

Agilent 4291B RF Impedance/Material Analyzer
Programming Manual

SERIAL NUMBERS

This manual applies directly to instruments with serial number prefix “JP1KE” and above, or whose firmware is version 1.0.
For additional important information about serial numbers, read “Serial Number” in Appendix A of this Manual.



Agilent Part No. 04291-90037
Printed in JAPAN March 2001

Fourth Edition

Notice

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

This document contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of the Agilent Technologies.

Agilent Technologies Japan, Ltd.
Component Test PGU-Kobe
1-3-2, Murotani, Nishi-ku, Kobe-shi,
Hyogo, 651-2241 Japan

The customer shall have the personal, non-transferable rights to use, copy, or modify SAMPLE PROGRAMS in this manual for the Customer's internal operations. The customer shall use the SAMPLE PROGRAMS solely and exclusively for their own purpose and shall not license, lease, market, or distribute the SAMPLE PROGRAMS or modification of any part thereof.

Agilent Technologies shall not be liable for the quality, performance, or behavior of the SAMPLE PROGRAMS. Agilent Technologies especially disclaims that the operation of the SAMPLE PROGRAMS shall be uninterrupted or error free. The SAMPLE PROGRAMS are provided AS IS.

AGILENT TECHNOLOGIES DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Agilent Technologies shall not be liable for any infringement of any patent, trademark, copyright, or other proprietary rights by the SAMPLE PROGRAMS or their use. Agilent Technologies does not warrant that the SAMPLE PROGRAMS are free from infringements of such rights of third parties. However, Agilent Technologies will not knowingly infringe or deliver software that infringes the patent, trademark, copyright, or other proprietary right of a third party.

MS-DOS[®] is a U.S. registered trademark of Microsoft Corporation.

Manual Printing History

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates that are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

December 1997	First Edition (part number: 04291-90027)
March 1998	Second Edition (part number: 04291-90037)
December 1999	Third Edition (part number: 04291-90037)
March 2001	Fourth Edition (part number: 04291-90037)

Typeface Conventions

Bold	Boldface type is used when a term is defined. For example: icons are symbols.
<i>Italics</i>	Italic type is used for emphasis and for titles of manuals and other publications. Italic type is also used for keyboard entries when a name or a variable must be typed in place of the words in italics. For example: copy <i>filename</i> means to type the word copy, to type a space, and then to type the name of a file such as file1.
Computer	Computer font is used for on-screen prompts and messages.
 HARDKEYS	Labeled keys on the instrument front panel are enclosed in  .
 SOFTKEYS	Softkeys located to the right of the LCD are enclosed in  .

Related Documentation Information

You can obtain more detailed information than provided by this manual by referring to the following documents.

The following manuals are provided with the 4291B :

- *4291B GPIB Command Reference* for the complete GPIB command list of the analyzer.
- *4291B Quick Start Guide* for learning about the analyzer itself and its front panel key operation.
- *HP Instrument BASIC User's Handbook & its supplement for the 4291B* for Instrument BASIC information.

The following documents also provide related information:

- *HP BASIC Programming Guide* for learning HP BASIC programming. (Furnished with the HP BASIC system.)
- *Beginner's Guide to SCPI* for learning about a generic SCPI standard command set and its use (Agilent part no. H2325-90001).

Contents

1. Introduction	
How to Use This Manual	1-1
Target Reader	1-1
What's in This Manual?	1-1
How to Use the Program Modules	1-3
Building a Working Program Using Program Modules	1-3
Initializing Module	1-3
Example	1-4
GPIB Overview	1-5
Controller	1-5
Device Selector	1-6
GPIB Commands	1-7
Common Commands	1-7
Instrument Control Commands	1-7
Simple Commands	1-7
Program Message Syntax	1-8
Command Abbreviations	1-8
Upper and Lower Cases	1-8
Program Message Terminator	1-8
Multiple Messages	1-9
Query and Response Message Syntax	1-9
Parameters	1-9
Variable Types	1-9
Command Tree and Compound Header Usage	1-11
Preparation for Operation	1-13
Using Instrument BASIC for Controller	1-13
1. Connecting the GPIB Cables	1-13
2. Setting the GPIB Address	1-13
3. Preparing Instrument BASIC	1-13
Using an External Controller	1-14
1. Connecting the GPIB Cables	1-14
2. Setting the GPIB Address	1-14
3. Preparing HP BASIC	1-14
Sample Program Disk	1-15
Loading a Program from Disk	1-15

2. Setup and Measurement Program	
Overview of GPIB Control	2-1
Sending GPIB Commands	2-1
Reducing Keystrokes by Eliminating Node Repetition	2-2
Sending a Query and Reading the Response	2-2
Automating the Impedance Measurement Procedure	2-3
1. Setting the Active Channel	2-3
2. Setting Stimulus	2-4
Setting Frequency Sweep Range and Level	2-4
Setting OSC Level Sweep	2-4
Setting dc Voltage Sweep (Option 001 Only)	2-5
Setting dc Current Sweep (Option 001 Only)	2-5
3. Performing Calibration	2-6
Checking Calibration State	2-6
4. Setting Port Extension and Electrical Length	2-7
5. Performing Fixture Compensation	2-7
6. Setting Measurement Parameter	2-9
7. Setting Display Format	2-10
8. Setting dc Bias (Option 001 Only)	2-11
9. Triggering a Measurement	2-12
10. Setting Scale and Reference	2-13
11. Getting Measured Data to the Controller	2-13
Sample Program: Basic Impedance Measurement Program	2-14
Automating the Permittivity Measurement (Option 002 Only)	2-16
4. Selecting Fixture	2-16
5. Performing Fixture Compensation	2-16
6. Setting MUT Thickness	2-17
7. Setting Measurement Parameter	2-18
Cole-Cole Plot	2-18
Automating the Permeability Measurement (Option 002 Only)	2-19
4. Selecting Fixture	2-19
5. Performing Fixture Compensation	2-19
6. Setting MUT Size	2-20
7. Setting Measurement Parameter	2-20
3. Data Processing and Transfer	
Data Arrays	3-1
Raw Data Array	3-2
Data Array	3-2
Data Trace Array	3-2
Calibration Coefficient Array	3-3
Accessing Arrays	3-4
Compensation Coefficient Array	3-4
Accessing Arrays	3-4
Monitor Array	3-5
Stimulus Array	3-5
Arrays for Memory Trace	3-6
Accessing Memory Array	3-6
Accessing Memory Trace Array	3-6
Data Transfer Methods	3-7
ASCII Transfer	3-7
Binary Transfer	3-8
Data Header	3-9
Getting Data from Analyzer	3-10
Sample Programs: Compensation Data Transfer	3-11

Storing Compensation Data to Disk (CMP_STOR)	3-11
Loading Compensation Data from Disk (CMP_LOAD)	3-13
4. Using Status Reporting System	
General Status Register Model	4-1
Event Register	4-2
Enable Register	4-2
Status Byte Register	4-2
Transition Filter and Condition Register	4-3
Status Register Structure	4-4
How to Use the Status Registers in a Program	4-8
Reading an Event Register Directly	4-8
SRQ and Interrupt	4-9
Sample Program: Performing Calibration	4-11
5. Using the Trigger System	
Trigger System	5-2
Idle State	5-2
Wait for Trigger State	5-2
Measurement State	5-3
Sweeping Once Using the GPIB Trigger	5-4
Sweeping a Specified Number of Times	5-4
Triggering on Each Point Using the Manual Trigger	5-5
6. Using the I/O Port	
I/O Port Pin Assignment	6-1
Accessing I/O Port	6-2
Access I/O Port from the External Controller	6-2
Sample Program: BIN Sorting Using the I/O Port	6-3
7. Using the User Traces	
What's the User Trace?	7-1
Using a User Trace	7-2
Setting A Grid	7-3
Setting Data Train for The Trace	7-3
Turning ON the User Trace	7-4
Using the Marker on a User Trace	7-4
Clearing a User Trace	7-4
Sample Program: Time Characteristic Measurement	7-5
8. Programming Miscellaneous	
Using Disks	8-1
Saving the Analyzer Status	8-1
Entering Trace Data From the Disk into a Program Variable	8-2
Sample Program: Making HP CITIfile	8-3
Printing	8-5
To Print Analyzer Display	8-5
Printer Preparation	8-5
Execute Print	8-5
To Observe Printing	8-5
Controlling Instrument BASIC from an External Controller	8-6
Reading or Putting the Variable Data	8-6
Reading Numeric Variable	8-6
Putting Numeric Variable	8-6
Reading String Variable	8-7

Putting String Variable	8-7
Simultaneously Running Instrument BASIC and External Controller Programs	8-8
Controlling Instrument BASIC Execution Status	8-8
Determining Instrument BASIC Execution State	8-8
Transferring Program Source	8-10
Uploading a Program from Controller to Instrument BASIC	8-10
Downloading a program from Instrument BASIC to the Controller	8-10
Debugging Program	8-11
Processing Time Measurement	8-12
Key Sequence Logging	8-13
Generating Equivalent Program For Empty Editor	8-13
Inserting Equivalent Codes into Your Program	8-13
Limitations	8-14
File Transfer Function	8-15
File Transfer from 4291B to External Controller	8-16
File Transfer from External Controller to 4291B	8-18
Displaying List of Files in Current Directory	8-20
9. Facilitating Program Execution and Utilizing Storage Devices	
How to Save Programs of Instrument BASIC	9-1
The Procedure to Save Programs	9-1
Running a Program through the Softkey Interface	9-2
Automatically Starting a Program at Power-ON (AUTOST)	9-3
About AUTOREC	9-3
Using Storage Devices	9-4
BASIC Commands for Setting up the Storage Devices	9-4
Floppy Disk Drive	9-4
Memory Disk	9-4
Transferring Data between Floppy Disk and Memory Disk	9-4
10. Introducing HP instrument BASIC System	
Overview of HP instrument BASIC	10-1
Controlling the Analyzer	10-2
Using HP instrument BASIC for the First Time	10-2
Allocating Screen Area for HP instrument BASIC	10-2
Entering BASIC Statements from the Front Panel Keys	10-3
Getting into/out of the EDIT Mode	10-3
Getting into the EDIT Mode	10-3
Entering the EDIT Mode from the Keyboard	10-3
Getting Out of the EDIT Mode	10-3
Editing Programs in the EDIT Mode	10-4
Deleting Characters	10-4
Back Space	10-4
Deleting Characters	10-4
Inserting Characters	10-4
Moving the Cursor	10-4
Scrolling Lines and Pages	10-4
Scrolling Lines	10-4
Scrolling Pages	10-4
Jumping from the Current Line	10-5
Jumping to a Specified Line	10-5
Jumping to the Top/Bottom of a Program	10-5
Inserting/Deleting/Recalling Lines	10-5
Clearing Line	10-5
Renumbering Program Line Numbers	10-5

Listing Programs	10-6
Listing on the Screen	10-6
Listing to the Printer	10-6
Saving Programs (SAVE)	10-6
Listing File Names (CAT)	10-7
Listing to Screen	10-7
Listing to Printer	10-8
Getting Programs (GET)	10-8
On Key Label Function	10-8
Pass Control Between the External Controller	10-9
Pass Control	10-9
To Execute an HP instrument BASIC Command from the External Controller	10-10
To Load an Array in an HP instrument BASIC Program to the External Controller	10-11
Available I/O Interfaces and Select Codes	10-11
External RUN/CONTInue Connector	10-11
Graphics	10-12
HP instrument BASIC Graphics Commands	10-12
Hard Copies	10-13
Initial settings	10-13
Example of Graphics Programming	10-13
Drawing a Straight Line	10-13
Drawing a Circle	10-13
The Keyboard	10-14
Character Entry Keys	10-14
Cursor-Control and Display-Control Keys	10-14
Numeric Keypad	10-14
Editing Keys	10-15
Program Control Keys	10-15
System Control Keys	10-15
Softkeys and Softkey Control	10-16
Softkey Control Keys	10-16
Softkeys	10-16
Softkeys Accessed form (F10) Key	10-16
Using (CTRL) Key in Edit Mode	10-17
Run Light Indications	10-17
BASIC Commands Specific to 4291B	10-18
DATE	10-18
DATE\$	10-18
READIO	10-18
SET TIME	10-19
SET TIMEDATE	10-19
TIME	10-19
TIME\$	10-19
WRITEIO	10-20

11. Command Reference

Conventions and Definitions	11-1
ABORT	11-2
BLIGHT {OFF ON 0 1}	11-3
CALCulate Subsystem	11-4
CALCulate:EVALuate Subsystem	11-4
:BAND:FULL[:STATe] {OFF ON 0 1}	11-4
:BAND:SPAN DMARker	11-4
:BAND:STARt MARKer	11-4
:BAND:STOP MARKer	11-5

:COUple {OFF ON 0 1}	11-5
:EFFect:ON {1 2}	11-5
:EPARameters	11-6
:EPARameters:CIRCuit {A B C D E}	11-6
:EPARameters:SIMulation	11-6
:INTerpolate {OFF ON 0 1}	11-6
:MSTatistics {OFF ON 0 1}	11-6
:MSTatistics:DATA?	11-7
:MSTatistics[:STATe] {OFF ON 0 1}	11-7
:ON[1] "TR{1-21}"	11-7
:ON2 {"OFF" "ACV" "ACC" "DCV" "DCC"}	11-8
:PEAK:EXCursion {<numeric> DMARKer}	11-8
:PEAK:EXCursion:X {<numeric> DMARKer}	11-8
:PEAK:EXCursion[:Y] {<numeric> DMARKer}	11-9
:PEAK:POLarity {POSitive NEGative}	11-9
:PEAK:THReshold {<numeric> MARKer}	11-9
:PEAK:THReshold:STATe {OFF ON 0 1}	11-10
:R:FORMat {RIMaginary MLIPhase MLOPhase RX GB SWRPhase}	11-10
:REFerence:DATA?	11-10
:REFerence:X <numeric>	11-11
:REFerence:Y[1] <numeric>	11-11
:REFerence:Y2 <numeric>	11-11
:WIDTh:DATA?	11-12
:WIDTh:STATe {OFF ON 0 1}	11-12
:WIDTh:XPOSition:IN	11-12
:WIDTh:XPOSition:OUT	11-12
:WIDTh:Y {DIVS2 MULS2 DIV2 FIXed[,<numeric>]}	11-13
:Y{1-8}:DATA? [CH1 CH2]	11-13
:Y{1-8}:VALue{1 2}? [CH1 CH2]	11-14
:Y{1-8}:XPOSition <numeric>	11-14
:Y[1]:XPOSition:LPEak	11-14
:Y[1]:XPOSition:LTARget	11-15
:Y[1]:XPOSition:MAXimum	11-15
:Y[1]:XPOSition:MINimum	11-15
:Y[1]:XPOSition:NPEak	11-15
:Y[1]:XPOSition:PEAK	11-15
:Y{1-8}:XPOSition:POINt <numeric>	11-15
:Y[1]:XPOSition:RPEak	11-16
:Y[1]:XPOSition:RTARget	11-16
:Y[1]:XPOSition:TARGet <numeric>	11-16
:Y[1]:XPOSition:TRACk {MAXimum MINimum TARGet PEAK OFF}	11-16
CALCulate:FORMat {MLINear PHASe UPHase REAL IMAGinary LFACTor LTANgent CP CS LP LS D Q RP RS COMPLex}	11-17
CALCulate:FORMat:UNIT:ANGLE {DEG RAD}	11-18
CALCulate:LIMit Subsystem	11-18
:BEEPer {OFF ON 0 1}	11-18
:BEEPer:CONDition {PASS FAIL}	11-18
:BEEPer[:STATe] {OFF ON 0 1}	11-18
:CLEar	11-19
:CONTRol:OFFSet <numeric>	11-19
:LINE {OFF ON 0 1}	11-19
:OFFSet {<numeric> MARKer}	11-20
:SAVE	11-20
:SEGment <numeric>	11-20
:SEGment:ADD	11-21

