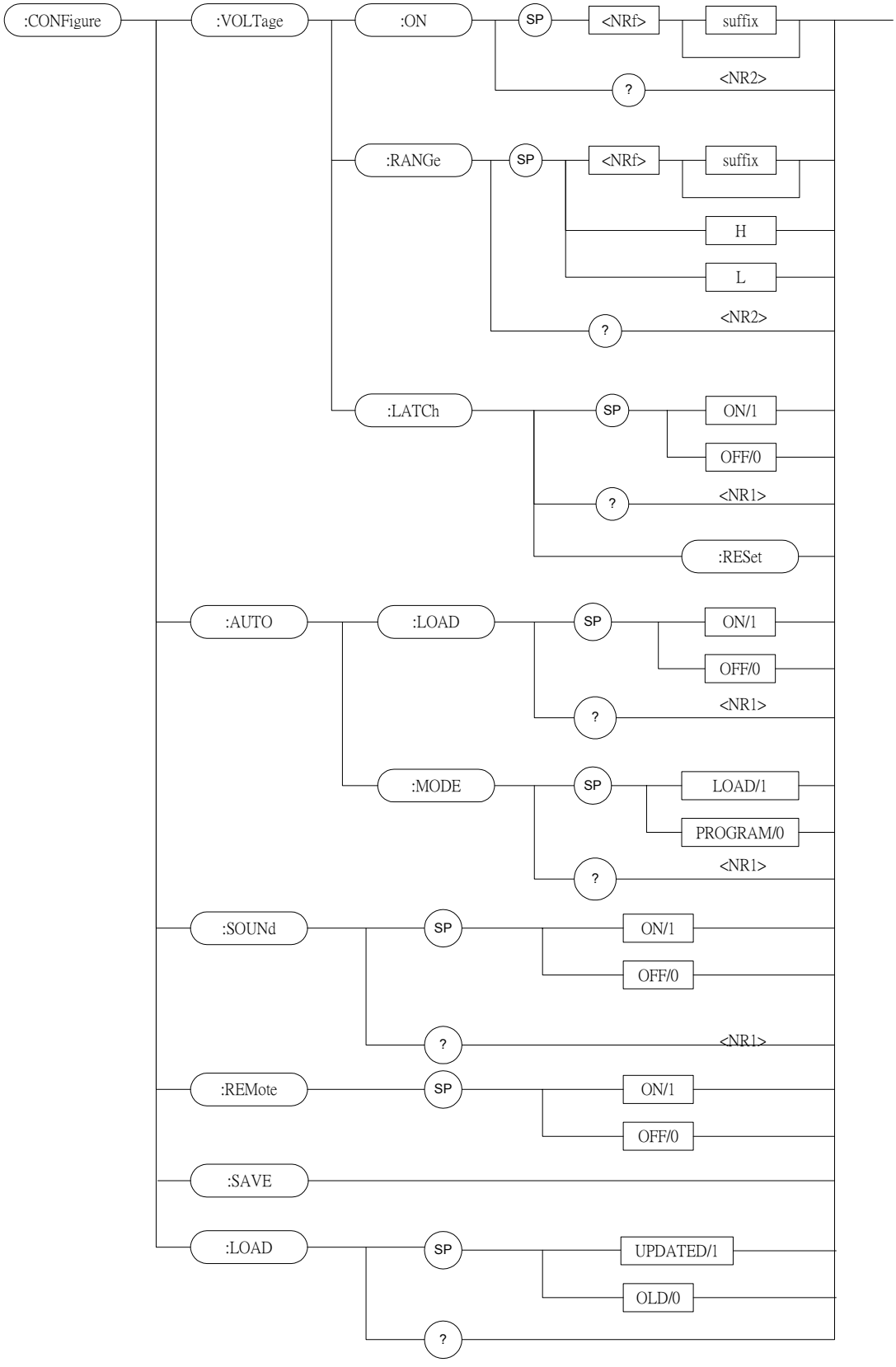
SYNChronous:RUN

Type: All Channels
Description: Set all electronic loads to "ON" in sync. parallel run.
Syntax: SYNChronous: RUN ON
SYNChronous: RUN OFF
Parameters: ON/1, OFF/0
Example: SYNC: RUN ON Set the load to "ON" on sync. parallel.
SYNC: RUN OFF Set the load to "OFF" on sync. parallel.

SYNChronous:TYPE

Type: All Channels
Description: Set the specified mainframe to master or slave for sync. in parallel run.
Syntax: SYNChronous: TYPE MASTER
SYNChronous: TYPE SLAVE
Parameters: MASTER /1, SLAVE /0
Example: SYNC: TYPE MASTER Set the mainframe to master for sync. in parallel run.
SYNC: TYPE SLAVE Set the mainframe to slave for sync. in parallel run.

3.2.4 CONFIGURE Subsystem



CONFigure:VOLTage:ON

Type:	Channel-Specific	
Description:	Set the voltage of sink current on.	
Syntax:	CONFigure:VOLTage:ON <NRf> [suffix]	
Parameters:	For valid voltage range, refer to the respective specification.	
Example:	CONF:VOLT: ON 1	Set Von=1V.
	CONF:VOLT: ON 300mV	Set Von=300mV.
Query Syntax:	CONFigure:VOLTage:ON?	
Return Parameters:	<NR2> [Unit=Voltage]	
Query Example:	CONF: VOLT: ON?	Return the setting Von value.
Return Example:	3.5	

CONFigure:VOLTage:RANGe

Type:	Channel-Specific	
Description:	Set the voltage measurement range in CC mode.	
Syntax:	CONFigure:VOLTage:RANGEe <NRf> [suffix]	
Parameters:	Value ranges depend on Load Module. For details, refer to the specification.	
Example:	CONF:VOLT:RANG 16	Set full-range to Low, for example, in 63303.
	CONF:VOLT:RANG 80V	Set full-range to High, for example, in 63303.
	CONF:VOLT:RANG H	Set full-range to High.
	CONF:VOLT:RANG L	Set full-range to Low.
Query Syntax:	CONFigure:VOLTage:RANGe?	
Return Parameters:	<NR2> [Unit = Voltage]	
Query Example:	CONF:VOLT:RANG?	Return to Voltage range.
Return Example:	16	

CONFigure:VOLTage:LATCh

Type:	Channel-Specific	
Description:	Set the action type of Von.	
Syntax:	CONFigure:VOLTage:LATCh ON CONFigure:VOLTage:LATCh OFF	
Parameters:	ON/1, OFF/0	
Example:	CONF:VOLT:LATC ON	Set the action type of Von to Latch.
	CONF:VOLT:LATC OFF	Set the action type of Von to Non Latch (For detailed action, refer to the operator's manual).
Query Syntax:	CONFigure:VOLTage:LATCh?	
Return Parameters:	<NR1>	
Query Example:	CONF:VOLT:LATC?	
Return Example:	0 (non latch), 1 (latch)	Return the action type of Von.

CONFigure:VOLTage:LATCh:RESet

Type: Channel-Specific
Description: Reset the Von signal.
Syntax: CONFigure:VOLTage:LATCh:RESet
Example: CONF:VOLT:LATC:RES Reset the Von signal.

CONFigure:AUTO:LOAD

Type: All Channels
Description: Set if the load module to perform Auto Load On during power-on.
Syntax CONFigure:AUTO:LOAD ON
CONFigure:AUTO:LOAD OFF
Parameters: ON/1, OFF/0
Example: CONF:AUTO:LOAD ON Start Auto Load On during power-on.
CONF:AUTO:LOAD OFF Close Auto Load On during power-on.
Query Syntax: CONFigure:AUTO:LOAD?
Return Parameters: <NR1>
Query Example: CONF:AUTO:LOAD?
Return Example: 0 or 1 Return the status of Auto Load On

CONFigure:AUTO:MODE

Type: All Channel
Description: Set type of Auto Load On as LOAD ON or PROGRAM RUN.
Syntax: CONFigure:AUTO:MODE LOAD
CONFigure:AUTO:MODE PROGRAM
Parameters: LOAD/1, PROGRAM/0
Example: CONF:AUTO:MODE LOAD Set Auto Load On to general LOAD ON.
CONF:AUTO:MODE PROGRAM Set Auto Load On to PROGRAM RUN.
Query Syntax: CONFigure:AUTO:MODE?
Return Parameters: <NR1>
Query Example: CONF:AUTO:MODE? Return the execution
Return Example: 0 or 1 type of Auto Load On.

CONFigure:SOUND

Type: Channel-Specific
Description: Set the buffer sound of the load module to ON/OFF.
Syntax: CONFigure:SOUND ON
CONFigure:SOUND OFF
Parameters: ON/1, OFF/0
Example: CONF:SOUND ON
CONF:SOUND OFF
Query Syntax: CONFigure:SOUND?
Return Parameters: <NR1>
Query Example: CONF:SOUND? Return the buzzer sound control status of the load module.
Return Example: 0 or 1

CONFigure:REMote

Type: All Channel
Description: Set the status of remote control (only available in RS232C).
Syntax: CONFigure:REMote ON
CONFigure:REMote OFF
Parameters: ON/1, OFF/0
Example: CONF:REM ON Set to remote control.

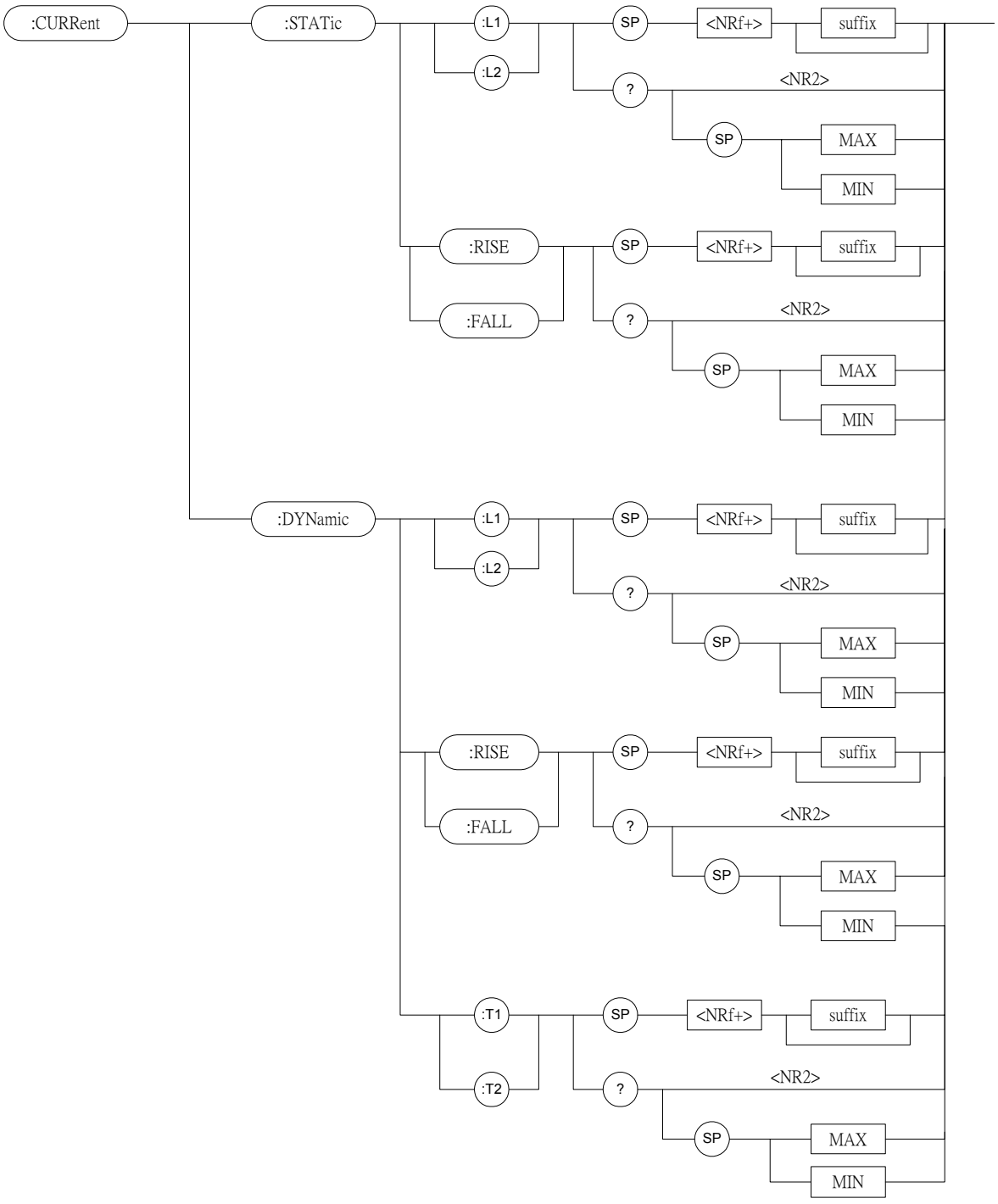
CONFigure:SAVE

Type: All Channels
Description: Store the data of CONFigure into EEPROM.
Syntax: CONFigure:SAVE
Parameters: none
Example: CONF:SAVE