:SEGment:CONTRol[:DATA] <numeric>	11-21
:SEGment:DELEte	11-21
:SEGment:DELTA <numeric>	11-21
:SEGment:EDIT	11-22
:SEGment:LOWer <numeric>	11-22
:SEGment:MIDDLE {<numeric> MARKer}	11-22
:SEGment:SAVE	11-22
:SEGment:UPPER <numeric>	11-23
:STATe {OFF ON 0 1}	11-23
CALCulate:MATH1 Subsystem	11-23
:CATalog?	11-23
:DIMension1 <numeric>	11-24
:DIMension2 <numeric(in)>,<numeric(out)>,<numeric(hei)>	11-24
[:EXPRession]:CATalog?	11-24
[:EXPRession]:NAME {ADM DCO PER RCO}	11-24
:STATe {OFF ON 0 1}	11-25
CALCulate:MATH2 Subsystem	11-25
[:EXPRession]:CATalog?	11-25
[:EXPRession]:NAME {SUB ADD DIV MUL}	11-26
:STATe {OFF ON 0 1}	11-26
CENT <numeric>, CHAN{1 2}, CLEM, CONT	11-27
DATA Subsystem	11-27
DATA[:DATA] {AOFF GAIN MZAP},<numeric>	11-27
DATA[:DATA] OFFS,{<numeric> MARKer}	11-28
DATA[:DATA] <array>,{<block> <numeric11>,<numeric12>,...,<numeric n2>}	11-28
DATA[:DATA] {EQC0 EQC1 EQL1 EQR1},<numeric>	11-29
DATA[:DATA]? LFA	11-29
DATA[:DATA]? LLIS	11-29
DATA[:DATA]? LMAR	11-30
DATA[:DATA]? MEM	11-30
DATA[:DATA]? SPAR	11-30
DATA[:DATA]:VALue? {SPAR DATA MEM MON},<numeric>	11-30
DATA:DEFine {OADM SIMP LIMP},{<numeric> DATA DTR TR1}	11-31
DATA:DELEte {OADM SIMP LIMP}	11-31
DATA:POINT? LFA	11-31
DATMEM	11-32
DIAGnostic Subsystem	11-32
DIAGnostic:EREFeRence:STATe?	11-32
DIAGnostic:FREVIision?	11-32
DIAGnostic:INIT:RESult?	11-32
DIAGnostic:SERVice Subsystem	11-33
DIAGnostic:TEST Subsystem	11-33
DFLT, DISP {DATA MEMO DATM}	11-34
DISPlay Subsystem	11-34
DISPlay:ANNOtation:FREQuency {OFF ON 0 1}	11-34
DISPlay:BACKlight {OFF ON 0 1}	11-34
DISPlay:BRIGhtness <numeric>	11-35
DISPlay:CMAP Subsystem	11-36
:COLor{1-14}:DEFault	11-36
:COLor{1-14}:HSL <numeric(Hue)>,<numeric(Sat)>,<numeric(Lum)>	11-36
:DEFault	11-37
:LOAD	11-37
:STORE	11-37
DISPlay:CONTRast <numeric>	11-37

DISPlay[:WINDow]:ALLocation {INSTrument HIHB BASic BSTatus}	11-37
DISPlay[:WINDow]:FORMat {FBACk ULOWer}	11-38
DISPlay[:WINDow]:GRAPhics:STATe {OFF ON 0 1}	11-38
DISPlay[:WINDow]:TEXT{1-40} Subsystem	11-38
:CLEar	11-39
:COLor <numeric>	11-39
[:DATA] <string>	11-39
:LOCate <numeric(x)>[,<numeric(y)>]	11-40
:PAGE {UP DOWN <numeric>}	11-40
:STATe {OFF ON 0 1}	11-41
DISPlay[:WINDow]:TRACe{1-21} Subsystem	11-41
:CLEar	11-41
:GRATicule:AXIS:COUPlE {OFF ON 0 1}	11-42
:GRATicule:FORMat {RECTangle POLar SMITH ADMittance CPLane}	11-42
:GRATicule:GRID[:STATe] {OFF ON 0 1}	11-43
:MARKer[1]:ALL DEFault	11-43
:MARKer[1]:ALL:STATe {OFF ON 0 1}	11-43
:MARKer[1]:RELative {OFF ON 0 1}	11-43
:MARKer[1]:RELative:REFerence {FIXed MARKer TRACked}	11-44
:MARKer{2-8}:STATe {OFF ON 0 1}	11-44
:MARKer{1-8}:UNIT {SPARameter TIME IOMega}	11-45
:STATe {OFF ON 0 1}	11-45
:X[:SCALE]:LEFt <numeric>	11-46
:X[:SCALE]:RIGHt <numeric>	11-46
:X[:SCALE]:RLEVel <numeric>	11-46
:X:SPACing {LINear LOGarithmic OBASe}	11-47
:X:UNIT <string>	11-47
:Y[:SCALE]:AUTO ONCE	11-48
:Y[:SCALE]:BOTTom <numeric>	11-48
:Y[:SCALE]:COUPlE {OFF ON 0 1}	11-48
:Y[:SCALE]:PDIVision <numeric>	11-49
:Y[:SCALE]:RLEVel {<numeric> MARKer}	11-49
:Y[:SCALE]:RPOSITION <numeric>	11-50
:Y[:SCALE]:TOP <numeric>	11-50
:Y:SPACing {LOGarithmic LINear}	11-50
:Y:UNIT <string>	11-51
DISSMEMO {OFF ON 0 1}, DUAC {OFF ON 0 1}	11-52
DPI <numeric>	11-52
DUAM {IMPH IRIM APPH ARIM LSQ LPQ CSD CPD DMPH DRIM PMPH PRIM}	11-52
FMT {LINY LOGY POL SMIT ADM COMP}	11-52
FORMat Subsystem	11-52
FORMat[:DATA] {ASCIi REAL,32 REAL,64 PACKed,32}	11-52
FREO	11-54
FORMFEED {OFF ON 0 1}	11-54
HCOPY Subsystem	11-54
HCOPY	11-54
HCOPY:ABORt	11-54
HCOPY:DEFault	11-54
HCOPY:DEVIce:CMAP:COLor {FIXed VARiable}	11-55
HCOPY:DEVIce:COLor {OFF ON 0 1}	11-55
HCOPY:DEVIce:DPI <numeric>	11-55
HCOPY:DEVIce:FORMFeed {OFF ON 0 1}	11-56
HCOPY:DEVIce:LANDScape {OFF ON 0 1}	11-56
HCOPY:DEVIce:LEFtMarg <numeric>	11-56
HCOPY:DEVIce:SKEY {OFF ON 0 1}	11-57

HCOPy:DEvIce:TOPMarg <numeric>	11-57
HCOPy[:IMMediate]	11-57
HCOPy:ITEM Subsystem	11-57
:TDSTamp:STATe {OFF ON 0 1}	11-58
HOLD	11-59
INITiate Subsystem	11-59
INITiate:CONTInuous {OFF ON 0 1}	11-59
INITiate[:IMMediate]	11-59
INITiate[:IMMediate]:AGain:ALL	11-60
INSTrument Subsystem	11-61
INSTrument {CH1 CH2}	11-61
INSTrument:COUPlE {ALL NONE}	11-61
INSTrument:NSElect {1 2}	11-61
INSTrument[:SElect] {CH1 CH2}	11-62
INSTrument:STATe {OFF ON 0 1}	11-62
LANDSCAPE {OFF ON 0 1}, LISD {FBAS OBAS}, LMARG <numeric>	11-63
MARDCENT, MARDSPAN, MARK {OFF ON 0 1}, MARKCENT, MARKREF MARKSTAR, MARKSTOP, MARZ	11-63
MATH {DATA DMNM DPLM DDVM DMLM}	11-63
MEAS {IMAG IPH IRE IIM AMAG APH ARE AIM RCM RCPH RCR RCIM DCM DCPH DCR DCIM PHMA PPH PRE PIM CP CS LP LS D Q RP RS}	11-63
MMEMory Subsystem	11-63
MMEMory:CDIRectory [<string>]	11-63
MMEMory:COpy {<string(s)>,<string(m_s)>,<string(d)>,<string(m_d)>}	11-64
MMEMory:CREate:DIRectory <string>	11-64
MMEMory:DELEte <string(file_name)>[,<string(m)>]	11-64
MMEMory:INITialize <string>,{LIF DOS}	11-65
MMEMory:LOAD Subsystem	11-65
:STATe <string(file_name)>[,<string(m)>]	11-65
:TRACe SEL,<string(file_name)>[,<string(m)>]	11-65
MMEMory:STORe Subsystem	11-66
:DINTerchange:TIFF <string(file_name)>[,<string(m)>]	11-66
:DINTerchange:TRACe SEL,<string(file_name)>[,<string(m)>]	11-66
:ITEM:TRACe:CATalog?	11-66
:ITEM:TRACe:DELEte {CCO DATA DTR UTR MEM MTR RAW}	11-67
:ITEM:TRACe:SELEct {CCO DATA DTR UTR MEM MTR RAW}	11-67
:STATe <string(file_name)>[,<string(m)>]	11-68
:TRACe SEL,<string(file_name)>[,<string(m)>]	11-68
NUMG <numeric>	11-69
PROGram Subsystem	11-69
PROGram:CATalog?	11-69
PROGram[:SELEcted] Subsystem	11-69
:DEFine <block>	11-69
:DELEte[:SELEcted]	11-70
:DELEte:ALL	11-70
:EXECute <string>	11-70
:MALLocate {<numeric> DEFault}	11-71
:NAME <string>	11-71
:NUMBer <var>,<numeric1>,<numeric2>,. . . ,<numeric n>	11-71
:STATe {RUN PAUSe STOP CONTInue}	11-72
:STRing <var>,<string1>,<string2>,. . . ,<string n>	11-72
:WAIT	11-72
PROGram:EXPLicit Subsystem	11-73
:DEFine "PROG",<block>	11-73

:DELEte "PROG"	11-73
:EXECute "PROG",<string>	11-73
:MALLocate "PROG",<numeric> DEFault}	11-73
:NAME "PROG",<string>	11-73
:NUMBer "PROG",<var>,<numeric1>,<numeric2>,...,<numeric n>	11-74
:STATe "PROG",{RUN PAUSe STOP CONTInue}	11-74
:STRing "PROG",<var>,<string1>,<string2>,...,<string n>	11-74
:WAIT "PROG"	11-74
PEAKCENT, POIDTIME <numeric>, PRSOFT {OFF ON 0 1}, SELM <numeric>	11-75
SAVDSTAC <string>, SAVDTIF <string>, STOD{DISK MEMO}, STORMDISK	11-75
SENSe Subsystem	11-75
SENSe:AVERAge1:COUNT <numeric>	11-75
SENSe:AVERAge1[:STATe] {OFF ON 0 1}	11-75
SENSe:AVERAge2:CLEAr	11-76
SENSe:AVERAge2:COUNT <numeric>	11-76
SENSe:AVERAge2[:STATe] {OFF ON 0 1}	11-76
SENSe:CORRection1 Subsystem	11-76
:CKIT {APC7 UDEfined}	11-77
:CKIT:LABel <string>	11-77
:CKIT:SAVE	11-77
:CKIT:STANdard1:C <numeric>	11-77
:CKIT:STANdard1:G <numeric>	11-78
:CKIT:STANdard2:L <numeric>	11-78
:CKIT:STANdard2:R <numeric>	11-78
:CKIT:STANdard3:R <numeric>	11-78
:CKIT:STANdard3:X <numeric>	11-79
:COLLect[:ACQuire] {STANdard1 STANdard2 STANdard3 STANdard4}	11-79
:COLLect:FPOints {FIXed USER}	11-79
:COLLect:SAVE	11-80
:EDELay:STATe {OFF ON 0 1}	11-80
:EDELay[:TIME] <numeric>	11-80
[:STATe]?	11-81
SENSe:CORRection2 Subsystem	11-82
:CKIT[1]:LABel <string>	11-82
:CKIT[1]:SAVE	11-82
:CKIT[1]:STANdard1:C <numeric>	11-82
:CKIT[1]:STANdard1:G <numeric>	11-82
:CKIT[1]:STANdard1[:SELEct] {LIST LPARAmeter}	11-83
:CKIT[1]:STANdard2:L <numeric>	11-83
:CKIT[1]:STANdard2:R <numeric>	11-83
:CKIT[1]:STANdard2[:SELEct] {LIST LPARAmeter}	11-83
:CKIT[1]:STANdard3:L <numeric>	11-84
:CKIT[1]:STANdard3:R <numeric>	11-84
:CKIT[1]:STANdard3[:SELEct] {LIST LPARAmeter}	11-84
:CKIT2 {TEFLon UDEfined}	11-84
:CKIT2:LABel <string>	11-85
:CKIT2:SAVE	11-85
:CKIT2:STANdard6:PREAl <numeric>	11-85
:CKIT2:STANdard6:PLFactor <numeric>	11-85
:CKIT2:STANdard6:THICKness <numeric>	11-86
:COLLect[:ACQuire] STANdard{1-7}	11-86
:COLLect:FPOints {FIXed USER}	11-86
:COLLect:SAVE	11-87
:OPEN[:STATe] {OFF ON 0 1}	11-87
:SHORT[:STATe] {OFF ON 0 1}	11-87

:LOAD[:STATe] {OFF ON 0 1}	11-87
SENSE:FREQUency Subsystem	11-88
:CENTer {<numeric> DMARker MARKer TPEak}	11-88
:MODE {FIXed LIST SWEep}	11-88
:SPAN {<numeric> DMARker MZAPerture}	11-89
:START {<numeric> MARKer}	11-89
:STOP {<numeric> MARKer}	11-89
SENSE:LIST Subsystem	11-90
:CLEar	11-90
:SAVE	11-90
:SEGment <numeric>	11-90
:SEGment:ADD	11-90
:SEGment:AVERage:COUNT <numeric>	11-90
:SEGment:CURRent <numeric>	11-91
:SEGment:DELEte	11-91
:SEGment:EDIT	11-91
:SEGment:FREQUency:CENTer <numeric>	11-91
:SEGment:FREQUency:SPAN <numeric>	11-92
:SEGment:FREQUency:START {<numeric> MARKer}	11-92
:SEGment:FREQUency:STOP {<numeric> MARKer}	11-92
:SEGment:POINts <numeric>	11-93
:SEGment:POWEr <numeric>	11-93
:SEGment:QUIT	11-93
:SEGment:SAVE	11-93
:SEGment:VOLTagE <numeric>	11-94
SENSE:SWEep:COUNT <numeric>	11-94
SENSE:SWEep:DWELl{1 2} <numeric>	11-94
SENSE:SWEep:DWELl{1 2}:AUTO {OFF ON 0 1}	11-95
SENSE:SWEep:POINts <numeric>	11-95
SENSE:SWEep:SPACing {LINear LOGarithmic}	11-95
SENSE:SWEep:TIME <numeric>	11-96
SENSE:SWEep:TIME:AUTO {OFF ON 0 1}	11-96
SING	11-97
SOURce Subsystem	11-97
SOURce1:FREQUency[:CW]:FIXed <numeric>	11-97
SOURce1:{CURRent POWEr}[:LEVel][:IMMediate][:AMPLitude] <numeric>	11-97
SOURce1:SWEep:DIRection {UP DOWN}	11-98
SOURce1:SWEep:SPACing {LINear LOGarithmic}	11-98
SOURce1:VOLTagE Subsystem	11-98
:CENTer {<numeric> DMARker MARKer TPEak}	11-98
[:LEVel][:IMMediate][:AMPLitude] <numeric>	11-99
:MODE {FIXed LIST SWEep}	11-99
:SPAN {<numeric> DMARker MZAPerture}	11-100
:START {<numeric> MARKer}	11-100
:STOP {<numeric> MARKer}	11-100
SOURce2:{CURRent VOLTagE} Subsystem	11-101
:ALC[:STATe] {OFF ON 0 1}	11-101
:CENTer {<numeric> DMARker MARKer TPEak}	11-101
[:LEVel][:IMMediate][:AMPLitude] <numeric>	11-101
:LIMit[:AMPLitude] <numeric>	11-102
:MODE {FIXed SWEep}	11-102
:SPAN {<numeric> DMARker MZAPerture}	11-103
:START {<numeric> MARKer}	11-103
:STATe {OFF ON 0 1}	11-103
:STOP {<numeric> MARKer}	11-104

SOURce2:SWEep:DIRection {UP DOWN}	11-104
SOURce2:SWEep:SPACing {LINear LOGarithmic}	11-104
SPAN <numeric>, STAR <numeric>	11-105
STATus Subsystem	11-105
STATus:INSTrument Subsystem	11-105
:ENABle <numeric>	11-105
[:EVENT]?	11-105
STATus:OPERation Subsystem	11-106
:CONDition?	11-106
:ENABle <numeric>	11-106
[:EVENT]?	11-106
:NTRansition <numeric>	11-107
:PTRansition <numeric>	11-107
STATus:PRESet	11-107
STATus:QUEStionable Subsystem	11-108
:CONDition?	11-108
:ENABle <numeric>	11-108
[:EVENT]?	11-108
STOP <numeric>	11-109
STYPE {LIN LOG LIST}, SWED {UP DOWN}, SWEDTIME <numeric>, SWES {FREQ OLEV DCV DCI}	11-109
SYSTem Subsystem	11-109
SYSTem:BEEPer{1 2}:STATe {OFF ON 0 1}	11-109
SYSTem:COMMunicate:GPIB Subsystem	11-109
:CONTRoller:ADDRess <numeric>	11-110
SYSTem:COMMunicate:PARallel Subsystem	11-110
[:RECEive]:DATA?	11-110
:TRANsmiT:DATA <numeric>	11-110
SYSTem:DATE <numeric(year)>,<numeric(month)>,<numeric(day)>	11-111
SYSTem:DATE:MODE {MDY DMY}	11-111
SYSTem:ERRor?	11-111
SYSTem:FIXTure {NONE HP16191 HP16192 HP16193 HP16194 HP16453 HP16454S HP16454L UDEFined}	11-111
SYSTem:FIXTure:DISTance <numeric>	11-112
SYSTem:FIXTure:LABel <string>	11-112
SYSTem:FIXTure:SAVE	11-112
SYSTem:KEY <numeric>	11-112
SYSTem:KLOCK {OFF ON 0 1}	11-113
SYSTem:PRESet	11-114
SYSTem:SECurity[:STATe] {ON 1}	11-114
SYSTem:TIME <numeric(hour)>,<numeric(min)>,<numeric(sec)>	11-114
SYSTem:VERSion?	11-115
TMARG <numeric>	11-116
TRACe Subsystem	11-116
TRACe:COPIY TR{2-17},TR1	11-116
TRACe:COPIY TR{18-21},TR{1-17}	11-116
TRACe[:DATA] <trace>,{<block> <numeric1>,<numeric2>,...,<numeric n2>}	11-116
TRACe[:DATA] {TRX{18-21} TRY{18-21}},{<block> <numeric1>, <numeric2>,...,<numeric n>}	11-117
TRACe[:DATA]:VALue? <trace>,<numeric>	11-117
TRACe:POINts TR{18-21}[,<numeric>]	11-118
TRIGger Subsystem	11-119
TRIGger:EVENt:TYPE {POINt SWEep}	11-119
TRIGger:SLOPe {POSitive NEGative}	11-119

TRIGger:SOURce {BUS EXTeRnal INTeRnal MANual}	11-119
USKEY	11-121
Common Commands	11-121
*CLS	11-121
*ESE <numeric>	11-121
*ESR?	11-121
*IDN?	11-121
*OPC	11-122
*OPT?	11-122
*PCB <numeric>	11-122
*RST	11-123
*SRE <numeric>	11-123
*STB?	11-123
*TRG	11-123
*TST?	11-123
*WAI	11-124
Simple Commands	11-125
BLIGHT {OFF ON 0 1}	11-125
CENT <numeric>	11-125
CHAN1	11-125
CHAN2	11-126
CLEM	11-126
CLOSE	11-126
CONT	11-126
CWD?	11-126
DATMEM	11-126
DFLT	11-127
DISP {DATA MEMO DATM}	11-127
DISSMEMO {OFF ON 0 1}	11-127
DPI <numeric>	11-128
DUAC {OFF ON 0 1}	11-128
DUAM {IMPH IRIM AMPH ARIM LSQ LPQ CSD CPD DRLF DRLT DLFLT DMLT PRLF PRCLT PLFLT PMLT}	11-128
FMT {LINY LOGY POL SMIT ADM COMP}	11-129
FNAME? <numeric>	11-130
FNUM?	11-130
FORMFEED {OFF ON 0 1}	11-130
FREQ	11-130
FSIZE? <string>	11-131
HOLD	11-131
LANDSCAPE {OFF ON 0 1}	11-131
LMARG <numeric>	11-131
LISD {FBAS OBAS}	11-132
MARDCENT	11-132
MARDSPAN	11-132
MARK {OFF ON 0 1}	11-132
MARKCENT	11-133
MARKREF	11-133
MARKSTAR	11-133
MARKSTOP	11-133
MARZ	11-133
MATH {DATA DMNM DPLM DDVM DMLM}	11-133
MEAS {IMAG IPH IRE IIM AMAG APH ARE AIM RCM RCPH RCR RCIM DCR DCLF DCLT DCM PRE PLF PLT PMAG CP CS LP LS D Q RP RS}	11-134
NUMG <numeric>	11-135

PEAKCENT	11-135
POIDTIME <numeric>	11-135
PRSOFT {OFF ON 0 1}	11-135
READ?	11-136
ROPEN <string>	11-136
RESTMDISK	11-137
SAVDSTAC <string>	11-137
SAVDTIF <string>	11-137
SELM <numeric>	11-138
SING	11-138
SPAN <numeric>	11-138
STAR <numeric>	11-138
STOD{DISK MEMO}	11-139
STOP <numeric>	11-139
STORMDISK	11-139
STYPE {LIN LOG LIST}	11-139
SWED {UP DOWN}	11-140
SWEDTIME <numeric>	11-140
SWES {FREQ OLEV DCV DCI}	11-140
TMARG <numeric>	11-141
USKEY	11-141
WOPEN <string>[,<numeric>]	11-141
WRITE <block>	11-142

A. Manual Changes

Introduction	A-1
Manual Changes	A-1
Serial Number	A-2
Change 1	A-3

B. Complex Operation Sub Program

Complex Operation Sub Program	B-1
Sample Program: Using Complex Operation Sub Program	B-2

C. GPIB Command List by Function

Front Panel Key List with Equivalent GPIB Commands	C-2
GPIB Only Functions and the GPIB Commands	C-33

D. SCPI Conformance Information

SCPI Version	D-1
GPIB Commands Compatible to SCPI	D-1
Simple Commands	D-14

E. Measurement Parameter Settings Using GPIB Commands

Messages

Index

Figures

1-1. GPIB Device and Address	1-6
1-2. Program Message Terminators	1-8
1-3. Command Tree and Compound Header Usage	1-12
2-1. Setting Parameters	2-3
3-1. Simplified Data Processing Flow	3-1
3-2. Simplified Internal Process of ASCII and Binary Transfer	3-7
3-3. IEEE 64-bit floating point format	3-8
3-4. IEEE 32-bit floating point format	3-9
3-5. Binary Data Header	3-9
4-1. General Status Register Model	4-1
4-2. Transition Filter and Condition Register	4-3
4-3. Status Register Structure	4-4
4-4. SRQ Generation Sequence	4-10
5-1. Simplified Trigger System	5-2
6-1. I/O Port Pin Assignment	6-1
6-2. Connecting I/O Port	6-3
6-3. Timing Chart	6-4
7-1. User Trace	7-2
8-1. Example of HP CITIfile	8-3
8-2. Sample Program : To Observe Printing	8-5
10-1. Sample Program : To Transfer the Program to IBASIC (on External Controller)	10-10
10-2. Sample Program : To Load HP instrument BASIC Program Array (on External Controller)	10-11
10-3. Screen Structure	10-12
11-1. Key Codes	11-113
11-2. Fixed length block format	11-136
11-3. Procedure of executing commands to read/write data	11-137
A-1. Serial Number Plate	A-2
E-1. Data Formatting Inside 4291B	E-1

Tables

3-1. Calibration Coefficients	3-3
4-1. Status Bit Definitions of the Status Byte Register	4-5
4-2. Status Bit Definitions of the Instrument Event Status Register	4-6
4-3. Status Bit Definitions of the Standard Event Status Register	4-7
4-4. Status Bit Definitions of the Operation Status Register	4-7
7-1. GPIB Commands for User Trace 1 to 4	7-2
A-1. Manual Changes by Serial Number	A-1
A-2. Manual Changes by Firmware Version	A-1
D-1. IEEE 488.2 Common Commands	D-1
D-2. Instrument Control Commands	D-2
D-3. Simple Commands	D-14
E-1. GPIB Commands Setting Measurement Parameters (1/2)	E-2
E-2. GPIB Commands Setting Measurement Parameters (2/2)	E-3

Introduction

How to Use This Manual

This manual introduces GPIB programming for the 4291B. It provides additional information on how to write programs that might be difficult to understand when using only the GPIB Command Reference. It also provides information, techniques, and examples of how to effectively control GPIB instruments.

To use this manual effectively, you need one of the following GPIB controllers:

- HP instrument BASIC. That is an internal GPIB controller in the 4291B.
- HP Vectra PC (or IBM compatible PC) with HP BASIC for Windows. An HP 9000 Series 700 (200 or 300) UNIX computer that has BASIC/UX.

This manual helps you to learn how to write programs that control the 4291B. To help you learn quickly, many sample modules and programs are provided.

Target Reader

A target reader of this manual is a programmer who wants to control the 4291B through the GPIB interface.

This manual explains GPIB programming using HP BASIC. Therefore, you should have some experience using BASIC. If you have never written a program in BASIC, review the applicable documentation listed at the end of this chapter before starting this manual. This manual does not require extensive knowledge of BASIC programming.

This manual assumes you understand the operations and features of the 4291B. If you have never operated the 4291B, read the *Quick Start Guide* to learn how to operate the 4291B.

What's in This Manual?

The following chapters are provided in this manual:

- Chapter 1 “Introduction” provides an introduction to this manual, how to use a sample program, an GPIB overview, hardware preparation, and a description of the sample program disk. This chapter provides important information that is used throughout this manual. You should read this chapter first.
- Chapter 2 “Setup and Measurement Program” provides GPIB command basics. It also shows how to build a measurement program including setups, compensating, triggering, and getting data. If you want to build an automated measurement program, read this chapter.
- Chapter 3 “Data Processing and Transfer” shows the data processing flow and the arrays of the analyzer, describes how to access an internal data array (including trace data or calibration data). If you want to get measured trace data from the analyzer, read this chapter.

- Chapter 4 “Using Status Reporting System” describes the status reporting system of the analyzer and how to use it. This chapter also describes an SRQ interrupt. If you want to obtain the analyzer’s status using a BASIC program, read this chapter.
- Chapter 5 “Using the Trigger System” describes the SCPI trigger system and the procedures for using several types of sweeps and triggers. Read this chapter to learn how to use single or multiple sweeps.
- Chapter 6 “Using the I/O Port” provides information on how to use the I/O port on the rear panel. If you want to use the I/O port for communicating with an external instrument (such as a handler), read this chapter.
- Chapter 7 “Using the User Traces” describes how to use a user trace.
- Chapter 8 “Programming Miscellaneous” provides information not directly concerned with measurements, but useful for programming. This includes disk access, controlling Instrument BASIC, or debugging a program.
- Chapter 9 “Facilitating Program Execution and Utilizing Storage Devices” provides information on how to execute instrument BASIC programs easily, and on storage system relation to this function.
- Chapter 10 “Introducing HP instrument BASIC System” explains how to use 4291B’s Instrument BASIC function.
- Chapter 11 explains all the GPIB command functions and their syntax.
- Appendix A “Manual Changes” shows revision information for this manual.
- Appendix B provides the complex operation subprogram for the instrument BASIC which does not has the complex operation function.
- Appendix C lists all the GPIB commands sorted by function (key label). This list helps you to find the commands that are functionally equivalent to the key operations. Functions that are only available from GPIB and corresponding commands are also listed.
- Appendix D provides the Standard Commands for Programmable Instruments (SCPI) conformance information. All commands implemented in the analyzer are listed in this chapter.
- Appendix E helps you to understand about the measurement parameter setting commands by explaining the theory of the data formatting inside the analyzer.
- Error Messages lists all error messages with an explanation for each error.

How to Use the Program Modules

This manual provides many sample program modules that are not in a complete program style. You can easily understand the module's objective because the program module does not include unnecessary code. You can use these modules to build your own program by combining them.

The program modules are provided in the following style and typeface:

```
THIS IS A SAMPLE CODE.  This is a comment for a sample code.
```

As shown in the example above, a module has no line number, no initializing part, and no END statement. All these are required for an executable BASIC program.

Building a Working Program Using Program Modules

To make a program that uses sample program modules, perform the following steps:

1. Add an initializing module at the beginning of your program.
2. Arrange the program modules.
3. Add an END statement on the last line executed by your program.

The line numbers are added automatically by the BASIC editor.

Initializing Module

The initializing module defines a hardware identifier as a variable to eliminate the difference between Instrument BASIC and HP BASIC. Usually, you can use the same program for Instrument BASIC and HP BASIC by changing the initializing module. The initializing module also initializes an GPIB.

The following are the initializing modules for a program:

```
ASSIGN @Hp4291 TO 800  Assigning GPIB address to 800.
Scode=8                Assigning interface select code to 8.
ABORT Scode            Get active control.
CLEAR @Hp4291         Preset the interface.
```

Module 1-1. Initialize Module for Instrument BASIC

```
ASSIGN @Hp4291 TO 717  Assigning GPIB address to 717.
Scode=7                Assigning interface select code to 7.
ABORT Scode            Get active control.
CLEAR @Hp4291         Preset the interface.
```

Module 1-2. Initialize Module for the External Controller

Each module of this manual assumes that one of the initializing modules exists at the beginning of the program, and uses the following variables without notice:

@Hp4291 Represents the device selector of the 4291B. 800 is for Instrument BASIC and 717 is for the external controller.

Scode Represents the interface select code to which the 4291B is connected. 8 is for Instrument BASIC and 7 is for the external controller.

Example

For example, a complete program using Module 2-2 in Chapter 2 and Instrument BASIC, is shown below:

```
10 ASSIGN @Hp4291 TO 800 !
20 Scode=8 ! Module 1-1
30 ABORT Scode !
40 CLEAR @Hp4291 !
50 !
60 OUTPUT @Hp4291;"SYST:PRES;:INST CH2" ! Module 2-2
70 !
80 END
```

GPIB Overview

The GPIB is a general purpose digital interface system that is used to integrate the controller, measurement instruments, and peripherals into a system.

Controller

The controller is a device that can address an GPIB device to talk (output data) or listen (receive data).

The active controller can control the other devices on the bus at that time (when multiple controllers are connected). Only one controller can be active at a time. The active controller can pass control to another controller by using the `PASS CONTROL` command.

Only one controller can be a system controller on the same bus. The system controller is the active controller when the system is turned on. When another controller is the active controller, the system controller can become the active controller at any time by executing `ABORT select-code`.

Device Selector

GPIB device control is accomplished by sending commands from the active controller. The active controller can select the target device for the commands by specifying the device selector.

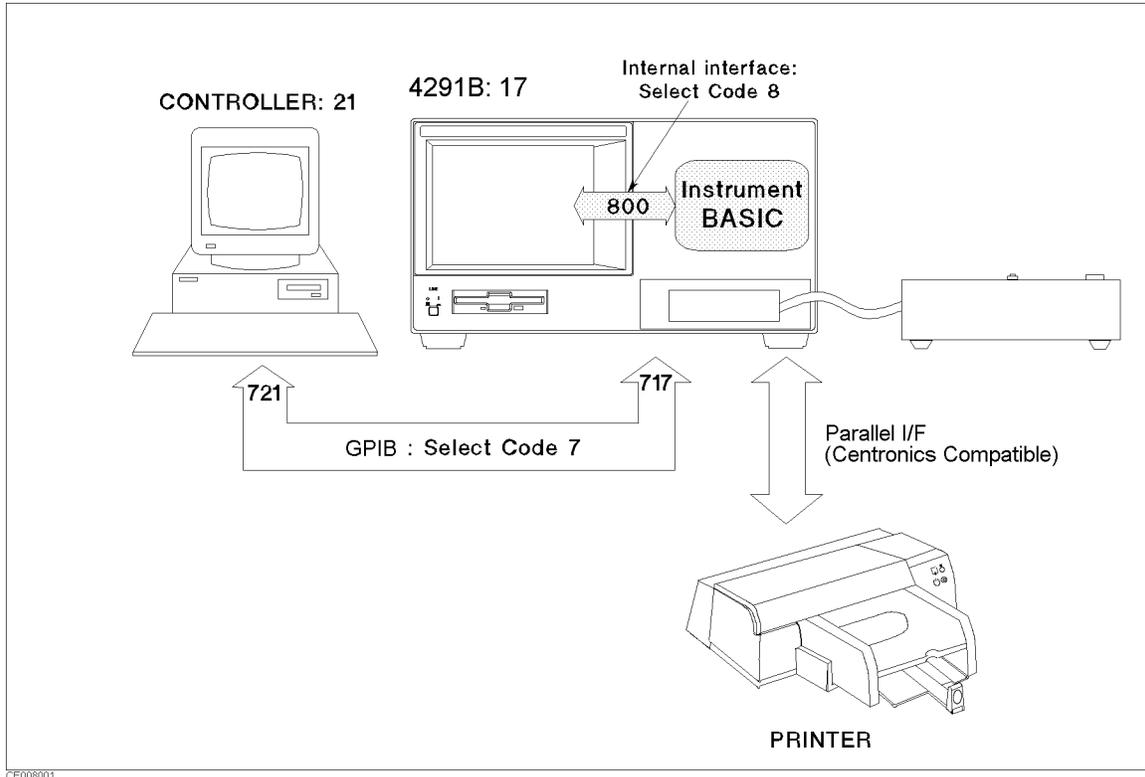


Figure 1-1. GPIB Device and Address

Figure 1-1 shows the relationship between the GPIB address and the device selector. For example, the device selector of the printer on GPIB with an address of "1," is "701" on the GPIB.

HP instrument BASIC is connected in the 4291B internally by the internal interface. The interface select code of the internal interface is "8" to distinguish it from the external select code of "7."

You can use any address from "00" to "30" to specify the internally connected analyzer from Instrument BASIC, because only the analyzer is connected on the internal interface. This manual uses address "00," thus the device selector is "800."

GPIB Commands

The analyzer is equipped with the GPIB remote programming digital interface.

The GPIB commands implemented in the 4291B are divided into the following three categories: common commands, instrument control commands, and simple commands.

The 4291B's GPIB commands conform to the Standard Commands for Programmable Instruments (SCPI). SCPI is the new instrument command language for controlling instruments that goes beyond IEEE 488.2 to address a variety of instrument functions in a standard manner.

Common Commands

Common commands are defined by the IEEE 488.2 standard. All common commands begin with an asterisk (*).

For example,

```
*CLS
```

Instrument Control Commands

Instrument control commands are defined by SCPI, and include all measurement functions and some general purpose functions. Instrument control commands consist of subsystems. Each subsystem is a set of commands that roughly corresponds to a functional block inside the instrument.

Instrument control commands have a hierarchical structure, called a **command tree**, that consists of several nodes separated by colons.

For example,

```
CALCulate:EVALuate:BAND:FULL
```

Simple Commands

Simple commands are analyzer-specific commands that conform to IEEE 488.2. Each simple command controls some measurement function that is normally programmed by sending multiple instrument control commands. To reduce the number of program lines and make the program simpler, these functions can also be executed by using a simple command instead of the multiple instrument control commands.