CONFigure:LOAD

Type: All Channels
Description: The value setting for load module changed by the rotary knob (UPDATED/1) as LOADON, or the original set value (OLD/0).
Syntax: CONFigure:LOAD UPDATED
CONFigure:LOAD OLD
Parameters: UPDATED/1, OLD/0
Example: CONF:LOAD UPDATED Set the value of LOADON to be changed by the rotary knob.
CONF:LOAD OLD Set the value of LOADON to be the original set value.
Query Syntax: CONFigure:LOAD?
Return Parameters: <NR1>
Query Example: CONF:LOAD?
Return Example: 1 (UPDATED) or 0 (OLD)

3.2.5 CURRENT Subsystem



CURRent:STATic:L1/L2
C:L1/L2

Type:	Channel-Specific	
Description:	Set Static Load Current for constant current mode.	
Syntax:	CURRent:STATic:L1 <NRf+>[suffix]	
	CURRent:STATic:L2 <NRf+>[suffix]	
Parameters:	Refer to respective specification for valid value range.	
Example:	CURR:STAT:L1 20	Set Constant Current = 20A for Static Load L1.
	CURR:STAT:L2 10	Set Constant Current = 10A for Static Load L2.
	CURR:STAT:L1 MAX	Set Constant Current = maximum value for Static Load L1.
	CURR:STAT:L2 MIN	Set Constant Current = minimum value for Static Load L2.
Query Syntax:	CURRent:STATic:L1?	
	CURRent:STATic:L2?	
	CURRent:STATic:L1? MAX	
	CURRent:STATic:L2? MIN	
Return Parameters:	<NR2> [Unit=Ampere]	
Query Example:	CURR:STAT:L1?	Return the set current value to Static Load L1.
Return Example:	3.12	

CURRent:STATic:RISE/FALL
C:RISE/FALL

Type:	Channel-Specific	
Description:	Set the current slew rate for constant current static mode.	
Syntax:	CURRent:STATic:RISE<NRf+> [suffix]	
	CURRent:STATic:FALL <NRf+> [suffix]	
Parameters:	Refer to respective specification for valid value range	
Example:	CURR:STAT:RISE 2.5	Set rise slew rate as 2.5A/μS of static load.
	CURR:STAT:FALL 1A/μS	Set fall slew rate as 1A/μS of static load.
Query Syntax:	CURRent:STATic:RISE?	
	CURRent:STATic:FALL?	
	CURRent:STATic:RISE? MAX	
	CURRent:STATic:FALL? MIN	
Return Parameters:	<NR2> [Unit=A/μS]	
Query Example:	CURR:STAT:RISE?	Return the rise slew rate of static load.
Return Example:	2.5	

CURRent:DYNamic:L1/L2

Type:	Channel-Specific
Description:	Set the Dynamic Load Current during constant current mode.
Syntax:	CURRent:DYNamic:L1 <NRf+>[suffix]
	CURRent:DYNamic:L2 <NRf+>[suffix]

Parameters: Refer to respective specification for valid value range.
Example: CURR:DYN:L1 20 Set the dynamic load parameter L1 = 20A.
CURR:DYN:L2 10 Set the dynamic load parameter L2 = 10A.
CURR:DYN:L1 MAX Set the dynamic load parameter L1 = maximum value.
CURR:DYN:L2 MIN Set the dynamic load parameter L2 = minimum value.
Query Syntax: CURRent:DYNamic:L1?
CURRent:DYNamic:L2?
CURRent:DYNamic:L1? MAX
CURRent:DYNamic:L2? MIN
Return Parameters: <NR2> [Unit=Ampere]
Query Example: CURR:DYN:L1? Return the setting current in dynamic load L1.
Return Example: 35.6

CURRent:DYNamic:RISE/FALL

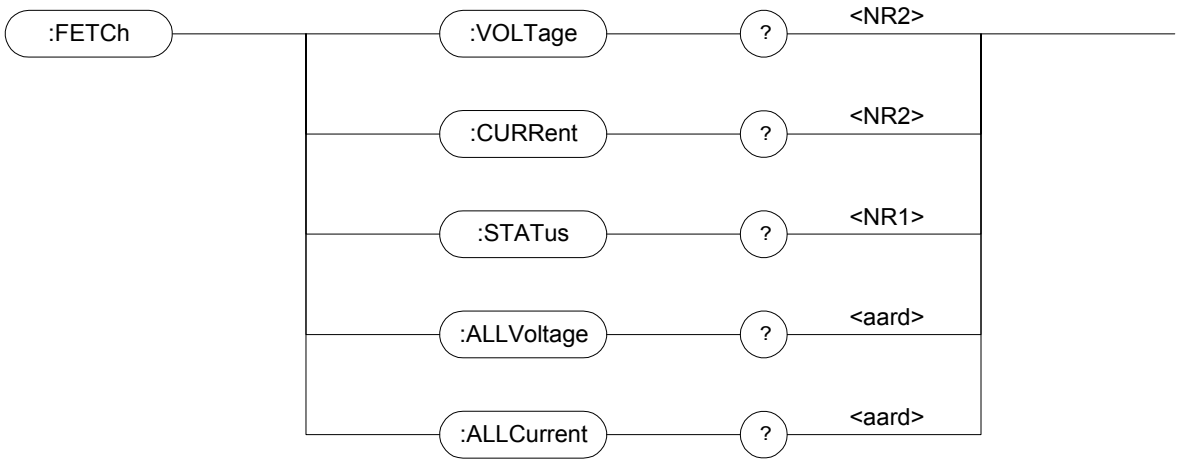
Type: Channel-Specific
Description: Set the current slew rate for constant current dynamic mode.
Syntax: CURRent:DYNamic:RISE <NRf+> [suffix]
CURRent:DYNamic:FALL <NRf+> [suffix]
Parameters: Refer to respective specification for valid value range.
Example: CURR:DYN:RISE 2.5 Set rise slew rate to 2.5A/μS.
CURR:DYN:FALL 1A/μS Set fall slew rate to 1A/μS.
CURR:DYN:RISE MAX Set rise slew rate to the maximum value of dynamic load.
CURR:DYN:FALL MIN Set fall slew rate to the minimum value of dynamic load.
Query Syntax: CURRent:DYNamic:RISE?
CURRent:DYNamic:FALL?
CURRent:DYNamic:RISE? MAX
CURRent:DYNamic:FALL? MIN
Return Parameters: <NR2> [Unit=A/μS]
Query Example: CURR:DYN:RISE? Return the rise slew rate of dynamic load.
Return Example: 2.5

CURRent:DYNamic:T1/T2

Type: Channel-Specific
Description: Set duration parameter T1 or T2 for dynamic load.
Syntax: CURRent:DYNamic:T1 <NRf+> [suffix]
CURRent:DYNamic:T2 <NRf+> [suffix]
Parameters: Refer to respective specification for valid value range.
Example: CURR:DYN:T1 10mS Set the dynamic duration T1 = 10mS.
CURR:DYN:T2 2S Set the dynamic duration

		T2 = 2S.
	CURR:DYN:T1 MAX	Set the dynamic duration T1 as maximum value.
	CURR:DYN:T2 MIN	Set the dynamic duration T2 as minimum value.
Query Syntax:	CURRent:DYNamic:T1?	
	CURRent:DYNamic:T2?	
	CURRent:DYNamic:T1? MAX	
	CURRent:DYNamic:T2? MIN	
Return Parameters:	<NR2> [Unit=Sec]	
Query Example:	CURR:DYN:T1?	Return the dynamic duration parameter T1.
Return Example:	0.15	

3.2.6 FETCH Subsystem



FETCh:VOLTage?

Type: Channel-Specific
Description: Return the voltage measured at electronic load input.
Query Syntax: FETCh:VOLTage?
Return Parameters: <NR2> [Unit=Voltage]
Query Example: FETC:VOLT?
Return Example: 8.12

FETCh:CURRent?

Type: Channel-Specific
Description: Return the current measured at electronic load input.
Query Syntax: FETCh:CURRent?
Return Parameters: <NR2> [Unit=Ampere]
Query Example: FETC:CURR?
Return Example: 3.15

FETCh:STATus?

Type: Channel-Specific
Description: Return real time status of the load module.
Query Syntax: FETCh:STATus?
Return Parameters: <NR1>

FETCh:ALLVoltage?

Type: All Channel
Description: Return the voltage measured at the input of the all load channels. The return value is 0 when the channel does not exist.
Query Syntax: FETCh:ALLVoltage?
Return Parameters: <aard> [Unit=Voltage]
Query Example: FETC:ALLV?
Return Example: 1.2, 2, 0, 0, 10.2, 0, 0, 0

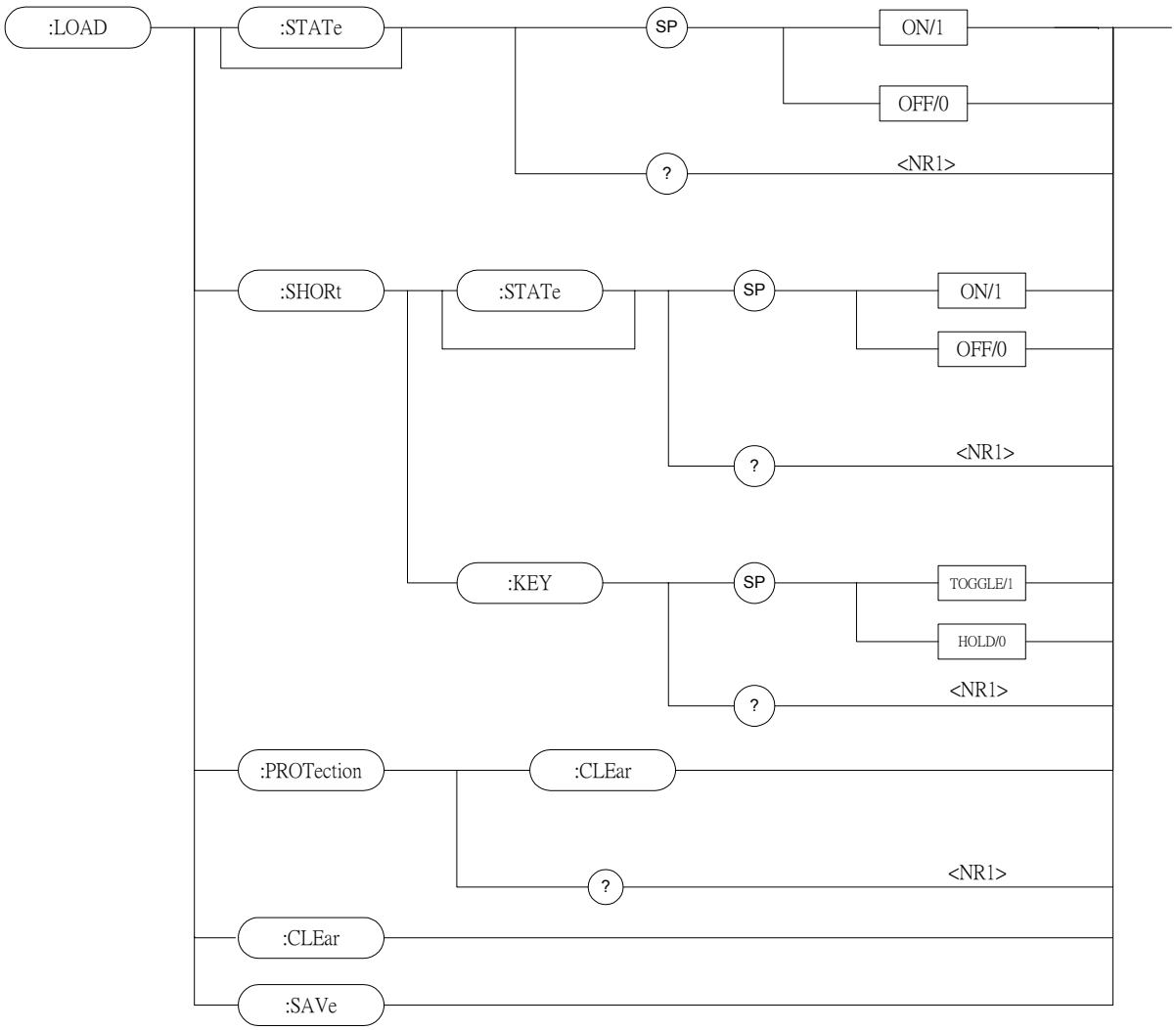
FETCh:ALLCurrent?

Type: Channel-Independent
Description: Return the current measured at the input of the all load modules. The return value is 0 when the channel does not exist.
Query Syntax: FETCh:ALLCurrent?
Return Parameters: <aard> [Unit=Ampere]
Query Example: FETC:ALLC?
Return Example: 0, 0, 0, 0, 5.12, 0, 12, 0

Bit Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Condition												OT	RV	OP	OV	OC
Bit Weight												16	8	4	2	1

Query Example: FETC:STAT? Read back the present status of load module.
Return Example: 4

3.2.7 LOAD Subsystem



LOAD:[STATe]

Type: Channel-Specific
Description: The LOAD command makes the electronic load active/on or inactive/off.
Syntax: LOAD:[STATe] ON
LOAD:[STATe] OFF
Parameters: ON/1, OFF/0
Example: LOAD ON Activate the electronic load.
LOAD OFF Inactivate the electronic load.
Query Syntax: LOAD:[STATe]?
Return Parameters: <NR1>
Query Example: LOAD? Return if the electronic load is active.
Return Example: 1

LOAD:SHORt:[STATe]

Type: Channel-Specific
Description: Activate or inactivate short-circuited simulation.
Syntax: LOAD:SHORt:[STATe]
Example: LOAD:SHOR ON Activate short-circuited simulation.
LOAD:SHOR OFF Inactivates short-circuited simulation.
Parameters: ON/1, OFF/0
Query Syntax: LOAD:SHORt:[STATe]?
Return Parameters: <NR1>
Query Example: LOAD:SHOR? Return the short-circuit simulation state.
Return Example: 1

LOAD:SHORt:KEY

Type: Channel-Specific
Description: Set the mode of short key in the electronic load.
Syntax: LOAD:SHORt:KEY TOGGLE
Parameters: TOGGLE/1, HOLD/0
Example: LOAD:SHOR:KEY TOGGLE Set the short key mode to Toggle.
LOAD:SHOR:KEY HOLD Set the short key mode to Hold.
Query Syntax: LOAD:SHORt:KEY?
Return Parameters: <NR1>
Query Example: LOAD:SHOR:KEY? Return the mode of short key in the electronic load.
Return Example: 1

LOAD:PROTection:CLEar

Type: Channel-Specific
Description: This command resets or returns the status of electronic load.
Syntax: LOAD:PROTection:CLEar
Parameters: Refer to respective specification for valid value range.
Example: LOAD:PROT:CLE
Query Syntax: LOAD:PROTection:CLEar?
Return Parameters: <NR1>

Bit Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Condition	0	0	0	0	0	0	0	0	0	0	0	OT	RV	OP	OV	OC
Bit Weight												16	8	4	2	1

Query Example: LOAD:PROT? Return the status of electronic load.
Return Example: 0

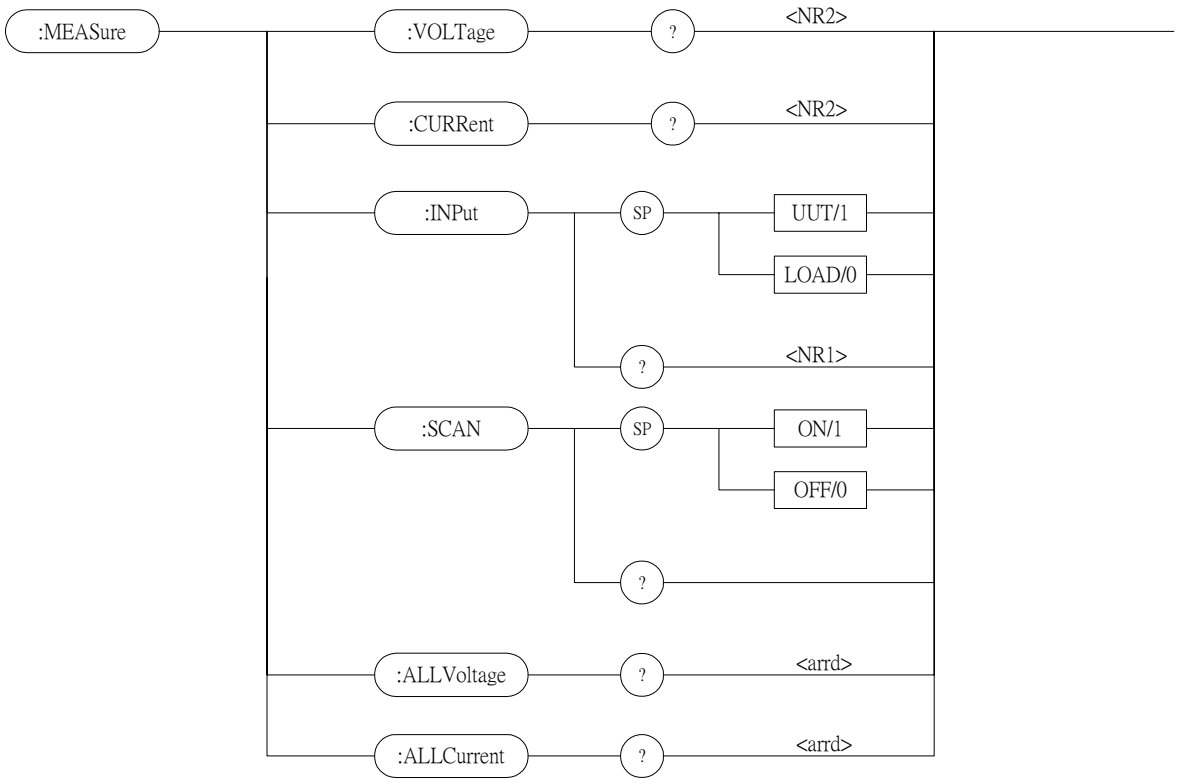
LOAD:CLEar

Type: All Channel
Description: Clear all data and return it to default.
Syntax: LOAD:CLEar
Parameters: None
Example: LOAD:CLE

LOAD:SAVe

Type:	All Channel
Description:	Save the current data as default.
Syntax:	LOAD:SAVe
Parameters:	None
Example:	LOAD:SAV

3.2.8 MEASURE Subsystem



MEASure:VOLTage?

Type: Channel-Specific
Description: Return the real time voltage measured at load module input.
Query Syntax: MEASure:VOLTage?
Return Parameters: <NR2> [Unit=Voltage]
Query Example: MEAS:VOLT?
Return Example: 8.12

MEASure:CURRent?

Type: Channel-Specific
Description: Return the real time current measured at the load module input.
Query Syntax: MEASure:CURRent?
Return Parameters: <NR2> [Unit=Ampere]
Query Example: MEAS:CURR?
Return Example: 3.15

MEASure:INPut

Type: Channel-Specific
Description: Select the electronic load input port to measure the voltage.
Syntax: MEASure:INPut?
Parameters: UUT/1, LOAD/0
Example: MEAS:INP UUT
MEAS:INP LOAD
Query Syntax: MEASure:INPut? Return the input port that has been set.

Return Parameters: <NR1>
Query Example: MEAS:INP?
Return Example: 0

MEASure:SCAN

Type:	All Channel	
Description:	Set the frame-scanning mode to load module.	
Syntax:	MEASure:SCAN ON	Enable the frame to scan the load module.
	MEASure:SCAN OFF	Disable the frame to scan the load module.
Parameters:	ON/1, OFF/0	
Example:	MEAS:SCAN ON MEAS:SCAN OFF	
Query Syntax:	MEASure:SCAN?	Return the scanning mode of the frame.
Return Parameters:	<NR1>	
Query Example:	MEAS:SCAN?	
Return Example:	1	

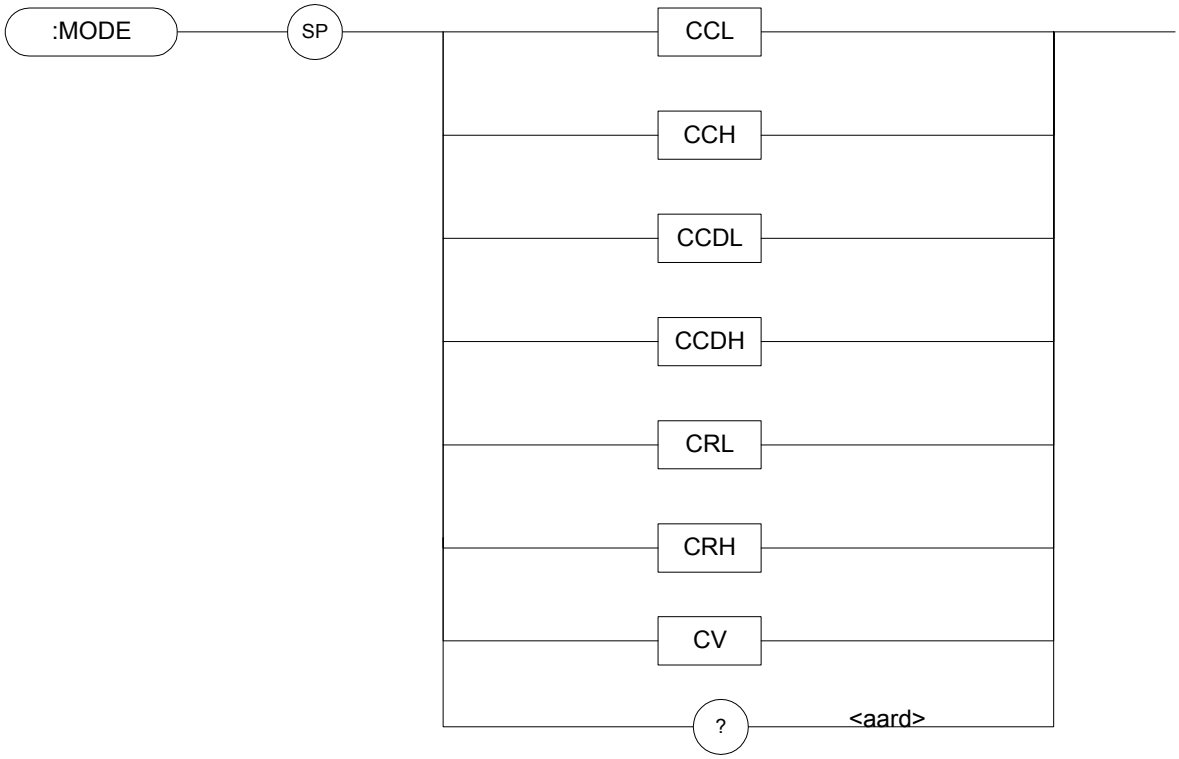
MEASure:ALLVoltage?

Type:	All Channel
Description:	Returns real time voltage measured at the input of the all load channel. The return value is 0 when the channel is not existed.
Query Syntax:	MEASure:ALLVoltage?
Return Parameters:	<aard> [Unit=Voltage]
Query Example:	MEAS:ALLV?
Return Example:	1.2, 2, 0, 0, 10.2, 0, 0, 0

MEASure:ALLCurrent?