Note All GPIB commands implemented the analyzer are listed in Chapter 11.



Program Message Syntax

This section explains the construction of program messages. A **program message** is the message that you send from a computer to an instrument. Program messages consist of commands combined with appropriate punctuation and program message terminators.

Command Abbreviations

- Many instrument control commands have a long and a short form. The short form is obtained by deleting the lower case letters. The analyzer accepts both forms.

For example, the short form of :INITiate is :INIT and the long form of it is :INITIATE. (The analyzer does *not* accept anything in between, such as :INITIA.)

- Some commands have a numerical suffix. The numerical suffix can be omitted, and the analyzer recognizes that a numerical suffix of 1 is implied in this command.

For example in DISPlay:CMAP:COLor{1-14}, the numerical suffix is {1-14}. If you send DISP:CMAP:COL, it is recognized as DISP:CMAP:COL1 (the 1 is implied).

Upper and Lower Cases

Letter cases (upper and lower) are ignored.

Program Message Terminator

A program message must end with the **program message terminators**.

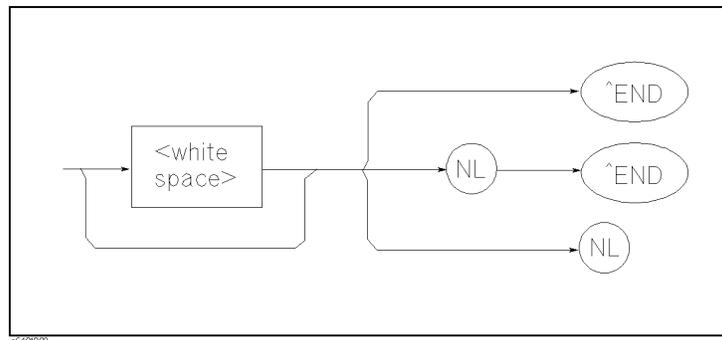


Figure 1-2. Program Message Terminators

<^END> means that End of Identify (EOI) is asserted on the GPIB interface at the same time the preceding data byte is sent.

The HP BASIC OUTPUT statement automatically sends program message terminators after the last data byte.

Multiple Messages

To send more than one command in the same message, you must separate them with a semicolon(;):

```
SENS:FREQ:STAR 100MAHZ;STOP 1GHZ
```

For more information, see “Command Tree and Compound Header Usage”, later in this chapter.

Query and Response Message Syntax

All commands can be queried except the commands described as “no query” in the command reference. To send a query message, add ? after the last command mnemonic.

```
SENS:FREQ:STAR?
```

A query response indicates the current setting of the analyzer. A response message may contain both commas and semicolons as separators. When a single query command returns multiple values, a comma is used to separate each data item. When multiple queries are sent in the same message, the group of data items corresponding to each query are separated by a semicolon. For example, the fictitious query QUERY1?;QUERY2? might return a response message of:

```
<data1>,<data1>;<data2>,<data2>
```

After the message, <New Line><^END> is always sent as a response message terminator.

Parameters

There must be a <white space>* between the last command mnemonic (SOUR:FREQ, in the example below) and the first parameter (100MAHZ, in the example below).

```
SOUR:FREQ 100MAHZ
```

If you send more than one parameter with a single command, each parameter must be separated by a comma.

```
DATA AOFF,2
```

Each command reference contains information about the parameters available for the individual commands. There are parameters that are spelled out (for example OFF, ON, “TR1”) or parameters shown as a word enclosed in < >, that represents some value.

* <white space> is a white space character (ASCII-encoded byte in the range of 00-09, 0B-20 (0-9, 11-32 decimal)) or a series of the white space characters.

Variable Types

The variable parameters used in GPIB commands are of three types: <numeric>, <string>, and <block>.

■ <numeric> represents numeric parameters as follows:

100	integer
100.0	fixed decimal point
1.0E6	floating decimal point
100.	fractional digits optional
-1.23, +235	leading signs allowed
-7.89e-01	use either E or e in exponentials
.5	digits to the left of the decimal point are optional

The analyzer accepts *<numeric>* parameters in various formats and responds to a particular query in a predefined and fixed format.*

The analyzer setting programmed with a numeric parameter can assume a finite number of values, so the analyzer automatically rounds off the parameter. For example, if you specified the OSC level as 501MV, it would be rounded off to 500MV.

* The *<numeric>* whose absolute value is less than 1000000 is returned in the fixed decimal point format (If the value is integer, the return format is integer).

The *<numeric>* whose absolute value is, or more than 1000000 is returned in the floating decimal point format.

□ Suffix

When a command has a specified suffix, the suffix multiplier and suffix units can be used with parameters as follows (the suffix multiplier must be used with the suffix unit):

Parameter	Suffix Unit	Available Multipliers
Frequency	HZ (Hz)	G : G (giga; $\times 10^9$)
Power	DBM (dBm)	MA : M (mega; $\times 10^6$)
Voltage	V (Volt)	K : k (kilo; $\times 10^3$)
Current	A (Ampere)	M : m (milli; $\times 10^{-3}$)
Impedance ¹	OHM (Ω)	U : μ (micro; $\times 10^{-6}$)
Admittance ²	SIE (Siemens)	N : n (nano; $\times 10^{-9}$)
Inductance	H (henry)	P : p (pico; $\times 10^{-12}$)
Capacitance	F (farad)	F : f (femto; $\times 10^{-15}$)
Time	S (second)	
Phase	DEG ($^\circ$; default), RAD (radian)	
distance	M (meter)	

¹ Resistance and reactance are the same

² Conductance or susceptance are the same

The suffix is optional and can be omitted. If you omit it, the analyzer assumes that the default suffix is sent.

- *<string>* is a string parameter that contains ASCII characters. A string must begin with a double quote (ASCII 34 decimal) and end with a double quotation mark. You can include the quotation mark as part of the string by typing it twice without any characters in between the quotation marks. Or, you can avoid typing the quotation marks twice by using a single quotation mark (ASCII 39 decimal).

Example of *<string>* TITLE,

```
OUTPUT @Meter;"DISP:TEXT10 'TITLE'"      using single quote
OUTPUT @Meter;"DISP:TEXT10 ""TITLE""""   using double quote
```

The quote to mark the beginning and end of the string is called the **delimiter**. The query response is the string with double quote delimiters.

- *<block>* is typically used to transfer large quantities of related data in binary format. *<block>* can be sent as the definite length blocks.

General form of block parameters:

```
#<num_digits><num_bytes><data bytes>
```

The single decimal digit* <num_digits> specifies how many digits are contained in <num_bytes>. The decimal number <num_bytes> specifies how many data bytes will follow in <data bytes>.

Example of <block> 1234567890

```
OUTPUT @Meter;"#2101234567890"
```

(2 means two digits follow, 10 means ten bytes follow.)

* Digit is an ASCII-encoded byte in the range of 30-39 (48-57 decimal).

Command Tree and Compound Header Usage

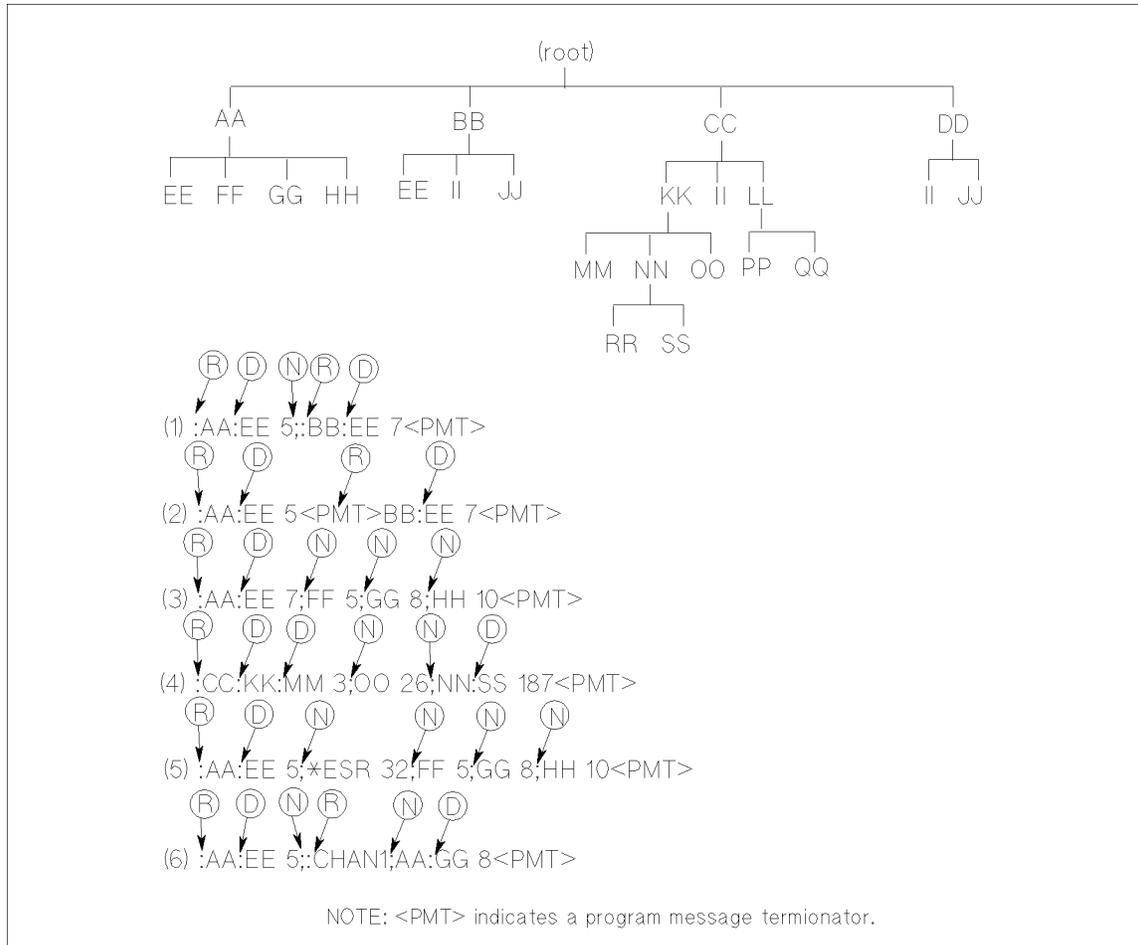
Figure 1-3 presents the instrument control command structure. The top of the command tree is called the **root**. To reach the low-level nodes, you must specify a particular path (like a DOS file directory path). Some of the notable aspects of this organization are:

1. The paths through the tree are not all the same length.
2. The number of sub-nodes under a node is not constant.
3. Node names are reused.

After Power ON or after presetting, the current path is set to the root. The path settings are changed as follows:

Program Message Terminator	A program message terminator (see page 1-4) sets the current path to the root.
Colon (:)	When a colon is placed between two command nodes, the colon moves the current path down one level on the command-tree. When the colon is the first character of a header, it specifies that the following command node is a root-level command.
Semicolon (;)	A semicolon separates two commands in the same message without changing the current path.
Common command	Common commands, such as *RST, are not part of the command tree. The parser interprets them in the same way, whatever the current path setting, and they have no effect on the current path.
Simple Command	Simple commands are root-level commands and they have no sub-node. The current path must be root, before and after the simple command is parsed.

Figure 1-3 shows examples of how to use the colon and semicolon to navigate efficiently through the command tree.



CG401001

Figure 1-3. Command Tree and Compound Header Usage

- Ⓡ sets the current path to the root.
- ⓓ moves the current path down one level.
- Ⓝ does not change the current path.

- (1) The leading colon before BB put the parser at the root.
- (2) This example works the same as example (1). A leading colon before BB is unnecessary because BB starts the parser at the root.
- (3) The entire path is not given before FF, GG, and HH.
- (4) OO 26 and NN:SS 187 are assumed to be prefixed by the implied prefix of the immediately previous command, :CC:KK.
- (5) The processing of common commands is unaffected by any previous compound commands. Also, the insertion of the common command does not affect the current path.
- (6) The current path must be set to the root prior to the simple command (in this example, CHAN1).

Preparation for Operation

This section describes GPIB cable connection and address setting for the GPIB control. You can choose one of the following two methods to control the analyzer:

- Using the Instrument BASIC controller.
- Using an external controller.

In both case, you can use the following procedure to prepare the controller and the analyzer:

1. Connect the GPIB cables.
2. Set the GPIB Address.
3. Prepare the controller for use

Using Instrument BASIC for Controller

The HP instrument BASIC system is a BASIC controller that is built in the analyzer. HP instrument BASIC is a subset of the HP BASIC.

You can control the 4291B internally by using the HP instrument BASIC. HP instrument BASIC has a capability to be a system controller. The other GPIB instruments are also controllable through the GPIB connector that is located on the rear panel of the 4291B.

1. Connecting the GPIB Cables

A connection between the analyzer and Instrument BASIC is not required because they are already connected via the internal interface in the analyzer. See Figure 1-1. However, if any other GPIB instruments must be connected, see “Connecting the GPIB Cables” on the next page.

2. Setting the GPIB Address

This is not required because the front panel setting of the GPIB address does not affect the internal interface. You can use any address to specify the analyzer via the internal interface as described in “Device Selector”.

3. Preparing Instrument BASIC

To set up Instrument BASIC, perform the following steps:

1. Connect the mini-DIN keyboard to the rear panel connector. (See the *Quick Start Guide*.)
2. Turn the analyzer power on.
3. To allocate the Instrument BASIC output area on the LCD for the PRINT statement, press **[Display]** **DISPLAY ALLOCATION HALF INST HALF BASIC**.
4. Press **[System]** **IBASIC** **EDIT** to open the Instrument BASIC editor.

For more information on how to use the Instrument BASIC editor, see Chapter 10.

If you connect the keyboard after turning on the analyzer, press **[Preset]** to enable key inputs.

Using an External Controller

You can use an external controller to control the analyzer. The analyzer has a GPIB interface on the rear panel. Connect the controller and the analyzer using a GPIB cable.

1. Connecting the GPIB Cables

Connect the analyzer and external instruments with GPIB cables. The GPIB connector is on the rear panel of the analyzer.

These are the rules for connecting GPIB cables:

- The total number of GPIB devices is up to 15 instruments.
- The total length of all the cables used is up to 20m or 2m for each instrument.

You can connect the GPIB cables in any configuration (linear, star, or combination), as long as the above rules are satisfied.

Note Do not use a screwdriver when connecting the GPIB cables. The screwdriver slots in the lock screws are provided for REMOVAL only.



2. Setting the GPIB Address

The analyzer has no hardware switch for setting the GPIB address. You can only set it by front panel operation. To change the GPIB address of the analyzer, perform the following steps:

1. Press **(Local)**.
2. Press **ADDRESS: 4291**.
3. Enter the new address by using the numerical keys. (Avoid duplication with the GPIB address of the external controller.)
4. Press **(x1)** to complete the operation.

When you want to control another 4291B, change the GPIB address to avoid duplication of addresses on the same bus.

When the analyzer receives any GPIB command from an external controller, the Rmt LED is turned on (above the **(Local)** key) to indicate the analyzer is in the remote mode. In remote mode, front panel key operation is disabled. To cancel the remote mode, press **(Local)**.

3. Preparing HP BASIC

To prepare HP BASIC for operation, see your HP BASIC system manual.

Sample Program Disk

The sample programs (not the program modules) in this manual are included on the furnished *Sample Program Disk* (Agilent part no. 04291-18020). This disk was formatted with MS-DOS Format.

Loading a Program from Disk

To use a sample program, load it into the Instrument BASIC and then run it.

1. Insert the sample program disk into the internal disk drive that is below the display.
2. Press **Display** **DISPLAY ALLOCATION BASIC STATUS**.
3. MSI ":INTERNAL"
4. Type, GET "*filename*" **Return**.

The applicable *filename* is printed in front of the sample programs in this manual.

Setup and Measurement Program

This chapter describes a basic measurement program that includes setups, compensating, triggering, and getting data. This chapter contains the following topics:

- Overview of GPIB Control
- Automating the Impedance Measurement Procedure
- Sample program: Basic Impedance Measurement Program
- Automating the Permittivity Measurement (Option 002 Only)
- Automating the Permeability Measurement (Option 002 Only)

Overview of GPIB Control

Before starting to program, you should know how to send an GPIB command to the analyzer. This section describes the following items:

- Sending GPIB commands.
- Sending a query and reading the response.

If you have experience programming GPIB instruments, you can skip this section and go to “Automating the Impedance Measurement Procedure”.

Sending GPIB Commands

HP BASIC and Instrument BASIC use the OUTPUT statement to send GPIB commands that control GPIB devices. For example:

```
OUTPUT @Hp4291; "SYSTEM:PRESET"
```

Module 2-1. Presetting the Analyzer

This line sends the GPIB command within the double quotes (SYSTEM:PRESET) to the GPIB device at address @Hp4291. This command presets the analyzer. This is equivalent to pressing the  key.

Note

The short form of SYSTem:PRESet is SYST:PRES. The short form of most commands is used in this manual. For an explanation and listing of all commands (including the short form) see the Chapter 11.