Type:	Channel-Independent
Description:	Return the real time current measured at the input of all load modules. The return value is 0 when the channel does not exist.
Query Syntax:	MEASure:ALLCurrent?
Return Parameters:	<aard> [Unit=Ampere]
Query Example:	MEAS:ALLC?
Return Example:	0, 0, 0, 0, 5.12, 0, 12, 0

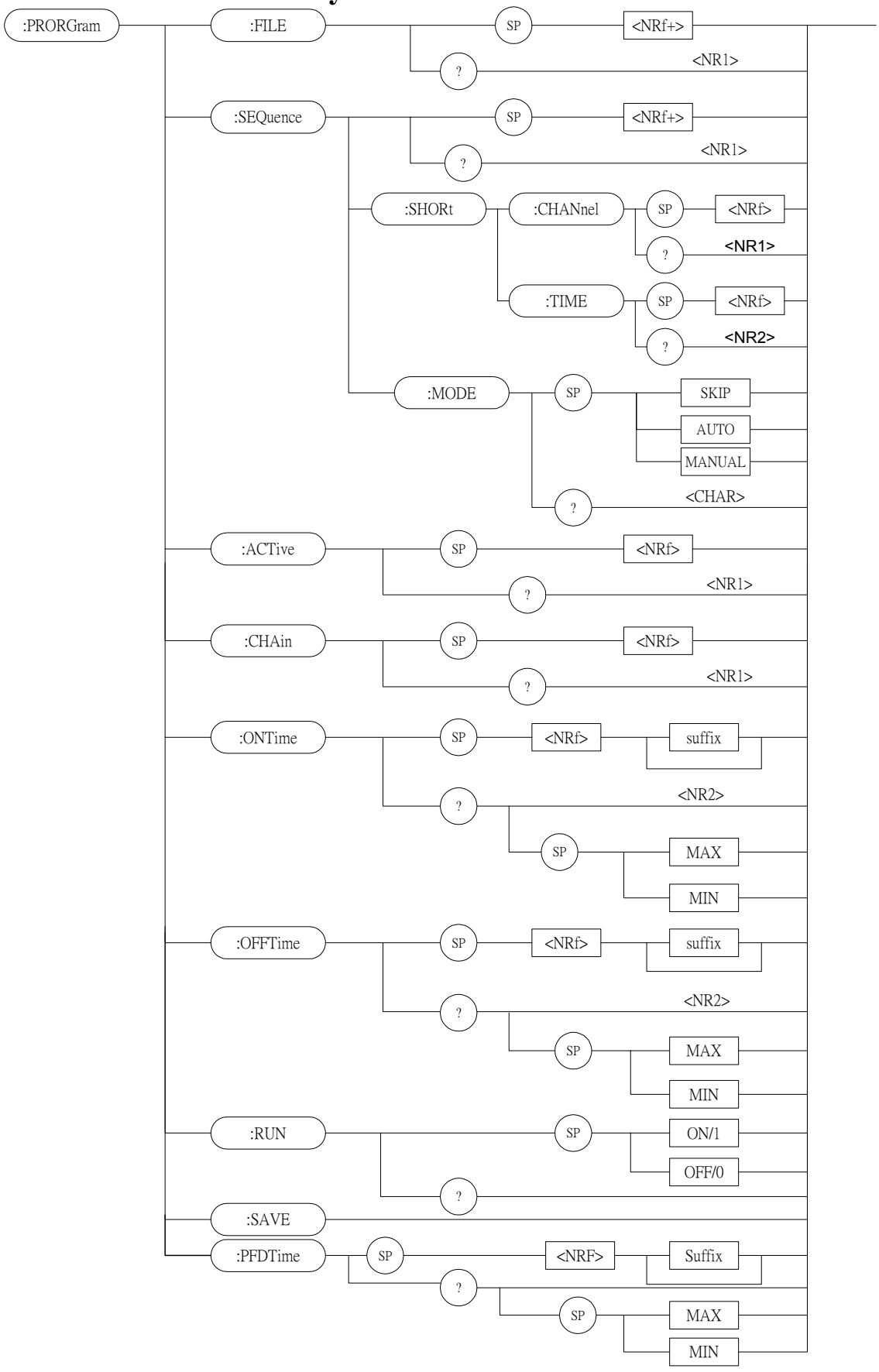
3.2.9 MODE Subsystem



MODE

Type:	Channel-Specific	
Description:	This command sets the operational mode for the electronic load.	
Syntax:	MODE CCL	Set CC mode of low range.
	MODE CCH	Set CC mode of high range.
	MODE CCDL	Set CC dynamic mode of low range.
	MODE CCDH	Set CC dynamic mode of high range.
	MODE CRL	Set CR mode of low range.
	MODE CRH	Set CR mode of high range.
	MODE CV	Set CV mode.
Parameters:	CCL, CCH, CCDL, CCDH, CRL, CRH, CV	
Example:	MODE CCL	
Query Syntax:	MODE?	Return the operational mode of the electronic load.
Return Parameters:	<aard>	
Query Example:	MODE?	
Return Example:	CCL	

3.2.10 PROGRAM Subsystem



PROGram:FILE

Type: By program file
Description: Set the program number.
Syntax: PROGram:FILE <NRf+>
Parameters: 1 to 10
Example: PROG:FILE 10
Query Syntax: PROGram:FILE? Return the active program number.
Return Parameters: <NR1>
Query Example: PROG:FILE?
Return Example: 10

PROGram:SEQuence

Type: By program file
Description: Set the sequence for program file.
Syntax: PROGram:SEQuence <NRf+>
Parameters: 1 to 10
Example: PROG:SEQ3
Query Syntax: PROGram:SEQuence?
Return Parameters: <NR1>
Query Example: PROG:SEQ?
Return Example: 3

PROGram:SEQuence:MODE

Type: By program file
Description: Set the type of sequence.
Syntax: PROGram:SEQuence:MODE SKIP
PROGram:SEQuence:MODE AUTO
PROGram:SEQuence:MODE MANUAL
Parameters: SKIP, AUTO, MANUAL
Example: PROG:SEQ:MODE SKIP
PROG:SEQ:MODE AUTO
PROG:SEQ:MODE MANUAL
Query Syntax: PROGram:SEQ:MODE?
Return Parameters: SKIP, AUTO, MANUAL
Query Example: PROG:SEQ:MODE?
Return Example: AUTO

PROGram:SEQuence:SHORt:CHANnel

Type: By program file
Description: Set the short channel for PROGRAM file SEQuence
Syntax: PROGram:SEQuence:SHORt:CHANnel <NRf>
Parameters: 0 – 255

Channel	8	7	6	5	4	3	2	1
Bit Weight	128	64	32	16	8	4	2	1

Example: PROG:SEQ:SHOR:CHAN 3
Query Syntax: PROGram:SEQuence:SHORt:CHANnel?

Return Parameter: <NR1>
Query Example: PROG:SEQ:SHOR:CHAN?
Return Example: 3

PROGram:SEQuence:SHORt:TIME

Type: By program file
Description: Set the short time for PROGRAM file SEQuence.
Syntax: PROGram:SEQuence:SHORt:TIME
Parameters: 0 - 30.0
Example: PROG:SEQ:SHOR: TIME 10
Query Syntax: PROGram:SEQuence:SHORt:TIME?
Return Parameter: <NR2>
Query Example: PROG:SEQ:SHOR:TIME?
Return Example: 10

PROGram:ACTive

Type: By program file
Description: Select the active load modules.
Syntax: PROGram:ACTive <NRf>
Parameters: 0 – 255

Channel	8	7	6	5	4	3	2	1
Bit Weight	128	64	32	16	8	4	2	1

Example: PROG:ACT 12
Query Syntax: PROGram:ACTive?
Return Parameters: <NR1>
Query Example: PROG:ACT?
Return Example: 12

PROGram:CHAin

Type: By program file
Description: Set the type of program file in serial execution.
Syntax: PROGram:CHAin <NRf>
Parameters: 0 to 10 0 does not chain.
Example: PROG:CHA 7
Query Syntax: PROGram:CHAin?
Return Parameters: <NR1>
Query Example: PROG:CHA?
Return Example: 7

PROGram:ONTime

Type: By program file
Description: Set the load on time for program file.
Syntax: PROGram:ONTime <NRf>
Parameters: Refer to respective specification for valid value range.
Example: PROG:ONT 10
PROG:ONT 100mS

Query Syntax: PROGram:ONTime?
Return Parameters: <NR2> [Unit=Sec]
Query Example: PROG:ONT?
Return Example: 10

PROGram:OFFTime

Type: By program file
Description: Set the load off time for program file.
Syntax: PROGram:OFFTime <NRf>
Parameters: Refer to respective specification for valid value range.
Example: PROG:OFFT 20
PROG:OFFT 200mS
Query Syntax: PROGram:OFFTime?
Return Parameters: <NR2> [Unit=Sec]
Query Example: PROG:OFFT?
Return Example: 0.2

PROGram:PFDTime

Type: By program file
Description: Set the pass/fail delay time of program file.
Syntax: PROGram:PFDTime <NRf>
Parameters: For valid value range refer to respective specification.
Example: PROG:PFDT 1
PROG: PFDT 200mS
Query Syntax: PROGram:PFDTime?
Return Parameters: <NR2> [Unit=Sec]
Query Example: PROG:PFDT?
Return Example: 0.2

PROGram:SAVE

Type: By program file
Description: Save the program settings.
Syntax: PROGram:SAVE
Parameters: NONE
Example: PROG:SAVE

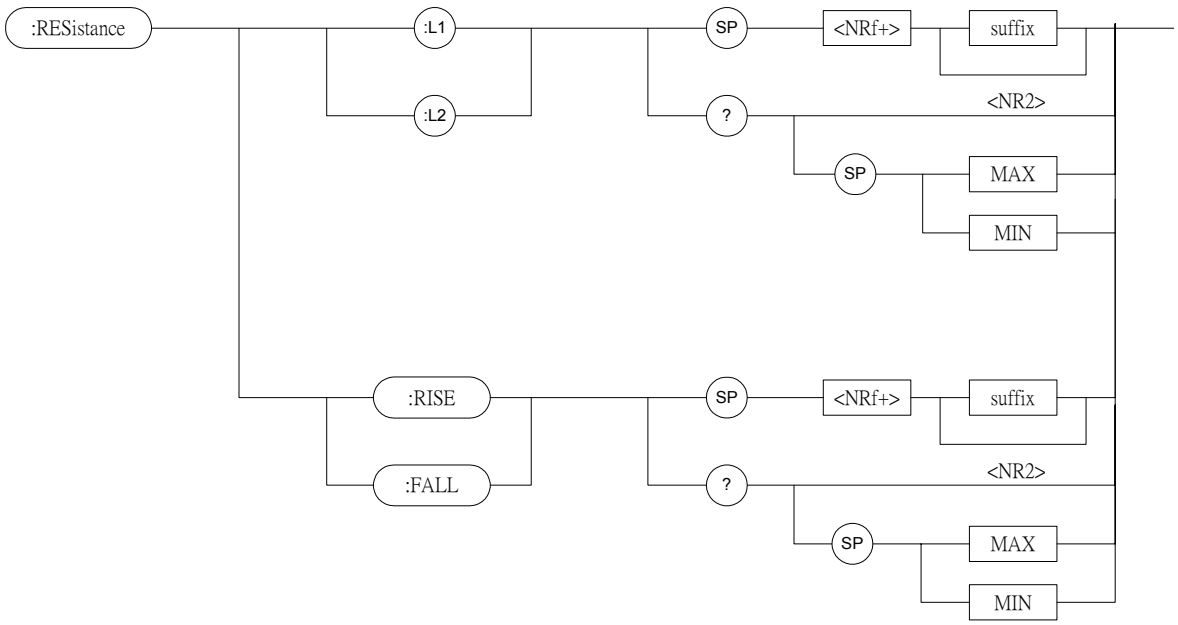
PROGram:RUN

Type: By program file
Description: Execute the program.
Syntax: PROGram:RUN ON
PROGram:RUN OFF
Parameters: ON/1, OFF/0
Example: PROG:RUN ON
Query Syntax: PROGram:RUN?
Return Parameter: <NR1>
Query Example: PROGram:RUN?
Return Example: 1

PROGram:KEY

Type:	By program file
Description:	Echo the manual key code
Syntax:	PROGram:KEY <NR1> PROGram:RUN OFF
Parameters:	0 – 9 -> K0 -> K9 10 -> Kup 11 -> Kdown
Example:	PROG:KEY 11

3.2.11 RESISTANCE Subsystem



RESistance:L1/L2

Type:	Channel-Specific	
Description:	Set static resistance level for constant resistance mode.	
Syntax:	RESistance:L1 <NRf+> [suffix]	
	RESistance:L2 <NRf+> [suffix]	
Parameters:	Refer to respective specification for valid value range.	
Example:	RES:L1 20 OHM	Set constant resistance = 20 ohm for Load L1.
	RES:L2 10 OHM	Set constant resistance = 10 ohm for Load L2.
	RES:L1 MAX	Set constant resistance = maximum L1 value for Load L1.
	RES:L2 MIN	Set constant resistance = minimum L2 value for Load L2.
Query Syntax:	RESistance:L1?	
	RESistance:L2?	
	RESistance:L1? MAX	
	RESistance:L2? MIN	
Return Parameters:	<NR2> [Unit=OHM]	
Query Example:	RES:L1?	Return the set resistance value of Load L1.
Return Example:	10	

RESistance:RISE/FALL

Type:	Channel-Specific
Description:	Set the resistive slew rate for constant resistance.
Syntax:	RESistance:RISE <NRf+> [suffix]
	RESistance:FALL <NRf+> [suffix]
Parameters:	Refer to respective specification for valid value range.

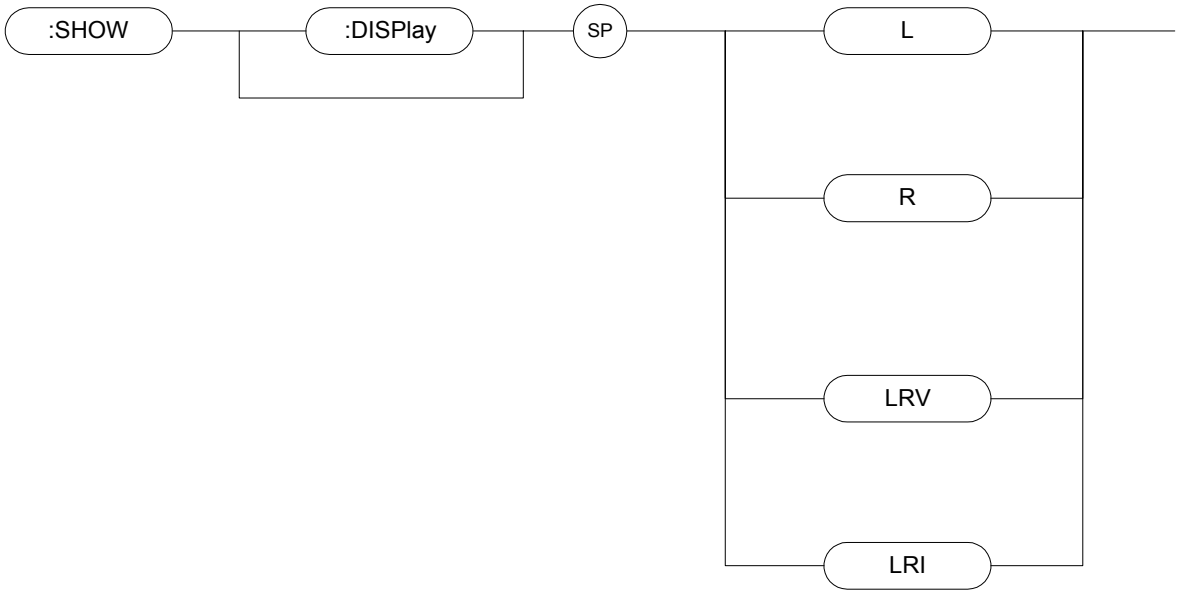
Example:	RES:RISE 2.5	Set CR rise slew rate to 2.5A/μS.
	RES:FALL 1A/μS	Set CR fall slew rate to 1A/μS.
	RES:RISE MAX	Set CR rise slew rate to the maximum programmable value.
	RES:FALL MIN	Set CR fall slew rate to the minimum programmable value.
Query Syntax:	RESistance:RISE?	
	RESistance:FALL?	
	RESistance:RISE? MAX	
	ESistance:FALL? MIN	
Return Parameters:	<NR2> [Unit=OHM]	
Query Example:	RES:RISE?	Return the CR rise slew rate.
Return Example:	2.5	

3.2.12 RUN Subsystem

:RUN

Type:	All Channels
Description:	Set all electronic loads to “ON”.
Syntax:	RUN

3.2.13 SHOW Subsystem



SHOW:DISPlay

Type: Channel-Specific (Double Channel Module Only)

Description: Set the display mode for the electronic load.

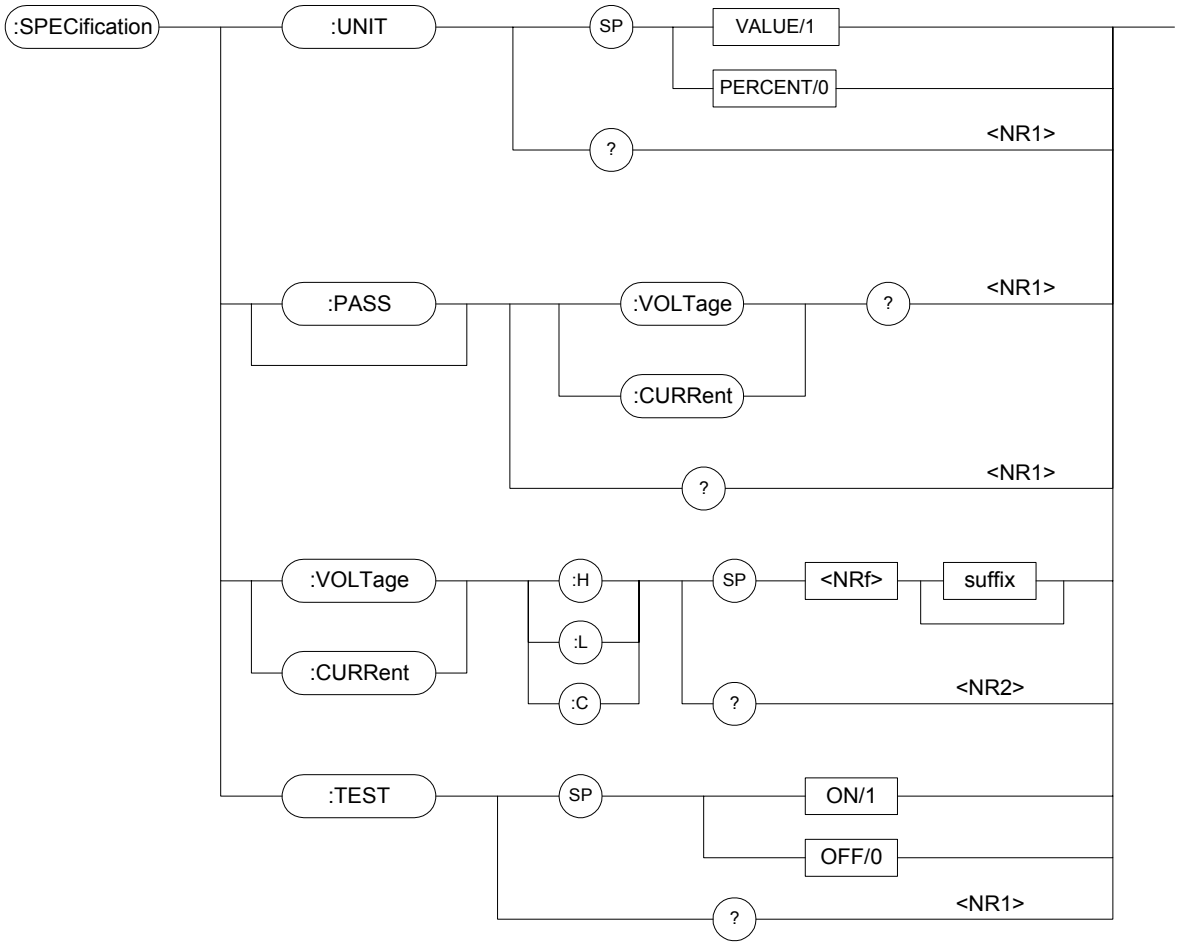
Syntax:
SHOW:DISPlay L
SHOW:DISPlay R
SHOW:DISPlay LRV
SHOW:DISPlay LRI

Parameters: L, R, LRV, LRI.

Example:

SHOW:DISP L	Display the voltage and current values of channel L.
SHOW:DISP R	Display the voltage and current values of channel R.
SHOW:DISP LRV	Display the voltage value of channel L and channel R.
SHOW:DISP LRI	Display the current value of channel L and channel R.

3.2.14 SPECIFICATION Subsystem



SPECification:UNIT

Type: All Channels
Description: Set the specific entry mode.
Syntax: SPECification:UNIT VALUE
SPECification:UNIT PERCENT
Parameters: VALUE/1, PERCENT/0
Example: SPEC:UNIT VALUE
SPEC: UNIT PERCENT
Query Syntax: SPECification:UNIT?
Query Example: SPEC:UNIT?
Return Parameters: <NR1>
Return Example: 0

SPECification:VOLTage?

Type: Channel-Specific
Description: Request GO-NG result reference to voltage specification.
Query Syntax: SPECification:VOLTage?
Query Example: SPEC:VOLT? Return voltage GO-NG result for CC and CR modes.
Return Parameters: <NR1>
Return Example: 0 (NG), 1 (GO)

SPECification:CURRent?

Type:	Channel-Specific
Description:	Request GO-NG result reference to current specification.
Query Syntax:	SPECification:CURRent?
Query Example:	SPEC:CURR? Return the current GO-NG result for CC mode.
Return Parameters:	<NR1>
Return Example:	0 (NG), 1 (GO)

SPECification?

Type:	All Channels
Description:	Request GO-NG result reference to all channels specifications.
Query Syntax:	SPECification?
Query Example:	SPEC? Return all channels GO-NG results.
Return Parameters:	<NR1>
Return Example:	0 (NG), 1 (GO)

SPECification:VOLTage

Type:	Channel-Specific
Description:	Set the voltage specification.
Syntax:	SPECification:VOLTage:H SPECification:VOLTage:L SPECification:VOLTage:C
Parameters:	Refer to respective specification for valid value range.
Example:	SPEC:VOLT:H <NRf+> [suffix] SPEC:VOLT:L <NRf+> [suffix] SPEC:VOLT:C <NRf+> [suffix]
Query Syntax:	SPECification:VOLTage:H? SPECification:VOLTage:L? SPECification:VOLTage:C?
Query Example:	SPEC:VOLT:H?
Return Parameters:	<NR2> [Unit=Voltage]
Return Example:	4.75