You can send multiple commands in a single line by separating each GPIB command with a semicolon (;).

```
OUTPUT @Hp4291;"SYST:PRES;:INST CH2"  Sends SYST:PRES and :INST CH2 by single line.
```

Module 2-2. Sending Multiple Commands in a Line

Reducing Keystrokes by Eliminating Node Repetition

When you are sending multiple consecutive commands from the same subsystem, for example:

```
SOUR2:VOLT:LEV 100mV
```

```
SOUR2:VOLT:STAT ON
```

You do not have to repeat the entire command string for each command. In place of the above example, you can send the second command by separating it with a semicolon (;) from the first command as shown below:

```
SOUR2:VOLT:LEV 100mV;STAT ON
```

This technique helps reduce the number of keystrokes when you are using several commands from the same subsystem.

Note



SCPI command does not correspond one-to-one to the key operations. Thus, to achieve the key operation, you must execute multiple commands. In that case, the analyzer provides a simple command that is equivalent one-to-one to the key operation. The simple command, however, is not compatible to the SCPI command. The correspondence chart of the key operation, SCPI command, and simple command is printed in the Appendix C.

This manual uses only SCPI commands.

Sending a Query and Reading the Response

There are commands that return a response after being sent. These commands are called queries. A query has a question mark (?) at the end of the command.

You can retrieve the response by using the ENTER statement as shown below:

```
OUTPUT @Hp4291;"INST:SEL?"  This line queries current channel setting.
ENTER @Hp4291;Ch$          This line retrieves the return string.
```

Module 2-3. Querying Active Channel Setting

You must retrieve the response into the correct type variable. In the example above, the query returns a string ("CH1" or "CH2"), depending on the current active channel setting. Therefore, the second line retrieves the response into a string type variable (Ch\$).

The response data type, numeric or string, for each command is shown in the reference section of the *GPIB Command Reference*.

Automating the Impedance Measurement Procedure

This section describes the sample program modules and equivalent commands for setting up the analyzer using the following functions:

1. Setting the active channel
2. Setting stimulus.
3. Performing calibration.
4. Setting port extension and electrical length.
5. Performing fixture compensation.
6. Setting measurement parameter.
7. Setting display format.
8. Setting dc bias.
9. Triggering a measurement.
10. Setting scale and reference.
11. Getting measured data to controller.

Figure 2-1 shows corresponding settings on the display of the analyzer to the above list.

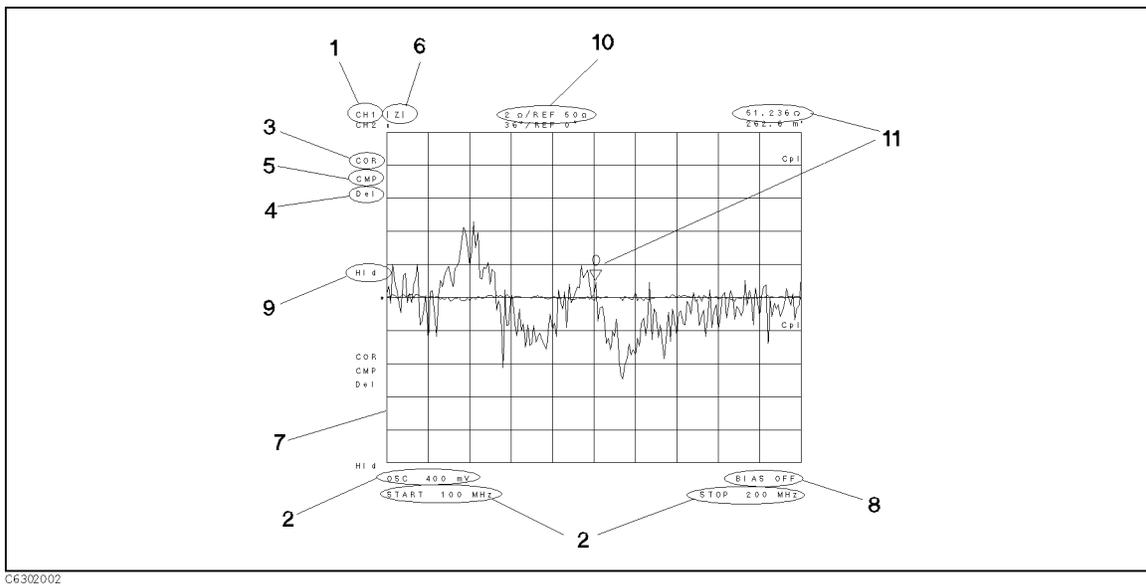


Figure 2-1. Setting Parameters

1. Setting the Active Channel

To begin setting up the analyzer, specify the active channel first because this affects all other settings.

```
OUTPUT @Hp4291;"INST CH1" Sets the active channel to channel 1.
```

Module 2-4. Setting the Measurement Mode and Active Channel

2. Setting Stimulus

The analyzer has four sweep sources. This section provides an example module for the following settings:

- Setting frequency sweep.
- Setting OSC level sweep.
- Setting dc voltage sweep. (Option 001 only)
- Setting dc current sweep. (Option 001 only)

Setting Frequency Sweep Range and Level

To use a frequency sweep source, you must turn off the OSC level sweep, dc voltage sweep, and dc current sweep.

OUTPUT @Hp4291;"SOUR1:VOLT:MODE FIX"	<i>OSC level sweep is off.</i>
OUTPUT @Hp4291;"SOUR2:VOLT:MODE FIX"	<i>dc voltage sweep is off.</i>
OUTPUT @Hp4291;"SOUR2:CURR:MODE FIX"	<i>dc current sweep is off.</i>
OUTPUT @Hp4291;"SENS:FREQ:MODE SWE"	<i>Frequency sweep is on.</i>
OUTPUT @Hp4291;"SENS:FREQ:STAR 100MHz;STOP 200MHz"	<i>Sets frequency range from 100 to 200 MHz.</i>

Module 2-5. Setting Frequency Sweep

To set the OSC level of the frequency sweep:

OUTPUT @Hp4291;"SOUR1:VOLT 400mV"	<i>Sets the OSC level to 400 mV.</i>
-----------------------------------	--------------------------------------

Module 2-6. Setting OSC Level

■ Related GPIB Commands

You can set the log frequency sweep mode by sending both of the following commands (the setting changes after both commands are executed):

SENS:SWE:SPAC LOG Selects log frequency sweep mode.
DISP:TRAC:X:SPAC LOG Sets the display to log format for x-axis.

To set back to the linear sweep:

SENS:SWE:SPAC LIN Selects linear frequency sweep mode.
DISP:TRAC:X:SPAC LIN Sets the display to linear format for x-axis.

Setting OSC Level Sweep

You can sweep by output level. To set an OSC level sweep:

OUTPUT @Hp4291;"SENS:FREQ:MODE FIX"	<i>Frequency sweep is off.</i>
OUTPUT @Hp4291;"SOUR2:VOLT:MODE FIX"	<i>dc voltage sweep is off.</i>
OUTPUT @Hp4291;"SOUR2:CURR:MODE FIX"	<i>dc current sweep is off.</i>
OUTPUT @Hp4291;"SOUR1:VOLT:MODE SWE"	<i>OSC level sweep is on.</i>
OUTPUT @Hp4291;"SOUR1:VOLT:STAR 0.5V;STOP 20V"	<i>Sets sweep range from 0.5 V to 20 V.</i>
OUTPUT @Hp4291;"SOUR2:FREQ 100MHz"	<i>Sets the CW frequency to 100 MHz.</i>

Module 2-7. Setting OSC Level Sweep

■ Related GPIB Commands

You can set the log sweep mode for OSC level sweep by sending both of the following commands (the setting changes after both commands are executed):

SOUR1:SWE:SPAC LOG Selects log OSC level sweep.
DISP:TRAC:X:SPAC LOG Sets the display to log format for x-axis.

To set back to the linear sweep:

SOUR1:SWE:SPAC LIN Selects linear OSC level sweep mode.
DISP:TRAC:X:SPAC LIN Sets the display to linear format for x-axis.

You can change the sweep direction by using one of the following commands:

SOUR1:SWE:DIR DOWN Sets the sweep direction from higher to lower level sweep.
SOUR1:SWE:DIR UP Sets the sweep direction from lower to higher level sweep.

Setting dc Voltage Sweep (Option 001 Only)

You can sweep by the dc bias level:

OUTPUT @Hp4291;"SENS:FREQ:MODE FIX"	<i>Frequency sweep is off.</i>
OUTPUT @Hp4291;"SOUR1:VOLT:MODE FIX"	<i>OSC level sweep is off.</i>
OUTPUT @Hp4291;"SOUR2:CURR:MODE FIX"	<i>dc current Sweep is off.</i>
OUTPUT @Hp4291;"SOUR2:VOLT:MODE SWE"	<i>dc voltage sweep is on.</i>
OUTPUT @Hp4291;"SOUR2:VOLT:STAR 0.5V;STOP 20V"	<i>Sets sweep range from 0 V to 20 V.</i>
OUTPUT @Hp4291;"SOUR2:FREQ 100MHz"	<i>Sets the CW frequency to 100 MHz.</i>
OUTPUT @Hp4291;"SOUR2:VOLT:STAT ON"	<i>Turns on dc bias.</i>

Module 2-8. Setting dc Voltage Sweep

■ Related GPIB Commands

You can set the log sweep mode for dc bias sweep by sending both of the following commands (the setting changes after both commands are executed):

SOUR2:SWE:SPAC LOG Selects log dc bias level sweep.
DISP:TRAC:X:SPAC LOG Sets the display to log format for x-axis.

To set back to the linear sweep:

SOUR2:SWE:SPAC LIN Selects linear dc bias sweep mode.
DISP:TRAC:X:SPAC LIN Sets the display to linear format for x-axis.

You can change the sweep direction by using one of the following commands:

SOUR2:SWE:DIR DOWN Sets the sweep direction from higher to lower level sweep.
SOUR2:SWE:DIR UP Sets the sweep direction from lower to higher level sweep.

Setting dc Current Sweep (Option 001 Only)

To sweep dc current level:

OUTPUT @Hp4291;"SENS:FREQ:MODE FIX"	<i>Frequency sweep is off.</i>
OUTPUT @Hp4291;"SOUR1:VOLT:MODE FIX"	<i>OSC level sweep is off.</i>
OUTPUT @Hp4291;"SOUR2:VOLT:MODE FIX"	<i>dc voltage sweep is off.</i>
OUTPUT @Hp4291;"SOUR2:CURR:MODE SWE"	<i>dc current sweep is on.</i>
OUTPUT @Hp4291;"SOUR2:CURR:STAR 0A;STOP 100mA"	<i>Sets sweep range from 0 A to 100 mA.</i>
OUTPUT @Hp4291;"SOUR2:FREQ 100MHz"	<i>Sets the CW frequency to 100 MHz.</i>
OUTPUT @Hp4291;"SOUR2:CURR:STAT ON"	<i>Turns ON the dc bias.</i>

Module 2-9. Setting dc Current Sweep

3. Performing Calibration

See "Sample Program: Performing Calibration" in Chapter 4 for the calibration program.

Checking Calibration State

To check if calibration is currently turned on:

OUTPUT @Hp4291;"SENS:CORR1:STAT?"	<i>Queries the current calibration status.</i>
ENTER @Hp4291;Stat	
IF NOT Stat THEN PRINT "Calibration was not performed!"	

Module 2-10. Checking Calibration State

4. Setting Port Extension and Electrical Length

You have to set the port extension to remove the phase shift error from the test head to the fixture electrode before performing a fixture compensation.

When a standard fixture of the analyzer is connected, you can set the length by specifying the fixture name as shown below:

```
OUTPUT @Hp4291;"SYST:FIXT HP16191" Sets the electrical length for the 16191A.
```

Module 2-11. Selecting the Connected Fixture

■ Related GPIB Command

To select a different fixture for the port extension setting:

SYST:FIXT HP16192	Set the electrical length for the 16192A.
SYST:FIXT HP16193	Set the electrical length for the 16193A.
SYST:FIXT UDEF	Set the electrical length for the user fixture. (See below.)

You can also set the port extension by using the following command:

SENS:CORR1:EDEL <i>value</i>	Set the electrical delay time due to the extension of the port. [s]
------------------------------	---

You can define your own fixture as a user fixture by entering the electrical length data and label. The following module defines the 16092A fixture as a user fixture:

```
OUTPUT @Hp4291;"SYST:FIXT:DIST 0.0034" Sets the electrical length for the 16092A.(0.34cm)
OUTPUT @Hp4291;"SYST:FIXT:LAB '16092A'" Sets label.
OUTPUT @Hp4291;"SYST:FIXT:SAVE" Saves the defined user fixture data into SRAM.
```

Module 2-12. Setting the User Fixture

5. Performing Fixture Compensation

Performing fixture compensation eliminates additional error from the test head to the electrode of fixture.

Before performing a fixture compensation, you have to note the following things:

- Calibration must be performed.
- dc bias must be turned off.
- Port extension is set if it required.

If user calibration is selected, you must select user compensation. The following module detects automatically which calibration mode is selected, then sets the user compensation if the user calibration is selected.

```
OUTPUT @Hp4291;"SENS:CORR1:COLL:FPO?" Queries current calibration mode is FIXed or USER.
ENTER @Hp4291;Cal_stat$ Retrieves a query response.
IF Cal_stat$="USER" THEN If user calibration is selected,
    OUTPUT @Hp4291;"SENS:CORR2:COLL:FPO USER" sets to the user compensation mode.
END IF
```

Module 2-13. Checking Calibration Mode (Fixed or User)

You can perform compensation for any combination of OPEN, SHORT, and LOAD. In this example, OPEN and SHORT compensations are performed by the following steps:

1. Set the interrupt for the completion of the compensation.
2. Perform the OPEN compensation.
3. Wait until the OPEN compensation is completed.
4. Perform the SHORT compensation.
5. Wait until the SHORT compensation is completed.
6. After the compensation sequence is completed, save the correction data.

When you execute the fixture compensation, you also have to note that the compensation is a series of processes that includes triggering, measuring OPEN/SHORT/LOAD, and calculating the compensation coefficients. It takes processing time to complete this sequence. To save the correction data, the compensation sequence must be completed. To confirm this, you must wait for the completion of the compensation process.

To verify that compensation processing is complete, you can use the SRQ interrupt method. Bit 8 of the instrument event status register represents the status of the compensation processing. For detailed information about the SRQ interrupt, see "SRQ and Interrupt" in Chapter 4.

OUTPUT @Hp4291;"TRIG:SOUR INT"	<i>Selects internal trigger source.</i>
!	
OUTPUT @Hp4291;"STAT:INST:ENAB 256"	<i>Sets to generate SRQ on the completion of</i>
OUTPUT @Hp4291;"*SRE 4"	<i>the compensation process.</i>
!	
ON INTR Scode GOTO Open_end	
OUTPUT @Hp4291;"*CLS;*OPC?"	<i>Clears status register; and confirm its</i>
ENTER @Hp4291;Opc	<i>completion.</i>
INPUT "Connect OPEN, then [x1]",Dummy\$	<i>Pauses program while connection.</i>
ENABLE INTR Scode;2	<i>Enables SRQ interrupt.</i>
OUTPUT @Hp4291;"SENS:CORR2:COLL STAN1"	<i>Executes OPEN measurement.</i>
Wait_open:GOTO Wait_open	<i>Waits for completion of measurement and calculation.</i>
Open_end:!	<i>When completed, the program jumps to this line.</i>
!	
ON INTR Scode GOTO Short_end	
OUTPUT @Hp4291;"*CLS;*OPC?"	
ENTER @Hp4291;Opc	
INPUT "Connect SHORT, then [x1]",Dummy\$	<i>Pauses program while connection.</i>
ENABLE INTR Scode;2	
OUTPUT @Hp4291;"SENS:CORR2:COLL STAN2"	
Wait_short:GOTO Wait_short	
Short_end:!	
!	
OUTPUT @Hp4291;"SENS:CORR2:COLL:SAVE"	<i>Completes the compensation and turn on.</i>

Module 2-14. Performing Fixture Compensation (OPEN and SHORT only)

Note



If the dc bias is turned on, you must turn off it before performing the fixture compensation. When in the fixture compensation process, dc bias must be turned off.

6. Setting Measurement Parameter

To set the measurement parameter, perform the following steps:

1. Set the conversion mode setting. (Off for the impedance measurement. On for the admittance or reflection coefficient measurement.)
2. Select the measurement parameter.

For example, select impedance as a parameter as follows:

```
OUTPUT @Hp4291; "CALC:MATH:STAT OFF"  Selects impedance. (No conversion)
OUTPUT @Hp4291; "CALC:FORM MLIN"      Selects |Z|.
```

Module 2-15. Setting Measurement Parameter

The measurement parameter, display format, and scale settings are processed after the measurement. Therefore, you can change these settings after the measurement is triggered. For example, when you need to measure more than two parameters, trigger once, then change the parameter from the present one to another parameter.