SPECification:CURRent

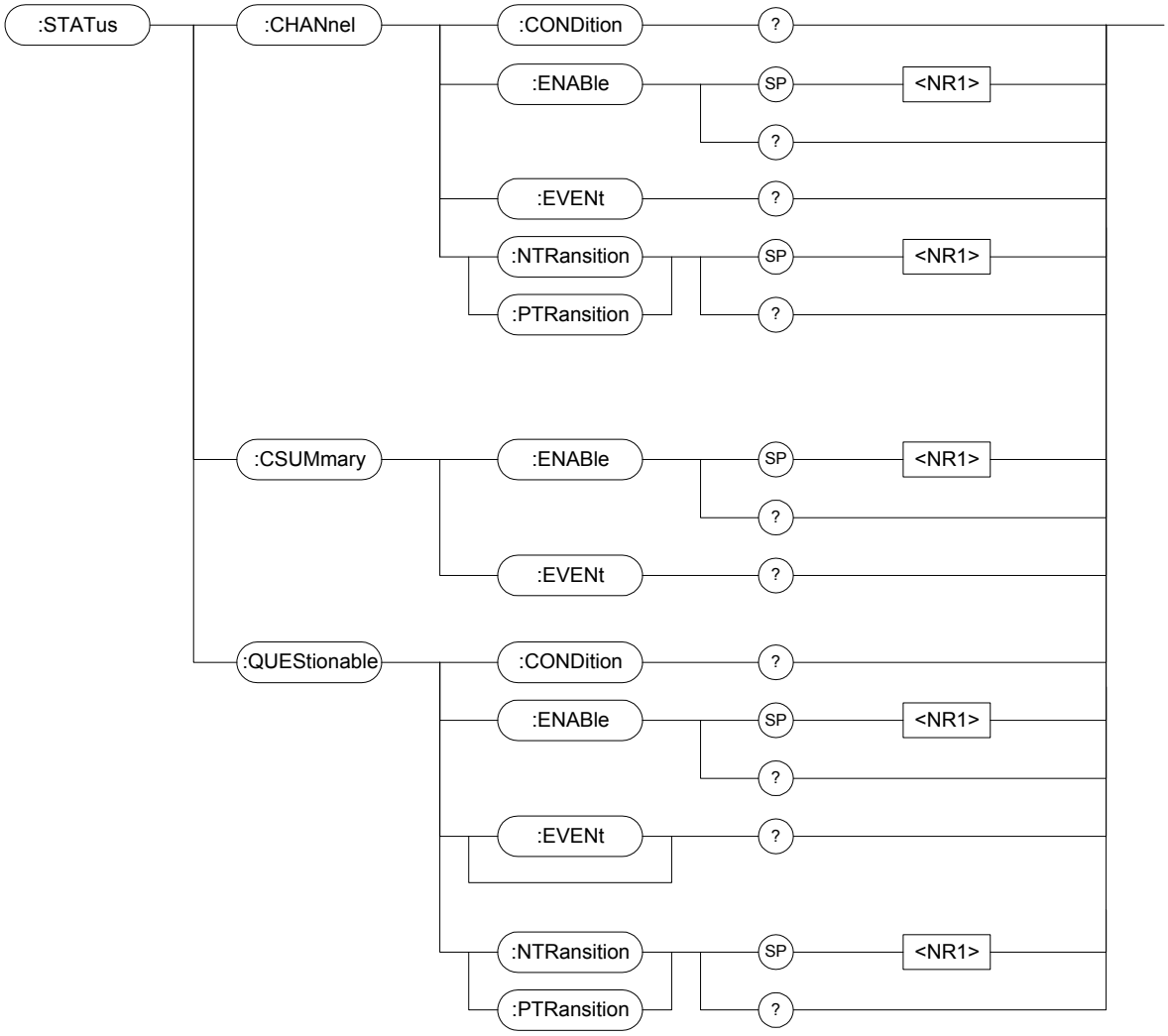
Type:	Channel-Specific
Description:	Sets the current specification.
Syntax:	SPECification:CURRent:H SPECification:CURRent:L SPECification:CURRent:C
Parameters:	Refer to respective specification for valid value range.
Example:	SPEC:CURR:H <NRf+> [suffix] SPEC:CURR:L <NRf+> [suffix] SPEC:CURR:C <NRf+> [suffix]
Query Syntax:	SPECification:CURR:H? SPECification:CURR:L? SPECification:CURR:C?
Query Example:	SPEC:CURR:H?

Return Parameters: <NR2> [Unit=Current]
Return Example: 4.75

SPECification:TEST

Type: Channel-Specific
Description: Start or close the specification test.
Syntax: SPECification:TEST ON
SPECification:TEST OFF
Parameters: ON/1, OFF/0
Example: SPEC:TEST ON
SPEC: TEST OFF
Query Syntax: SPECification:TEST?
Query Example: SPEC:TEST?
Return Parameters: <NR1>
Return Example: 1

3.2.15 STATUS Subsystem



STATus:CHANnel:CONDition

Type: Channel-Specific
Description: Return the real time channel status.
Query Syntax: STATus:CHANnel:CONDition?
Return Parameters: <NR1>

Bit Configuration of Channel Status Register

Bit Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Condition	0	0	0	0	0	0	0	0	0	0	0	OT	RV	OP	OV	OC
Bit Weight												16	8	4	2	1

Query Example: STAT:CHAN:COND? Return the status of the electronic load.
Return Example: 2048

STATus:CHANnel:ENABle

Type:	Channel-Specific	
Description:	Mask to select which bit in the Event register is allowed to be summed into the corresponding channel bit for the Channel Summary Event register.	
Syntax:	STATus:CHANnel:ENABle	
Parameters:	0 ~ 65535	
Example:	STAT:CHAN:ENABl 24	
Query Syntax:	STATus:CHANnel:ENABle	
Return Parameters:	<NR1>	
Query Example:	STAT:CHAN:ENABL?	Return the contents of the Status Channel Enable register.
Return Example:	24	

STATus:CHANnel:EVENt?

Type:	Channel-Specific	
Description:	Record all channel events that have occurred since last time the register was read, and reset the Channel Event register.	
Query Syntax:	STATus:CHANnel:EVENt?	
Return Parameters:	<NR1>	
Query Example:	STAT:CHAN:EVEN?	Read and reset the Channel Event register.
Return Example:	24	

STATus:CHANnel:PTRansition/NTRansition

Type:	Channel-Specific	
Description:	Programmable filters that determine what type of transition (0-to-1 or 1-to-0) in the Condition register will set the corresponding bit of the Event register.	
Syntax:	STATus:CHANnel:PTRansition/NTRansition <NRf>	
Parameters:	0 ~ 65535	
Example:	STAT:CHAN:PTR 4	Set OP(over power bit 2) from 0-to-1.
	STAT:CHAN:NTR 4	Set OP(over power bit 2) from 1-to-0.
Query Syntax:	STATus:CHANnel:PTRansition? STATus:CHANnel:NTRansition?	
Return Parameters:	<NR1>	
Query Example:	STAT:CHAN:PTR?	Inquiry setting for Channel PTRansition.
Return Example:	4	

STATus:CSUMmary:ENABle

Type:	Channel-Specific	
Description:	Mask to select which bit in the Channel Event register is allowed to be summed into the CSUM (Channel Summary) bit for the Status Byte register.	
Syntax:	STATus:CSUMmary:ENABle	
Parameters:		

Bit Configuration of Channel Summary Register

Bit Position	7	6	5	4	3	2	1	0
Channel	8	7	6	5	4	3	2	1
Bit Weight	128	64	32	16	8	4	2	1

Example: STAT:CSUM:ENAB 3
Query Syntax: STATus:CSUMmary:ENABle?
Return Parameters: <NR1>
Query Example: STAT:CSUM:ENAB? Return the setting of Channel Summary Enable register.
Return Example: 3

STATus:CSUMmary:EVENt

Type: Channel-Specific
Description: Indicate all channels of which an enabled STAT:CHAN Event has occurred since last time the register was read.
Syntax: STATus:CSUMmary:EVENt
Parameters:

Bit Configuration of Channel Summary Register

Bit Position	7	6	5	4	3	2	1	0
Channel	8	7	6	5	4	3	2	1
Bit Weight	128	64	32	16	8	4	2	1

Example: STAT:CSUM:EVEN 3
Query Syntax: STATus:CSUMmary:EVENt?
Return Parameters: <NR1>
Query Example: STAT:CSUM:EVEN? Return the value of the Channel Summary Event register.
Return Example: 3

STATus:QUEStionable:CONDition

Type: Channel-Specific
Description: Real-time ("live") recording of Questionable data
Query Syntax: STATus:QUEStionable:CONDition?
Return Parameters: <NR1>
Query Example: STAT:QUES:COND? Return the channel status.
Return Example: 6

STATus:QUEStionable:ENABle

Type: Channel-Specific
Description: Mask to select which bit on the Event register is allowed to be summed into the QUES bit for the Status Byte register.
Syntax: STATus:QUEStionable:ENABle
Parameters:

Bit Configuration of Questionable Status Register

Bit Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Condition	0	0	0	0	0	0	0	0	0	0	0	TE	RV	PE	VE	CE
Bit Weight												16	8	4	2	1

Example: STAT:QUES:ENAB 24

Query Syntax: STATus:QUEStionable:ENABle?
Return Parameters: <NR1>
Query Example: STAT:QUES:ENAB Return the setting of the Status Questionable Enable register.
Return Example: 24

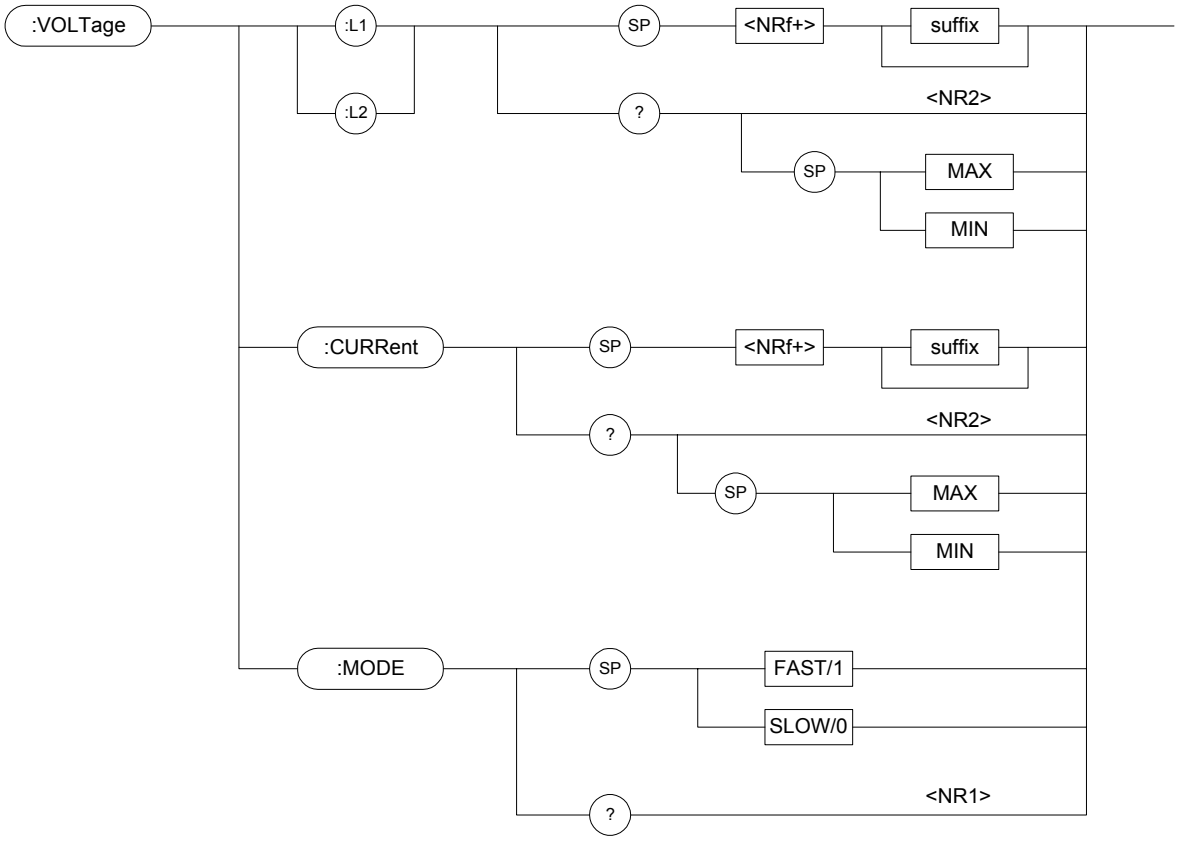
STATus:QUEStionable:EVENt?

Type: Channel-Specific
Description: Record all Questionable conditions that have occurred since last time the register was read.
Query Syntax: STATus:QUEStionable:EVENt?
Return Parameters: <NR1>
Query Example: STAT:QUES:EVEN? Return the contents of the Questionable Event register.
Return Example: 24

STATus:QUEStionable:PTRansition/NTRansition

Type: Channel-Specific
Description: Programmable filters determine what type of transition (0-to-1 or 1-to-0) in the Condition register will set the corresponding bit of the Event register.
Syntax: STATus:QUEStionable:PTRansition/NTRansition <NRf>
Parameters: 0 ~ 65535
Example: STAT:QUES:PTR 4 Set OP(over power bit 2) as 0-to-1.
STAT:QUES:NTR 4 Set OP(over power bit 2) as 1-to-0.
Query Syntax: STATus:QUEStionable:PTRansition?
STATus:QUEStionable:NTRansition?
Return Parameters: <NR1>
Query Example: STAT:QUES:PTR? Return the setting on the QUEStionable Ptransition/ Ntransition.
Return Example: 4

3.2.16 VOLTAGE Subsystem



VOLTage:L1/L2

Type: Channel-Specific

Description: Set the static load voltage in constant voltage mode.

Syntax: VOLTage:L1
VOLTage:L2

Parameters: Refer to respective specification for valid value range.

Example: VOLT:L1 8V — Set voltage of load L1 as 8V.
VOLT:L2 24V — Set voltage of load L2 as 24V.
VOLT:L1 MAX — Set voltage of load L1 as the maximum value.
VOLT:L2 MIN — Set voltage of load L2 as the minimum value.

Query Syntax: VOLTage:L1?
VOLTage:L2?
VOLTage:L1? MAX
VOLT:L2? MIN

Return Parameters: <NR2> [Unit=Voltage]

Query Example: VOLT:L1? — Return the set voltage value of load L1.

Return Example: 0

VOLTage:CURRent

Type:	Channel-Specific	
Description:	Set the current limit for constant voltage mode.	
Syntax:	VOLTage:CURRent	
Parameters:	Refer to respective specification for valid value range.	
Example:	VOLT:CURR 3	Set the loading current limit to 3A in constant voltage mode.
	VOLT:CURR MAX	Set the loading current limit to the maximum value in constant voltage mode.
	VOLT:CURR MIN	Set the loading current limit to the minimum value in constant voltage mode.
Query Syntax:	VOLTage:CURRent?	
Return Parameters:	<NR2> [Unit=Ampere]	
Query Example:	VOLT:CURR?	
Return Example:	3	

VOLTage:MODE

Type:	Channel-Specific
Description:	Set the response speed in CV mode.
Syntax:	VOLTage:MODE FAST VOLTage:MODE SLOW
Parameters:	FAST/1, SLOW/0
Example:	VOLT: MODE FAST VOLT:MODE SLOW
Query Syntax:	VOLTage:MODE?
Return Parameters:	<NR1>
Query Example:	VOLT:MODE?
Return Example:	0

3.2.17 System Commands

M

Type : All Channels
Description : Set the load mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : M “n,n,n,n,n,n,n,n”
Parameters(n) : 0: do not change, 1: CCL, 2: CCH, 3: CCDL, 4: CCDH, 5: CRL, 6: CRH, 7: CV
Example : M “1,1,2,2,2,2,5,5”
M ”2,2,2,2,2,2,2,0”

AC

Type : All Channels
Description : Set the current level 1(L1) of CC mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : AC “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=Ampere]
Example : AC “1.0,1,2.5,5.0,10.5,4.5,2.0,2.0”

AR

Type : All Channels
Description : Set the resistance level 1(L1) of CR mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : AR “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=OHM]
Example : AR “1.0,0.1,0.2,0.5,0.15,0.4,0.2,0.2”

AV

Type : All Channels
Description : Set the voltage level 1(L1) of CV mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : AV “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=Volt]
Example : AV “5.0,5.5,3.3,5.1,12.0,-5.5,5.0,5.2”

CCR

Type : All Channels
Description : Set the rising slew rate of CC mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : CCR “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=A/us]
Example : CCR “1.0,2.5,2.5,10,2.0,5.0,5.0,5.0”

CCF

Type : All Channels
Description : Set the falling slew rate of CC mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : CCF “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=A/us]
Example : CCF “1.0,2.5,2.5,10,2.0,5.0,5.0,5.0”

CRR

Type : All Channels
Description : Set the rising slew rate of CR mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : CRR “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=A/us]
Example : CRR “1.0,2.5,2.5,10,2.0,5.0,5.0,5.0”

CRF

Type : All Channels
Description : Set the falling slew rate of CR mode to the eight channels in one frame. The frame will ignore the setting when the channel does not exist.
Syntax : CRF “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=A/us]
Example : CRF “1.0,2.5,2.5,10,2.0,5.0,5.0,5.0”

LAT

Type : All Channels
Description : Set the action type of Von to the eight channels in one frame. The frame will ignore the setting when the channel does not exist.
Syntax : LAT “n,n,n,n,n,n,n,n”
Parameters(n) : 0: OFF, 1: ON
Example : LAT “0,1,1,1,0,1,0,1”

GO

Type : All Channels
Description : This command starts/stops current sinking of the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : GO “n,n,n,n,n,n,n,n”
Parameters(n) : 0: OFF, 1: ON, Other Value: no action
Example : GO “0,1,1,1,0,1,0,1”

VRB

Type : All Channels
Description : This command sets the voltage range of CC mode to the eight

channels in one frame. The frame will ignore the setting if the channel does not exist.

Syntax : VRB “n,n,n,n,n,n,n,n”
Parameters(n) : 0: LOW range, 1: HIGH range, Other Value: no action
Example : VRB “0,1,1,1,0,1,0,1”

VR

Type : All Channels
Description : This command sets the voltage range of CC mode to the eight channels in one frame. The frame will ignore the setting when the channel does not exist. The unit of the setting value is volt. Please refer to measurement section in the Specification table.
Syntax : VR “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=Volt]
Example : VR “-1,-1,2,16,80,10,80,16”

VON

Type : All Channels
Description : This command sets Von voltage to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : VON “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=Volt]
Example : VON “1.23,1.23,0,0,5,5,12,12”

CCSR

Type : All Channels
Description : Set both of the rising and the falling slew rate of CC mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : CCSR “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=A/us]
Example : CCSR “1.0,2.5,2.5,10,2.0,5.0,5.0,5.0”

CRSR

Type : All Channels
Description : Set both of the rising and the falling slew rate of CR mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : CRSR “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=A/us]
Example : CRSR “1.0,2.5,2.5,10,2.0,5.0,5.0,5.0”

CDL1

Type : All Channels
Description : Set the current level 1(L1) of CCDL/CCDH mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : CDL1 “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=Ampere]

Example : CDL1 “1.0,1,2.5,5.0,10.5,4.5,2.0,2.0”

CDL2

Type : All Channels
Description : Set the current level 2(L2) of CCDL/CCDH mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : CDL2 “n,n,n,n,n,n,n,n”
Parameters(i) : <NR2> [Unit=Ampere]
Example : CDL2 “1.0,1,2.5,5.0,10.5,4.5,2.0,2.0”

CDT1

Type : All Channels
Description : Set the active time T1 of current level 1(L1) of CCDL/CCDH mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : CDT1 “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=Second]
Example : CDT1 “1.0,1,2.5,5.0,10.5,4.5,2.0,2.0”

CDT2

Type : All Channels
Description : Set the active time T2 of current level 2(L2) of CCDL/CCDH mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : CDT2 “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=Second]
Example : CDT2 “1.0,1,2.5,5.0,10.5,4.5,2.0,2.0”

CDR

Type : All Channels
Description : Set the rising slew rate of CCDL/CCDH mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : CDR “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=A/us]
Example : CDR “1.0,2.5,2.5,10,2.0,5.0,5.0,5.0”

CDF

Type : All Channels
Description : Set the falling slew rate of CCDL/CCDH mode to the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : CDF “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=A/us]
Example : CDF “1.0,2.5,2.5,10,2.0,5.0,5.0,5.0”

L

Type : All Channels

Description : Set the load level according to mode setting for the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : L “n,n,n,n,n,n,n,n”
Parameters(n) : <NR2> [Unit=Ampere(CCL/CCH)]
[Unit=OHM(CRL/CRH)]
[Unit=Volt(CV)]
Example : L “1.0,2.5,2.5,10,2.0,5.0,5.0,5.0”

SRA

Type : All Channels
Description : This command resets the Von control signal to initial state for the eight channels in one frame. The frame will ignore the setting if the channel does not exist.
Syntax : SRA “n,n,n,n,n,n,n,n”
Parameters(n) : 1: RESET, Other Value: no action
Example : SRA “0,0,1,1,1,1,1,0”

4. Status Reporting

4.1 Introduction

This chapter explains the status data structure of Chroma 6330 Series electronic load as shown in Figure 4-1(on the next page). The standard registers, such as the Event Status register group, the Output Queue, the Status Byte and Service Request Enable registers, perform the standard GPIB functions and are defined in IEEE-488.2 Standard Digital Interface for Programmable Instrumentation. Other status register groups implement the specific status reporting requirements for the electronic load. The Channel Status and Channel Summary groups are used by multiple channel electronic load to enable the status information that will be kept at its own Status register for each channel.

4.2 Register Information in Common

- *Condition register*

The condition register represents the present status of electronic load signals. Reading the condition register does not change the state of its bits. Only changes in electronic load conditions affect the contents of this register.

- *PTR/NTR Filter, Event register*

The Event register captures changes in conditions corresponding to condition bits in a condition register, or to a specific condition in the electronic load. An event becomes true when the associated condition makes one of the following electronic load-defined transitions:

Positive TRansition (0 - to - 1)
Negative TRansition (1 - to - 0)
Positive or Negative TRansition (0-to-1 or 1-to-0)

The PTR/NTR filters determine what type of condition transitions set the bits in the Event register. Channel Status, Questionable Status allow transitions to be programmed. Other register groups, i.e. Channel Summary, Standard Event Status register group use an implied Rise (0-to-1) condition transition to set bits in the Event register. Reading an Event register clears it (all bits set to zero).

- *Enable register*

The Enable register can be programmed to enable the bit that the corresponding Event register is logically ORed into the Channel Summary.

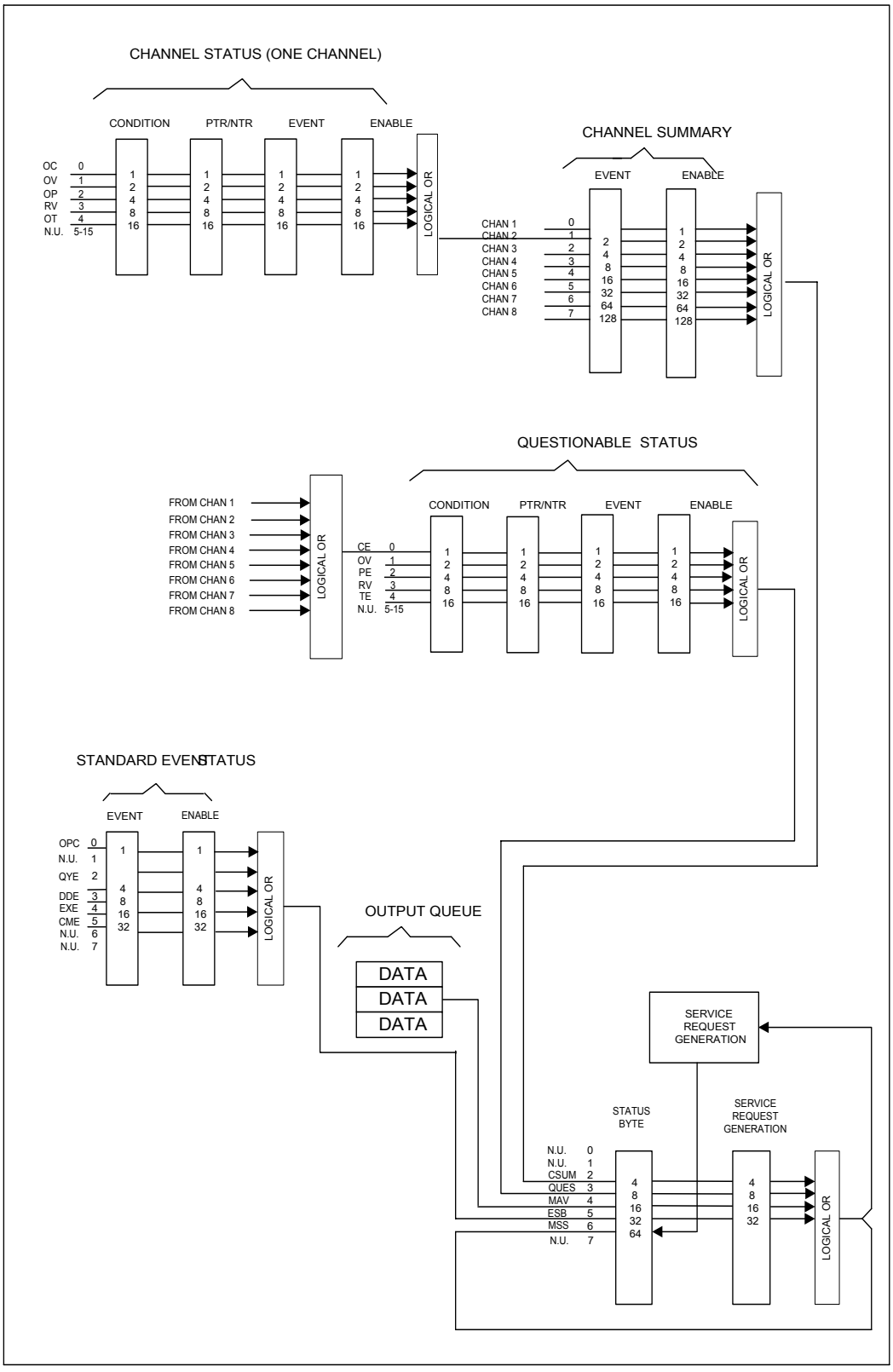


Figure 4-1 The Status Registers of Electronic Load

Mnemonic	Bit	Value	Meaning
OC	0	1	*Over current*. When an over current condition has occurred on a channel, Bit 0 is set and remains set until the over current condition is removed and LOAD:PROT:CLE is programmed.
OV	1	2	*Over voltage*. When an over voltage condition has occurred on a channel, Bit 1 is set and remains set until the over voltage condition is removed and LOAD:PROT:CLE is programmed.
OP	2	4	*Overpower*. An overpower condition has occurred on a channel, Bit 2 is set and remains set until the overpower condition is removed and LOAD:PROT:CLE is programmed.
RV	3	8	*Reverse voltage on input*. When a channel has a reverse voltage applied to it, Bit 3 is set. It remains set until the reverse voltage is removed and LOAD:PROT:CLE is programmed.
OT	4	16	*Over temperature*. When over temperature condition has occurred on a channel, Bit 4 is set and the channel is turned off. It remains set until the channel has cooled down below the over temperature trip point and LOAD:PROT:CLE is programmed.