■ Related GPIB Command

□ Impedance

To select the impedance measurement parameter, execute the following command first:

```
CALC:MATH:STAT OFF          Select impedance by turning off the conversion.
```

Then select the parameter by using one of the following commands:

```
CALC:FORM MLIN             Impedance, |Z|
CALC:FORM REAL             Resistance, R
CALC:FORM IMAG             Reactance, X
CALC:FORM PHAS             Phase,  $\theta_z$ 
CALC:FORM CP               CP
CALC:FORM CS               CS
CALC:FORM LP               LP
CALC:FORM LS               LS
CALC:FORM D                D
CALC:FORM Q                Q
CALC:FORM RP               RP
CALC:FORM RS               RS
```

□ Admittance

To select the admittance measurement parameter, execute the following command first:

```
CALC:MATH:NAME ADM;STAT ON  Selects admittance.
```

Then select the parameter by using one of the following commands:

```
CALC:FORM MLIN             Admittance, |Y|
CALC:FORM REAL             Conductance, G
CALC:FORM IMAG             Susceptance, B
CALC:FORM PHAS             Phase,  $\theta_y$ 
```

□ Reflection Coefficients

To select the reflection coefficient parameter, execute the following command first:

```
CALC:MATH:NAME RCO;STAT ON  Selects reflection coefficient.
```

Then select the parameter by using one of the following commands:

CALC:FORM MLIN	Reflection Coefficient, $ \Gamma $
CALC:FORM REAL	Reflection Coefficient, $ \Gamma_x $
CALC:FORM IMAG	Reflection Coefficient, $ \Gamma_y $
CALC:FORM PHAS	Phase, θ_Γ

7. Setting Display Format

To set the display format, perform the following steps:

1. Select the grid to rectangle, Smith, polar, admittance, or complex plane.
2. Select parameter:
 - a. Select a linear or log scale for the rectangular grid.
 - b. If you select the Smith, polar, admittance chart, or complex plane, choose the complex format to include phase information.

The following example sets the format to log format:

```
OUTPUT @Hp4291;"DISP:TRAC1:GRAT:FORM RECT"  Selects the rectangular grid.
OUTPUT @Hp4291;"DISP:TRAC1:Y:SPAC LOG"      Sets the log scale.
```

Module 2-16. Setting a Log Format

■ Related GPIB Command

You can set the display format by sending the both of the listed commands for each format:

□ Linear format

```
DISP:TRAC1:GRAT:FORM RECT  Selects rectangular grid.
DISP:TRAC1:Y:SPAC LIN      Selects linear format.
```

□ Log format

```
DISP:TRAC1:GRAT:FORM RECT  Selects rectangular grid.
DISP:TRAC1:Y:SPAC LOG      Selects log format.
```

□ Polar chart

```
CALC:FORM COMP              Select complex format.
DISP:TRAC1:GRAT:FORM POL    Select polar chart grid.
```

□ Smith chart

```
CALC:FORM COMP              Select complex format.
DISP:TRAC1:GRAT:FORM SMIT   Select Smith chart grid.
```

□ Admittance chart

```
CALC:FORM COMP              Select complex format.
DISP:TRAC1:GRAT:FORM ADM    Select admittance chart grid.
```

□ Complex Plane

```
CALC:FORM COMP              Select complex.
DISP:TRAC1:GRAT:FORM CPL    Select complex plane grid.
```

8. Setting dc Bias (Option 001 Only)

When you use a frequency sweep or OSC level sweep, you can apply a dc bias to the DUT.

OUTPUT @Hp4291;"SOUR2:VOLT:ALC ON"	<i>Selects dc voltage bias.</i>
OUTPUT @Hp4291;"SOUR2:VOLT:LEV 10V"	<i>Sets dc bias voltage to 10 V.</i>
OUTPUT @Hp4291;"SOUR2:CURR:LEV 100mA"	<i>Sets current compliance to 100 mA.</i>
OUTPUT @Hp4291;"SOUR2:VOLT:STAT ON"	<i>Turns on the dc bias.</i>

Module 2-17. Setting dc Bias (Voltage)

■ Related GPIB Command

To set the dc current instead of dc voltage:

SOUR2:CURR:ALC ON Selects dc current bias.

SOUR2:CURR:LEV 10mA;STAT ON Sets dc bias current to 10 mA and turns on.

9. Triggering a Measurement

To trigger a measurement, perform the following steps:

1. Specify the trigger source.
2. Specify the number of times to sweep.
3. Set up and enable an interrupt. This is needed to wait for sweep completion.
4. Wait while connecting a DUT.
5. Trigger the measurement.
6. Wait until the sweep is completed to synchronize the program with the analyzer.

You may notice that the triggering the measurement is more complex than just sending a trigger command. The extra steps are required to synchronize the program with the analyzer's processing.

When you send a trigger command, the analyzer executes the trigger command and starts sweeping. When the analyzer gets ready to accept the next command, sweeping might not be finished because sweeping takes time. If you send a measured data query immediately after the trigger, the retrieved data will be invalid. Therefore, a program should detect the completion of the sweep before executing the next command. This is similar to the fixture compensation program module.

Use an SRQ interrupt to detect the completion of the sweep as shown in the following example module:

OUTPUT @Hp4291;"TRIG:SOUR INT"	<i>Selects an internal trigger as source.</i>
OUTPUT @Hp4291;"INIT:CONT OFF"	
OUTPUT @Hp4291;"ABOR"	<i>Moves to the wait for trigger state.</i>
OUTPUT @Hp4291;"SENS:SWE:COUN 1"	<i>Sets a single time sweep.</i>
ON INTR Scode GOTO Sweep_end	<i>Sets interrupt branch.</i>
OUTPUT @Hp4291;"STAT:INST:ENAB 1;*SRE 4;"	<i>Sets to generate an SRQ at the end of sweep.</i>
OUTPUT @Hp4291;"*CLS;*OPC?"	<i>Clears Status Register.</i>
ENTER @Hp4291;0pc	<i>Confirms the clear operation is completed.</i>
INPUT "Connect DUT, then [x1]",Dummy\$	<i>Pauses program while connection.</i>
ENABLE INTR Scode;2	<i>Enables the interrupt.</i>
OUTPUT @Hp4291;"INIT"	<i>Triggers the measurement.</i>
Waiting:GOTO Waiting	<i>Looping until interrupt occurs.</i>
Sweep_end:!	

Module 2-18. Making a Single Sweep Measurement

More trigger information is contained in Chapter 5. The SRQ information is in "SRQ and Interrupt" in Chapter 4.

■ Related GPIB Commands

The following commands set the point averaging factor and delay:

SENS: AVER1: COUN *value*; STAT ON Sets the point averaging factor.

SENS: SWE: DWEL1 *value*; DWEL1: AUTO ON Sets the point delay time.

10. Setting Scale and Reference

After triggering the measurement, you can change the scale setting to fit the trace in a display.

```
OUTPUT @Hp4291;"DISP:TRAC:Y:AUTO ONCE" Sets optimum scale setting.
```

Module 2-19. Automatic Scaling

11. Getting Measured Data to the Controller

After a measurement is completed, you probably want to get the measured data. To obtain a specific point's data, use the marker search function. You can get the data by performing the following steps:

1. Turn on the marker.
(You must turn on the marker first to use a marker function.)
2. Move the marker to a specified value.
3. Query the marker value to get the measured data to the controller.

The following example module gets a measured value at the specified frequency from the analyzer:

```
OUTPUT @Hp4291;"DISP:TRAC:MARK:ALL:STAT ON" Turns on the marker.
OUTPUT @Hp4291;"CALC:EVAL:Y1:XPOS 100MHz" Moves the marker to the specified frequency, 100 MHz.
OUTPUT @Hp4291;"CALC:EVAL:Y1:DATA?" Queries the measured data at the marker position.
ENTER @Hp4291;Mkr_value,Dummy,Mkr_f Retrieves the queried marker value to a variable.
```

Module 2-20. Getting Marker Measurement Data

■ Related GPIB Commands

The following commands are also used for controlling the marker.

- To clear the marker:

```
DISP:TRAC1:MARK1:ALL DEF Clears all markers from display.
```

- To use a marker search function:

```
CALC:EVAL:Y1:XPOS:MAX Searches for the maximum value.
```

```
CALC:EVAL:Y1:XPOS:MIN Searches for the minimum value.
```

```
CALC:EVAL:Y1:XPOS:TARG value Searches for the target value.
```

```
CALC:EVAL:Y1:XPOS:PEAK Searches for the peak.
```

The following commands return a marker value independently of the active channel setting:

```
CALC:EVAL:Y1:DATA? {CH1|CH2} Returns the marker value and its stimulus for channel 1 or 2, respectively.
```

You can also get the trace data train of the all measurement point instead of only the marker value by accessing the trace data array. See Chapter 3 for information on the internal data arrays and how to access them.

Sample Program: Basic Impedance Measurement Program

This section describes a complete setup and measurement program for the analyzer. This program is built by combining sample modules contained in this chapter. Use this sample program to learn the flow of a measurement program.

This program performs the following:

- Set frequency sweep range from 100 MHz to 200 MHz.
- Set fixture to 16191A.
- Perform and set OPEN and SHORT compensation.
- Set parameter to $|Z|$.
- Triggers a single sweep.
- Display readout impedance at 150 MHz on the display.

This sample program assumes that calibration was already performed. See the *User's Guide* for the calibration procedure.

Disk



This program is included in the *Sample Program Disk*. Its filename is BASIC. This program is initialized for Instrument BASIC. The program for the external controller, BASIC_E, is also included in the *Sample Program Disk*.

```

100 !**** INITIALIZE: MODULE 1-1 ****
110 ASSIGN @Hp4291 TO 800
120 Scode=8
130 ABORT Scode
140 CLEAR @Hp4291
150 OUTPUT @Hp4291;"DISP:ALL HIHB"
160 !
170 !**** CH & MEAS. MODE: MODULE 2-4 ****
180 OUTPUT @Hp4291;"SENS:FUNC ""IMP"";"
190 OUTPUT @Hp4291;"INST CH1"
200 !
210 !*** FREQUENCY SWEEP SETUP:MODULE 2-5 ***
220 OUTPUT @Hp4291;"SENS:FREQ:MODE SWE"
230 OUTPUT @Hp4291;"SOUR1:VOLT:MODE FIX"
240 OUTPUT @Hp4291;"SOUR2:VOLT:MODE FIX"
250 OUTPUT @Hp4291;"SOUR2:CURR:MODE FIX"
260 OUTPUT @Hp4291;"SENS:FREQ:STAR 100MHZ;STOP 200MHZ"
270 !
280 !**** FIXTURE SELECTION : MODULE 2-11 ****
290 OUTPUT @Hp4291;"SYST:FIXT HP16191"
300 !
310 !**** FIXTURE COMPENSATION: MODULE 2-14 ****
320 OUTPUT @Hp4291;"TRIG:SOUR INT"
330 !
340 OUTPUT @Hp4291;"STAT:INST:ENAB 256"
350 OUTPUT @Hp4291;"*SRE 4"
360 OUTPUT @Hp4291;"*CLS;*OPC?"
370 ENTER @Hp4291;0pc
380 !

```

```
390 ON INTR Scode GOTO Open_end
400 INPUT "CONNECT OPEN, THEN [RETURN]",A
410 ENABLE INTR Scode;2
420 OUTPUT @Hp4291;"SENS:CORR2:COLL STAN1"
430 Wait_open:GOTO Wait_open
440 Open_end:!
450 !
460 ON INTR Scode GOTO Short_end
470 OUTPUT @Hp4291;"*CLS;*OPC?"
480 ENTER @Hp4291;0pc
490 INPUT "CONNECT SHORT, THEN [RETURN]",A
500 ENABLE INTR Scode;2
510 OUTPUT @Hp4291;"SENS:CORR2:COLL STAN2"
520 Wait_short:GOTO Wait_short
530 Short_end:!
540 !
550 OUTPUT @Hp4291;"SENS:CORR2:COLL:SAVE"
560 !
570 !**** SET |Z| AS parameter: MODULE 2-15 ****
580 OUTPUT @Hp4291;"CALC:MATH:STAT OFF"
590 OUTPUT @Hp4291;"CALC:FORM MLIN"
600 !
610 !**** SET LIN FORMAT: MODULE 2-16 ****
620 OUTPUT @Hp4291;"DISP:TRAC1:GRAT:FORM RECT"
630 OUTPUT @Hp4291;"DISP:TRAC1:Y:SPAC LIN"
640 !
650 !**** TRIGGER: MODULE 2-18 ****
660 OUTPUT @Hp4291;"TRIG:SOUR INT"
670 OUTPUT @Hp4291;"INIT:CONT OFF"
680 OUTPUT @Hp4291;"ABOR"
690 OUTPUT @Hp4291;"SENS:SWE:COUN 1"
700 ON INTR Scode GOTO Sweep_end
710 OUTPUT @Hp4291;"STAT:INST:ENAB 1;*SRE 4"
720 OUTPUT @Hp4291;"*CLS;*OPC?"
730 ENTER @Hp4291;0pc
740 INPUT "CONNECT DUT, THEN PRESS [RETURN]",A
750 ENABLE INTR Scode;2
760 OUTPUT @Hp4291;"INIT"
770 Waiting:GOTO Waiting
780 Sweep_end:!
790 !
800 !**** AUTO SCALING: MODULE 2-19 ****
810 OUTPUT @Hp4291;"DISP:TRAC:Y:AUTO ONCE"
820 !
830 !**** MARKER FUNCTION: MODULE 2-20 ****
840 OUTPUT @Hp4291;"DISP:TRAC:MARK1:ALL:STAT ON"
850 OUTPUT @Hp4291;"CALC:EVAL:Y1:XPOS 150MHZ"
860 OUTPUT @Hp4291;"CALC:EVAL:Y1:DATA?"
870 ENTER @Hp4291;Mkr_value,Dummy,Mkr_f
880 PRINT TABXY(1,1),Mkr_value;"[OHM] ",TABXY(1,2),Mkr_f;"[Hz]"
890 !
900 END
```

Automating the Permittivity Measurement (Option 002 Only)

When Option 002 (Add Material Measurement Firmware) is installed in the analyzer, you can measure the permittivity parameters of the materials directly.

As shown below, the permittivity measurement procedure is similar to the impedance measurement procedure previously described. However, when performing permittivity measurements, the sample modules shown in bold are different from the impedance measurement procedure. This section describes the sample program modules that are different. You can build a control program by combining the sample program module in this section with the impedance measurement program.

1. Setting the active channel
2. Setting stimulus
3. Performing calibration
4. **Selecting fixture**
5. **Performing fixture compensation**
6. **Setting MUT thickness**
7. **Setting measurement parameter**
8. Setting display format
9. Setting dc bias
10. Triggering a measurement
11. Setting scale and reference.
12. Getting measured data to controller

4. Selecting Fixture

To make a permittivity measurement, you must select a permittivity fixture as shown below:

```
OUTPUT @Hp4291;"SYST:FIXT HP16453" Selects the 16453A.
```

Module 2-21. Selecting the Connected Fixture

5. Performing Fixture Compensation

In the permittivity measurement, you must perform all OPEN/SHORT/LOAD compensations at the permittivity fixture electrodes. For the LOAD compensation, the analyzer must know the LOAD's thickness and permittivity. The permittivity of the LOAD furnished with the 16453A dielectric material fixture is preset, but the thickness must be entered manually.

```
OUTPUT @Hp4291;"SENS:CORR2:CKIT2:STAN6:THIC 3.01mm" Setting LOAD thickness.
OUTPUT @Hp4291;"SENS:CORR2:CKIT2:SAVE" Storing data into a non-volatile memory.
OUTPUT @Hp4291;"SENS:CORR2:CKIT2 UDEF" Selects the defined calibration kit.
```

Module 2-22. Modifying Permittivity Compensation Kit

The fixture compensation procedure is similar to the impedance measurement, the difference is that it requires all OPEN/SHORT/LOAD compensations.

OUTPUT @Hp4291;"TRIG:SOUR INT"	<i>Selects internal trigger source.</i>
!	
OUTPUT @Hp4291;"STAT:INST:ENAB 256"	<i>Sets to generate SRQ on the completion of the compensation process.</i>
OUTPUT @Hp4291;"*SRE 4"	
!	
ON INTR Scode GOTO Open_end	
OUTPUT @Hp4291;"*CLS;*OPC?"	<i>Clears status register; and confirm its completion.</i>
ENTER @Hp4291;Opc	
INPUT "Set electrode to maximum distance, then [x1]",Dummy\$	<i>Pauses program while setting electrode to maximum distance.</i>
	<i>Enables SRQ interrupt.</i>
ENABLE INTR Scode;2	<i>Executes OPEN measurement.</i>
OUTPUT @Hp4291;"SENS:CORR2:COLL STAN4"	<i>Waits for completion of measurement and calculation.</i>
Wait_open:GOTO Wait_open	
	<i>When completed, the program jumps to this line.</i>
Open_end:!	
!	
ON INTR Scode GOTO Short_end	
OUTPUT @Hp4291;"*CLS;*OPC?"	
ENTER @Hp4291;Opc	
INPUT "Contact Electrodes, then [x1]",Dummy\$	
ENABLE INTR Scode;2	
OUTPUT @Hp4291;"SENS:CORR2:COLL STAN5"	
Wait_short:GOTO Wait_short	
Short_end:!	
!	
ON INTR Scode GOTO Load_end	
OUTPUT @Hp4291;"*CLS;*OPC?"	
ENTER @Hp4291;Opc	
INPUT "Set Teflon Sheet, then [x1]",Dummy\$	<i>Pauses program while inserting Teflon sheet.</i>
ENABLE INTR Scode;2	
OUTPUT @Hp4291;"SENS:CORR2:COLL STAN6"	
Wait_load:GOTO Wait_load	
Load_end:!	
!	
OUTPUT @Hp4291;"SENS:CORR2:COLL:SAVE"	<i>Completes the compensation and turn on.</i>

Module 2-23. Performing Compensation (Permittivity)

6. Setting MUT Thickness

The analyzer uses a thickness of the MUT to calculate the permittivity. You must enter the MUT thickness manually before selecting the permittivity as a measurement parameter.

OUTPUT @Hp4291;"CALC:MATH1:DIM1 1.5mm" <i>Sets MUT thickness.</i>

Module 2-24. Setting MUT Thickness

7. Setting Measurement Parameter

To select a permittivity parameter, perform the following steps:

1. Turn the permittivity conversion on.
2. Select the measurement parameter.

For example, select the permittivity as follows:

OUTPUT @Hp4291;"CALC:MATH:NAME DCO;STAT ON"	<i>Selects the permittivity conversion.</i>
OUTPUT @Hp4291;"CALC:FORM REAL"	<i>Selects ϵ_r'.</i>

Module 2-25. Setting Measurement Parameter (Permittivity)

■ Related GPIB Commands

□ Permittivity

To select the permittivity measurement parameter, execute the following command first:

CALC:MATH:NAME DCO;STAT ON *Selects permittivity.*

Then select the parameter by using one of the following commands:

CALC:FORM MLIN	Absolute value of complex dielectric constant, $ \epsilon_r $
CALC:FORM REAL	Dielectric constant, ϵ_r' (real part of complex dielectric constant)
CALC:FORM LFAC	Dielectric loss index, ϵ_r'' (imaginary part of complex dielectric constant)
CALC:FORM LTAN	Dielectric dissipation factor, $\tan\delta (= \frac{\epsilon_r''}{ \epsilon_r' })$

Cole-Cole Plot

Plotting a trace of ϵ_r' on X axis and ϵ_r'' on Y axis is called a Cole-Cole plot that is used to analyze a evaluation of the hi-polymer materials. To display the Cole-Cole plot, use the following:

OUTPUT @Hp4291;"CALC:MATH:NAME DCO;STAT ON"	<i>Turns on the dielectric constant conversion.</i>
OUTPUT @Hp4291;"CALC:FORM MLIN"	<i>Selecting ϵ is ignored in the complex plane.</i>
OUTPUT @Hp4291;"CALC:FORM COMP"	<i>Select complex form.</i>
OUTPUT @Hp4291;"DISP:TRAC1:GRAT:FORM CPL"	<i>Select complex plane grid for displaying Cole-Cole plot.</i>

Module 2-26. Displaying Cole-Cole Plot

Automating the Permeability Measurement (Option 002 Only)

When Option 002 (Add Material Measurement Firmware) is installed in the analyzer, you can measure the permeability parameters of the magnetic materials directly.

As shown below, the permeability measurement procedure is similar to the impedance measurement procedure previously described. However, when performing permeability measurements, the sample modules shown in bold are different from the impedance measurement procedure. This section describes the sample program modules that are different. You can build a control program by combining the sample program module in this section with the impedance measurement program.

1. Setting the active channel
2. Setting stimulus
3. Performing calibration
4. **Selecting fixture**
5. **Performing fixture compensation**
6. **Setting MUT size**
7. **Setting measurement parameter**
8. Setting display format
9. Setting dc bias
10. Triggering a measurement
11. Setting scale and reference.
12. Getting measured data to controller

4. Selecting Fixture

To make a permeability measurement, you must select a permeability fixture as shown below:

```
OUTPUT @Hp4291;"SYST:FIXT HP16454S"  Sets the electrical length for the 16454A (Small).
```

Module 2-27. Selecting the Connected Fixture

■ Related GPIB Command

To select other fixture, use the following command:

```
SYST:FIXT HP16454L          Selects 16192A (Large).
```

5. Performing Fixture Compensation

For the permeability measurement, you only have to perform the SHORT compensation. You can set only the spacer without MUT in the magnetic material test fixture.

OUTPUT @Hp4291;"TRIG:SOUR INT"	<i>Selects internal trigger source.</i>
!	
OUTPUT @Hp4291;"STAT:INST:ENAB 256"	<i>Sets to generate SRQ on the completion of the compensation process.</i>
OUTPUT @Hp4291;"*SRE 4"	
!	
ON INTR Scode GOTO Short_end	
OUTPUT @Hp4291;"*CLS;*OPC?"	<i>Clears status register; and confirm its completion.</i>
ENTER @Hp4291;0pc	
INPUT "Set Spacer w/o MUT, then [x1]", Dummy\$	<i>Pauses program while setting SHORT condition.</i>
ENABLE INTR Scode;2	<i>Enables SRQ interrupt.</i>
OUTPUT @Hp4291;"SENS:CORR2:COLL STAN7"	<i>Executes SHORT measurement.</i>
Wait_short:GOTO Wait_short	<i>Waits for completion of measurement and calculation.</i>
Short_end:!	<i>When completed, the program jumps to this line.</i>
!	
OUTPUT @Hp4291;"SENS:CORR2:COLL:SAVE"	<i>Completes the compensation and turn on.</i>

Module 2-28. Performing Compensation (Permeability)

6. Setting MUT Size

To determine the permeability of the MUT, the analyzer uses the MUT (Material Under the Test) effective magnetic path length and effective cross-sectional area. The analyzer calculates these parameters from an inner length, outer length, and height of the MUT.

OUTPUT @Hp4291;"CALC:MATH1:DIM2 4mm,8mm,3mm"	<i>Sets the MUT size. (Inner, Outer, Height)</i>
--	--

Module 2-29. Setting MUT Size

7. Setting Measurement Parameter

To select a permeability parameter, perform the following steps:

1. Turn the permeability conversion on.
2. Select the measurement parameter.

For example, select the permeability as follows:

OUTPUT @Hp4291;"CALC:MATH1:NAME PER;STAT ON"	<i>Selects the permeability conversion.</i>
OUTPUT @Hp4291;"CALC:FORM MLIN"	<i>Selects μ.</i>

Module 2-30. Setting Measurement Parameter (Permeability)

■ Related GPIB Commands

□ Permeability

To select the permeability measurement parameter, execute the following command first:

CALC:MATH:NAME DCO;STAT ON Selects permeability.

Then select the parameter by using one of the following commands:

CALC:FORM MLIN	Absolute value of complex permeability, $ \mu $
CALC:FORM REAL	Real part of complex permeability, μ'
CALC:FORM LFAC	Imaginary part of complex permeability, μ''
CALC:FORM LTAN	$\tan\delta (= \frac{\mu''}{\mu'})$

Data Processing and Transfer

This chapter describes data processing and how to access the internal data arrays.

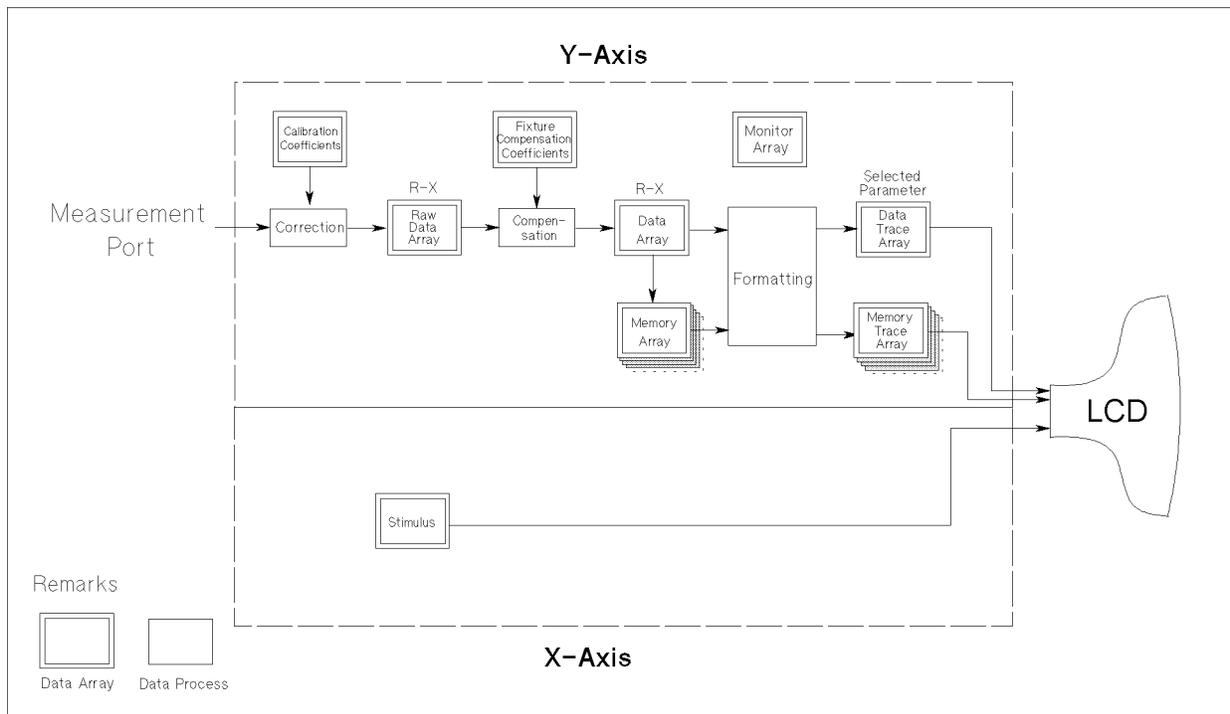
The following information is covered in this chapter:

- Data arrays
- Data transfer method
- Sample program: compensation data transfer

Data Arrays

The analyzer has data arrays that contain the measurement data, error correction data, and stimulus data. You can read or write data to these arrays using GPIB commands.

Figure 3-1 shows a simplified diagram of the data processing in the analyzer:



C6303001

Figure 3-1. Simplified Data Processing Flow

Double lined boxes represents data arrays that hold intermediate or processed data. The following section describes each of these data arrays.

Raw Data Array

The raw data array stores the results of all the preceding data processing operations including the correction by calibration data. The analyzer measures the voltage and current of a DUT and converts them to impedance. The raw data array stores this converted impedance value in a complex form (R and X). When you want to use your own compensation method for a measurement data, 1) take out the raw data from the raw data array (see module 3-1), 2) apply your compensation method to the taken out data, 3) enter compensated data into the data array (module 3-2).

The following example module queries for raw data and retrieves it:

```
DIM Dat(1:201,1:2)      Define NOPx2 for receiving complex data.
OUTPUT @Hp4291;"DATA? RAW"  Query raw data array.
ENTER @Hp4291;Dat(*)
```

Module 3-1. Getting Raw Data Array

■ Related GPIB Commands

The following command is used for setting data to the raw data array:

DATA RAW, *data* Sends data to the raw data array of the active channel.

Data Array

The results of error correction are stored in the data array as a complex form of impedance (R and X). The error correction process reflects the port extension and fixture compensation for the raw data array.

When you want impedance data in a complex format, use this array.

The following example module sets data for the data array:

```
OUTPUT @Hp4291;"DATA DATA, ";Dat(*)  Sets data to data array.
```

Module 3-2. Setting Data Array Data

■ Related GPIB Commands

The following command is used to query the data array data:

DATA? DATA Queries data in data array of the active channel.

Data Trace Array

This is the array for the data being displayed. All post processing functions; formatting, parameter setting, and data math are processed for the impedance data of the data array and stored in this array. The units of the array readout depend on the current display paramter. This data is the same data as listed by [Copy](#) [MORE LIST VALUES](#).

This data trace array is generally the most useful because it is the same information as that seen on the display. When you want to use data with a selected parameter unit, use this array.

The data trace array data is scalar data unless the Smith chart, polar chart, admittance chart, or complex plane is selected as the display format. Then the data trace array holds data in a complex form. Therefore, the imaginary part of the trace data array contains all "0".

3-2 Data Processing and Transfer

If the Smith chart, polar chart, admittance chart, or complex plane is selected, the data trace array data is complex data. Then the array contains complex numbers.

The following example module queries the data trace array and retrieves it:

```
DIM Dat(1:201,1:2)      Define NOPx2 for receiving complex data.
OUTPUT @Hp4291;"TRAC? DTR" Query data trace array.
ENTER @Hp4291;Dat(*)   TRAC? outputs scalar data in a complex format. Imaginary part is all 0.
```

Module 3-3. Getting Data Trace Array

■ Related GPIB Commands

The following commands are used for accessing the data trace array:

TRAC? DTR Outputs data trace array of the active channel.

TRAC? MTR Outputs memory trace array of the active channel.

Using the following commands, you can access a data trace array without depending on an active channel setting. These are useful to get data from both channels without altering the active channel.

TRAC? DTRCH1 Outputs data trace array of channel 1.

TRAC? DTRCH2 Outputs data trace array of channel 2.

Calibration Coefficient Array

The calibration process accessed by pressing **[Cal] CALIBRATE MENU**, measures the calibration standards and calculates data for the calibration coefficients A_1 , B_1 , and C_1 that are stored into this array. These coefficients are used for error correction at the test head. Each array corresponds to a specific error term in the error model.

The analyzer uses the following equation to calibrate a measured data at the test head port:

$$Z_{cal} = A_1 \times \frac{Z_m - B_1}{1 - Z_m \times C_1}$$

Where,

Z_{cal} Calibrated impedance

Z_m Measured impedance

A_1, B_1, C_1 Calibration coefficients

For more information about the calibration and error model, see Chapter 11 of the *Operation Manual*.

The analyzer uses three different sets of the calibration coefficients for the measurement range and oscillator level. The calibration coefficients are stored in the following arrays:

Table 3-1. Calibration Coefficients

Coefficient	Normal-vl Range	Normal-vh Range	Expand Range
A_1	CCO11	CCO21	CCO31
B_1	CCO12	CCO22	CCO32
C_1	CCO13	CCO23	CCO33

All coefficients are required for normal operation.

Accessing Arrays

If you want to enter calibration data from the controller to the calibration coefficient array, the analyzer must have previously done the calibration to enable the calibration data. To do this, perform the following steps:

1. Execute a dummy calibration to validate the correction.
2. Send the new calibration coefficients.

Note that the calibration coefficients are a complex data.

■ Related GPIB Commands

The following command is used for accessing the calibration coefficient array:

DATA(?) CC0{11|12|13|21|22|23|31|32|33} Sets (Outputs) the specified calibration coefficient array data of the active channel.

Compensation Coefficient Array

The fixture compensation process accessed by pressing **Cal** **FIXTURE COMPEN**, measures the OPEN, SHORT, and/or LOAD and calculates data for compensation coefficients A_3 , B_3 , and C_3 that are stored into the compensation coefficient array. Therefore, the fixture compensation array has three arrays.

The fixture compensation coefficients, A_3 , B_3 , and C_3 depend on the combination of OPEN, SHORT, and LOAD (ON or OFF for each). The analyzer calculates three coefficients using different equations for the combination of OPEN/SHORT/LOAD (on/off). If you change the combination of OPEN, SHORT, or LOAD (on/off), the analyzer calculates the coefficients again using the measured OPEN, SHORT, and/or LOAD data and sets new A_3 , B_3 , and C_3 coefficients into the compensation coefficient arrays.

The analyzer uses the following equation to compensate calibrated data after port extension:

$$Z_{comp} = A_3 \times \frac{Z_{port} - B_3}{1 - Z_{port} \times C_3}$$

Where,

Z_{comp} Compensated impedance

Z_{port} Impedance after corrected a port extension to the calibration data

A_3, B_3, C_3 Compensation coefficients

For more information on how the data is used to compensate for errors, see the *Operation Manual*.

Accessing Arrays

If you want to use custom compensation data by sending correction data to the fixture compensation array from the controller, you have to turn on the fixture compensation by performing the fixture compensation process. The CMP (or Cmp) notation shows that compensation is activated. To do this, perform the following steps:

1. Execute a dummy compensation either OPEN, SHORT, or LOAD to turn on it.
2. Set new compensation data.

Note that the fixture compensation coefficients are complex data.

Note

After setting a new compensation data, do not change the compensation settings, stimulus setting, or test head setting. Otherwise the analyzer recalculates the fixture compensation coefficients and overwrites the arrays.

- Related GPIB Commands

The following commands are used for accessing the compensation data arrays:

DATA? CMP{1|2|3} Outputs coefficients A_3 , B_3 , and C_3 data, respectively.
 DATA CMP{1|2|3}, data Enters coefficients A_3 , B_3 , and C_3 data, respectively into the fixture compensation array.

Monitor Array

The monitor array stores a level monitor value for all measurement points. The monitored level data is stored when the level monitor function is turned on and has executed a sweep.

This array is read-only.

To get the level monitor data:

```
DIM Monitor(1:201)            Defines NOP size array for receiving data.
OUTPUT @Hp4291;"CALC:EVAL:ON2 'ACV'"    Turning on the level monitor; "AC-V".
:
: Insert Module 2-18 here.
:
OUTPUT @Hp4291;"DATA? MON"            Queries stimulus array.
ENTER @Hp4291;Monitor(*)
```

Module 3-4. Getting Level Monitor Data

- Related GPIB Commands

The following commands are used for accessing the level monitor data array.

CALC:EVAL:ON2 "ACC" Monitors ac current level.
 CALC:EVAL:ON2 "DCV" Monitors dc voltage level.
 CALC:EVAL:ON2 "DCC" Monitors dc current level.
 CALC:EVAL:ON2 "OFF" Turn off the level monitor.