Table 4-1 Bit Description of Channel Status

4.3 Channel Status

- The Channel Status register informs you one or more channel status conditions, which indicate certain errors or faults have occurred to a specific channel. Table 4-1 explains the channel status conditions that are applied to the electronic load.
- When the bits of the Channel Status Condition register are set, the corresponding condition is true.
- Program the PTR/NTR filter to select the way of condition transition in the Channel Status Condition register that will be set in the Event registers.
- Reading the Channel Status Event register resets itself to zero.
- The Channel Status Enable register can be programmed to specify the channel status event bit that is logically ORed to become the corresponding channel bit in Channel Summary Event register.

4.4 Channel Summary

- The Channel Summary registers summarize the channel status conditions up to 8 channels.
- When an enabled bit in the Channel Status Event register is set, it causes the corresponding channel bit in the Channel Summary Event register to be set.
- Reading the Event register will reset it to zero.
- The Channel Summary Enable register can be programmed to specify the channel summary event bit from the existing channels that is logically ORed to become Bit 2 (CSUM bit) in the Status Byte register.

4.5 Questionable Status

- The Questionable Status registers inform you one or more questionable status conditions, which indicate certain errors or faults have occurred to at least one channel. Table 4-2 lists the questionable status conditions that are applied to the electronic load. These conditions are same as the channel status conditions. Refer to Table 4-1 for a complete description.
- When a corresponding bit of Questionable Status Condition register is set, it indicates the condition is true.
- Program the PTR/NTR filter to select the way of condition transition in the Channel Status Condition register that will be set in the Event registers.
- Reading the Questionable Status Event register will reset it to zero.
- The Questionable status Enable register can be programmed to specify the questionable status event bit that is logically ORed to become Bit 3 (QUES bit) in the Status Byte register.

Mnemonic	Bit	Value	Meaning
CE/OC	0	1	*Current Error (Over current).*
OV	1	2	*Over voltage.*
PE/OP	2	4	*Power Error (Over power).*
RV	3	8	*Reverse voltage on input.*
TE/OT	4	16	*Temperature Error (Over temperature).*

Table 4-2 Bit Description of Questionable Status

4.6 Output Queue

- The Output Queue stores output messages until they are read from the electronic load.
- The Output Queue stores messages sequentially on a FIFO (First-In, First-Out) basis.
- It sets to 4 (MAV bit) in the Status Byte register when there are data in the queue.

4.7 Standard Event Status

- All programming errors that have occurred will set one or more error bits in the Standard Event Status register. Table 4-3 describes the standard events that apply to the electronic load.
- Reading the Standard Event Status register will reset it to zero.
- The Standard Event Enable register can be programmed to specify the standard event bit that is logically ORed to become Bit 5 (ESB bit) in the Status Byte register.

Mnemonic	Bit	Value	Meaning
OPC	0	1	*Operation Complete.* This event bit generated is responding to the *OPC command. It indicates that the device has completed all of the selected pending operations.
QYE	2	4	*Query Error*. The output queue was read when no data were present or the data in the queue were lost.
DDE	3	8	*Device Dependent Error*. Memory was lost, or self-test failed.
EXE	4	16	*Execution Error*. A command parameter was out of the legal range or inconsistent with the electronic load's operation, or the command could not be executed due to some operating conditions.
CME	5	32	*Command Error*. A syntax or semantic error has occurred, or the electronic load has received a <GET> message from program.

Table 4-3 Bit Description of Standard Event Status

4.8 Status Byte Register

- The Status Byte register summarizes all of the status events for all status registers. Table 4-4 describes the status events that are applied to the electronic load.
- The Status Byte register can be read with a serial of pull or *STB? query.
- The RQS bit is the only bit that is automatically cleared after a serial of pull.
- When the Status Byte register is read with a *STB? query, Bit 6 of the Status Byte register will contain the MSS bit. The MSS bit indicates that the load has at least one reason for requesting service. *STB? does not affect the status byte.
- The Status Byte register is cleared by *CLS command.

Status Byte Bit Description

Mnemonic	Bit	Value	
CSUM	2	4	*Channel Summary*. It indicates if an enabled channel event has occurred. It is affected by Channel Condition, Channel Event and Channel Summary Event registers.
QUES	3	8	*Questionable*. It indicates if an enabled questionable event has occurred.
MAV	4	16	*Message Available.* It indicates if the Output Queue contains data.
ESB	5	32	*Event Status Bit*. It indicates if an enabled standard event has occurred.
RQS/MSS	6	64	*Request Service/Master Summary Status*. During a serial of pull, RQS is returned and cleared. For a *STB? query, MSS is returned without being cleared.

Table 4-4 Bit Description of Status Byte

4.9 Service Request Enable Register

- The Service Request Enable register can be programmed to specify the bit in the Status Byte register that will generate the service requests.

5. An Example of Use

In this chapter a basic example of controlling electronic load is provided for use of GPIB. The GPIB used here is made by NI (National Instruments).

Examples:

```
#include “dec1.h”

#include <stdio.h>
#include <stdlib.h>
#include <sring.h>
#include <iostream.h>
#include <time.h>

static int MTA,
           MLA;

static int bd;

const char LA = 0x20,
           TA = 0x40;

static void setNi( int pad, char *cardName )
{
    MTA = TA + pad;
    MLA = LA + pad;
    if ( (bd = ibfind ( cardName ) ) < 0 )    {
       puts ( “GPIB Card Found Error” );
       exit ( 1 );
    }
    if ( ibpad ( bd, pad ) & ERR )    {
        puts ( “GPIB Card Address Assignment Error” );
        exit ( 3 );
    }
    ibtmo ( bd, 10 );
  ibsic ( bd );
  ibsre ( bd, 1 );
}

static void Niwrite( int pad, char *cmdStr )
{
   char cmd[4];

  cmd[0] = UNL;
  cmd[1] = UNT;
```

```
    cmd[2] = MTA;
    cmd[3] = LA + pad;
    //
    ibcmd( bd, cmd, 4 );
    ibwrt ( bd, cmdStr, _fstrlen( cmdStr ) );
    ibcmd( bd, cmd, 2 );
}

static char rxBuf[ 64 ]

static void Niread( int pad, char *queryStr )
{
     char cmd[ 4 ];

      Niwrite( pad, queryStr );
      cmd[ 0 ] = UNL;
      cmd[ 1 ] = UNT;
      cmd[ 2 ] = TA + pad;
      cmd[ 3 ] = MLA;
     //
      ibcmd( bd, cmd, 4 );
      ibrd( bd, rxBuf, sizeof( rxBuf ) - 1 );
      rxBuf[ ibcnt ] = ' \0 ';
      ibcmd( bd, cmd, 2);
}



void main( )
{
    setNi( 0, "GPIB" );                  // Set the status of PC's GPIB CARD.
    //
    Niread( 8, "*IDN?" );                // Read back identity code of 6314.
    cout << rxBuf << " \n\r ";           // Display on the screen of PC.
    //
    Niwrite( 8, "CHAN 1" );              // Set CHANNEL as 1.
    //
    Niread( 8,    "CHAN:ID?" );          // Read back identity code of channel 1.
    cout << rxBuf << " \n\r ";           // Display on the screen of PC.
    //
    Niwrite( 8, "MODE CCL" );            // Set CHANNEL 1 MODE as CCL.
    Niwrite ( 8, "CURR:STATIC:L1 1" ):   // Set L1 current of CCL as 1A.
    //
    Niread( 8,    "LOAD ON" );                 // Start sinking current.
    //
    Niread( 8, "MEAS:VOLT?" );                 // Measure the readings of voltage.
    cout << rxBuf << " \n\r ";                 // Display on the screen of PC.
    //
    Niread( 8, "MEAS:CURR?" );                 // Measure the readings of current.
```

```
    cout << rxBuf << " \n\r ";          // Display on the screen of PC.
    Niread( 8, "LOAD OFF" );            // Stop sinking current.
    //
    ibsic ( bd );
    ibon1( bd, 0 );
    ibsre ( bd, 0 );
}
```

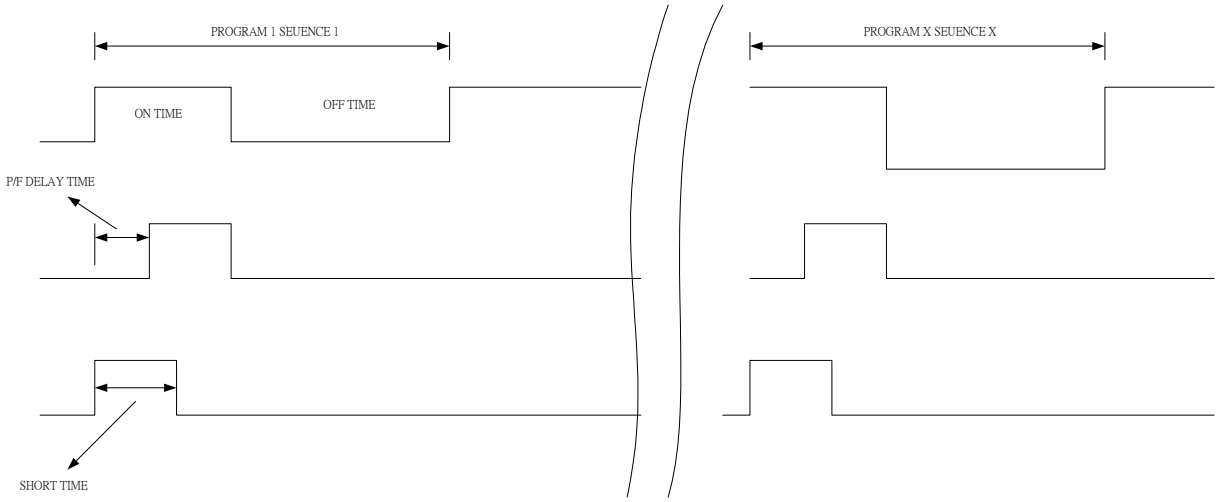
For the above example please refer to Chapter 3, and add corresponding commands according to the setting and control.

Example of PROGRAM RUN

You can use the following control procedures to run the PROGRAM.

```
<1> PROGram:FILE 1                  // Set the PROGRAM FILE to be run
<2> PROGram:ACTive 15               // Set the mapping action for Module Channel
                                    // chan 1 - chan 8 mapping value weights are
                                    // 1, 2, 4, 8, 16, 32, 64,128
<3> PROGram:CHAIN 0                 // program chain file No.
<4> PROGram:ONTime 3                // on time setting
<5> PROGram:OFFTime 2               // off time setting
<6> PROGram:SEQuence 1              // Sequence No. setting
<7> PROGram:SEQuence:MODE AUTO          // Sequence mode setting
<8> PROGram:SEQuence:SHORt:CHANnel 1    // Sequence short channel setting
<9> PROGram:SEQuence:SHORt:TIME 1       // Sequence short setting

<10> PROGram:SEQuence 2                 // sequence 2,sequence 3,....setting
.
.
.
<11> PROGram:SAVE                       // Save program setting data
.
.
<12> PROGram:RUN                        // Run PROGRAM
.
.
.
<13> PROGram:RUN?                       // Check if PROGRAM is running
```


PROGRAM 1 SEUENCE 1
PROGRAM X SEUENCE X
ON TIME
OFF TIME
P/F DELAY TIME
SHORT TIME