Stimulus Array

The stimulus array contains the stimulus (x-axis) value for all displayed points.

```
DIM Stim(1:201)            Define NOP size array for receiving data.
OUTPUT @Hp4291;"DATA? SPAR"    Query stimulus array.
ENTER @Hp4291;Stim(*)
```

Module 3-5. Getting Stimulus Array

Arrays for Memory Trace

When you store the trace data into the trace memory by sending the `TRAC: COPY TR2, TR1` command or by pressing `DATA` → `MEMORY`, the data array data is stored into the memory array. Memory array data passes through the formatting process, and then is stored into the memory trace array that is being displayed on the LCD. See Figure 3-1.

You can display up to 16 memory traces at a time for each channel. The front-panel operation for the memory trace is changed to the single memory trace activated by pressing `Display` `DEFINE TRACE` `SELECT MEMORY NO.`

Besides the 16-trace limitation, the number of traces that can be stored into trace memory at one time depends on the capacity of the system memory and the number of points in the traces. The analyzer always reserves memory for 3 traces (up to 801 points). The remaining memory is shared with the Instrument BASIC graphics. To maximize the memory area for memory traces, disable the Instrument BASIC graphics by toggling `Display` `DISPLAY ALLOCATION GRAPHICS: BASIC DRAM` to `ALL MEMORY TRACE` or sending the GPIB command: `DISP: GRAP: STAT OFF`.

Accessing Memory Array

You can only read data for the memory array that is activated. The memory array is read only. The following example module reads data from the memory array:

```
DIM Dat(1:201, 1:2)
OUTPUT @Hp4291;"DATA? MEM"  Queries an activated memory array.
ENTER @Hp4291;Dat(*)        Receives the query response.
```

Module 3-6. Getting Memory Array Data

Accessing Memory Trace Array

You can read or write memory trace array data. The following example module sets data for the memory trace array. You have to execute a trace-to-memory store procedure before setting a memory to display:

```
OUTPUT @Hp4291;"TRAC: COPY TR2, TR1"  Store trace data into trace memory to allocate a memory trace area.
OUTPUT @Hp4291;"TRAC TR2, ";Dat(*)    Sets data into memory trace 1.
OUTPUT @Hp4291;"DISP: TRAC2: STAT ON" Display a memory trace 1.
```

Module 3-7. Setting Memory Trace Array Data

■ Related GPIB Commands

The following commands are used to query a contents of the memory trace array:

```
TRAC? TR{2-17}           Reads the memory trace of the active channel.
TRAC? TR{2-17}CH{1-2}   Reads the memory trace of the specified channel without
                          changing the active channel.
```

Data Transfer Methods

This section describes the data transfer methods. When you get or send the data array's data, there are two methods for data transfer: ASCII and binary.

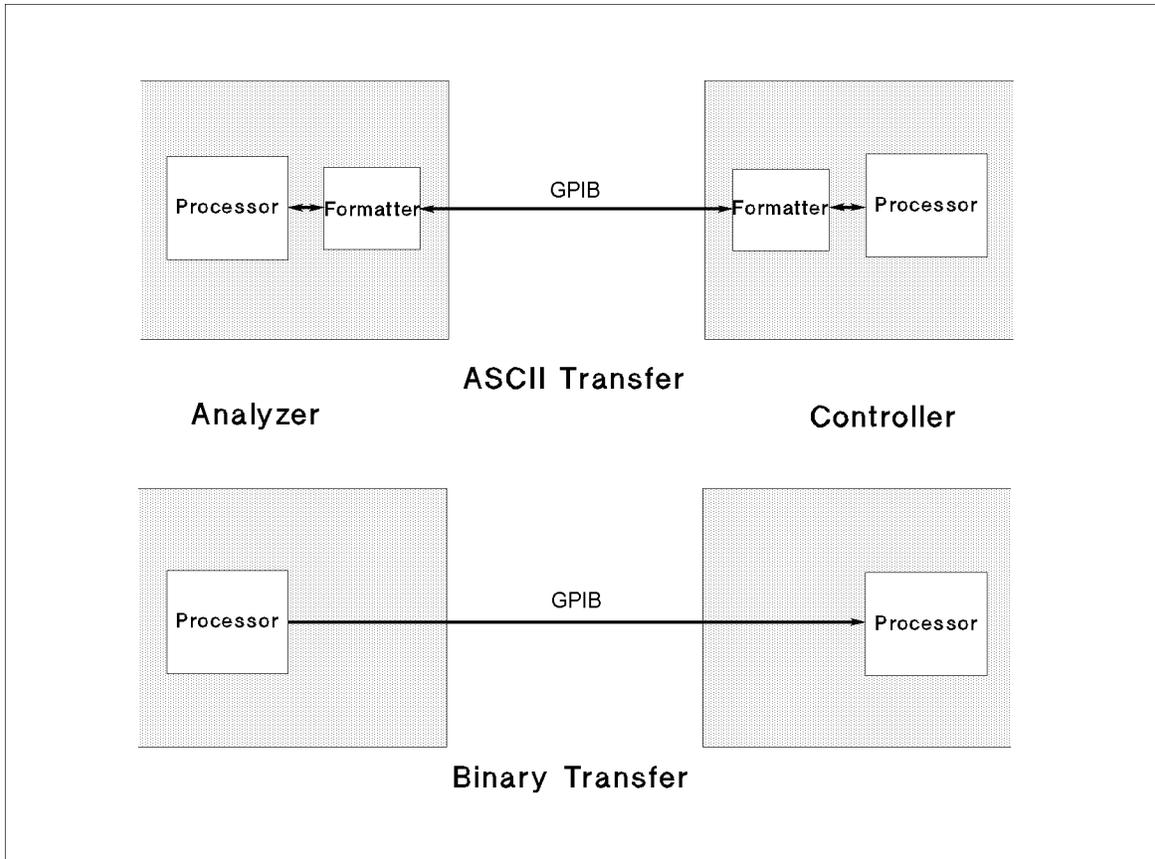


Figure 3-2. Simplified Internal Process of ASCII and Binary Transfer

Because the ASCII transfer passes through the formatted process, the program does not care about the data format. On the other hand, the binary transfer directly passes the data, but you have to indicate what data is transferred using the data header. The binary transfer is faster than the ASCII transfer.

ASCII Transfer

The ASCII format transfer is the easiest way to transfer array data between the analyzer and the controller. You do not have to worry about the data format because the analyzer and the controller automatically handles the formatting of the transferred data in this format. You can just send or retrieve array data by using the OUTPUT and ENTER statements.

The ASCII transfer format is sent as a 14-character (data) or 22-character (stimulus) string for each data point. This string includes a digit, sign or decimal point. Therefore, the data length of 201 points of complex data is 6030 bytes. (Including data delimiter “LF” for each data.)

To retrieve data from the analyzer using the ASCII format transfer, the following procedure is used:

1. Define a data array that is the same size as the data to be retrieved.

2. Specify the data transfer format is ASCII.
3. Send the data query command.
4. Retrieve the data.

DIM Dat (1:201, 1:2)	<i>Define the data array for receiving.</i>
OUTPUT @Hp4291;"FORM:DATA ASC"	<i>Specify the ASCII format.</i>
OUTPUT @Hp4291;"TRAC? DTR"	<i>Query the data trace.</i>
ENTER @Hp4291;Dat(*)	<i>Retrieve the data.</i>

Module 3-8. Retrieving Data from the Analyzer Using ASCII Transfer

Sending data to the analyzer is easy, just specify a format, then send the data:

OUTPUT @Hp4291;"FORM:DATA ASC"	<i>Specify the ASCII format.</i>
OUTPUT @Hp4291;"TRAC DTR, ";Dat(*)	<i>Send command and data.</i>

Module 3-9. Sending Data to Analyzer by ASCII Transfer

Binary Transfer

For a faster data transfer, use the binary format. There are three formats for binary transfer. The following list shows the data format that the analyzer outputs when you query the data:

■ IEEE 64-bit Floating Point Format

Figure 3-3 shows the data transfer format of IEEE 64-bit floating point format. Data is stored internally in the 200/300 series computer with the IEEE 64-bit floating point format, eliminating the need for any reformatting by the computer. In this mode, each number takes 8 bytes.

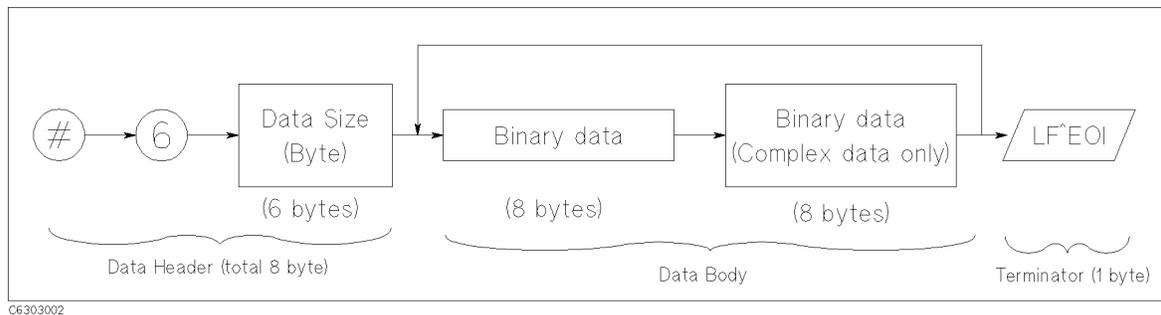


Figure 3-3. IEEE 64-bit floating point format

■ IEEE 32-bit Floating Point Format

Figure 3-4 shows the data transfer format scheme. In this mode, each data point is 4 bytes. The difference from the 64-bit format is a significant digit. The 64-bit format has double the precision of this format.

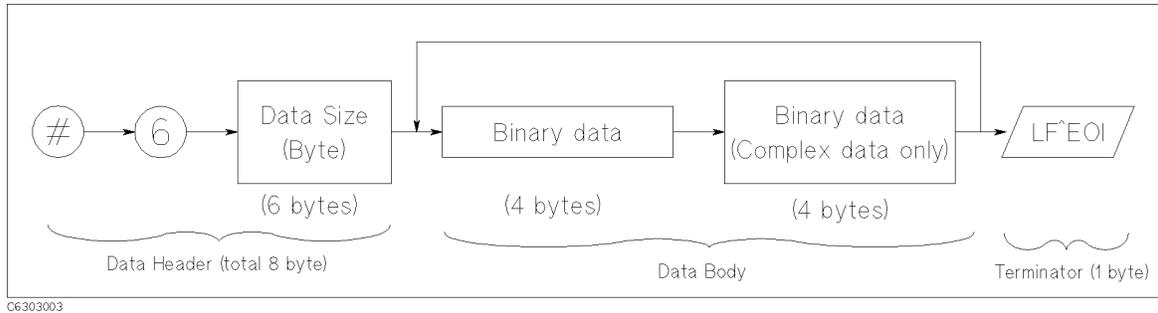


Figure 3-4. IEEE 32-bit floating point format

■ MS-DOS Format

This mode is a modification of the IEEE 32-bit floating point format with the byte order reversed. The MS-DOS format also has a four-byte header that must be read to maintain the data order. In this mode, a PC can store the data internally without reformatting it.

Data Header

As shown in Figure 3-3 and Figure 3-4, the data header always precedes the data itself in binary format transfer. When you use a binary transfer, you must handle the data header with the data body.

When you query data in binary format, the analyzer outputs a fixed length (8 byte) data header. You can handle the data header as 8-byte strings for this purpose.

When you send the data to data array using binary transfer, you must prepare the data header for the data you send. The data header indicates the size of the transferred data. The data header consists of the following three parts: sharp, Number of byte of “Data Size”, and data size.

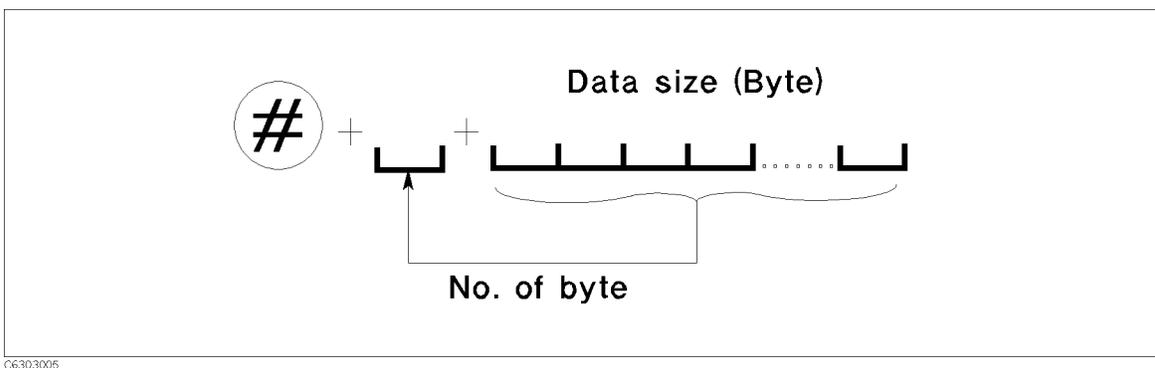


Figure 3-5. Binary Data Header

For example, the data size of 201 points of complex data in the 64-bit format is 3216 byte ($=201 \times 2 \times 8$). The “3216” is 4 digit (4 byte) number. Thus, the data header is “#43216”. The queried data header that is generated from the analyzer is a fixed length header of 8 bytes that is obtained by adding “0” before “Data Size”. For example, the data header above becomes “#6003216” as an 8-byte string. You can use either type of header to send data to the array.

Getting Data from Analyzer

To get a data from the analyzer using a binary transfer method, the following procedure is used:

1. Assign a binary data path. (Specifying format off.)
2. Specify the data transfer format as binary.
3. Define a data array that is the same size as the data that will be retrieved.
4. Send the data query command.
5. Retrieve the data header.
6. Retrieve the data.
7. Retrieve the terminator.
8. Set the transfer format to ASCII mode if binary transfer is finished.

The binary data is sent in a mixed format of an ASCII header and a binary data body as shown in Figure 3-3 and Figure 3-4. To retrieve data correctly, you must retrieve the data header and data itself independently.

The following is a sample module for receiving data using the IEEE 64-bit format:

ASSIGN @Dt TO 800;FORMAT OFF	<i>Binary path must turn off the formatting. Use 717 instead of 800 for the external controller.</i>
OUTPUT @Hp4291;"FORM:DATA REAL,64"	<i>Specify the format as IEEE 64-bit format.</i>
DIM Dat(1:201,1:2)	<i>Assume that the receiving data size is 201x2.</i>
OUTPUT @Hp4291;"TRAC? DTR"	<i>Query the data trace array.</i>
ENTER @Hp4291 USING "#,8A";Header\$	<i>Data header is 8 byte character.</i>
ENTER @Dt;Dat(*)	<i>Receiving data via binary path.</i>
ENTER @Hp4291;End\$	<i>Reading terminator.</i>
OUTPUT @Hp4291;"FORM:DATA ASC"	<i>Set ASCII mode if binary transfer is finished.</i>

Module 3-10. Getting Data from Analyzer Using Binary Transfer

The procedure for sending data is similar to the receiving procedure:

ASSIGN @Dt TO 800;FORMAT OFF	<i>Binary path must turn off the formatting.</i>
OUTPUT @Hp4291;"FORM:DATA REAL,64"	<i>Specify the format as IEEE 64-bit format.</i>
Header\$="#43216"	<i>Data header shows the data size is 3216 byte.</i>
OUTPUT @Hp4291;"TRAC DTR,";Header\$;	<i>Send an array access command with a data header. Do NOT forget last semicolon.</i>
OUTPUT @Dt;Dat(*),END	<i>Send Dat(*) contents and terminator.</i>
OUTPUT @Hp4291;"FORM:DATA ASC"	<i>Set ASCII mode if binary transfer is finished.</i>

Module 3-11. Sending Data to Analyze Using Binary Transfer

■ Related GPIB Commands

The following commands are used to specify the data transfer format.

- | | |
|-------------------|--|
| FORM:DATA REAL,64 | Selects IEEE 64-bit floating point format. |
| FORM:DATA REAL,32 | Selects IEEE 32-bit floating point format. |
| FORM:DATA PACK,32 | Selects MS-DOS format. |

Sample Programs: Compensation Data Transfer

The following programs allow you to store fixture compensation data to the disk and load it back into the analyzer. The `CMP_STOR` command stores the compensation data. The `CMP_LOAD` command loads the stored compensation data into the analyzer. These programs access the fixture compensation array by using a binary transfer method.

Storing Compensation Data to Disk (CMP_STOR)

This program queries the fixture's three compensation arrays and enter them into one array variable, `Cmp_data`. `Cmp_data` has 201×6 dimension to store three sets of the compensation array data. This program uses a 64-bit binary transfer format to send and retrieve data to speed up the transfer speed. See lines 660 to 800.

`CMP_STOR` has a fixture compensation routine that appears in chapter 2 (lines 340 to 620). This is an OPEN, SHORT, and LOAD version of program module 2-12. This program uses a common subroutine for detecting the end of the compensation process for each standard measurement to reduce program size. See `Compen:!` line for the compensation end detection subroutine.

Disk



This program is included in the sample program disk. Its filename is `CMP_STOR`. This program is initialized for Instrument BASIC. The program for the external controller, `CMPSTOR_E`, is also included in the *Sample Program Disk*.

```

100 !**** INITIALIZE: MODULE 1-1 ****
110 ASSIGN @Hp4291 TO 800
120 ASSIGN @Dt TO 800;FORMAT OFF
130 Scode=8
140 ABORT Scode
150 CLEAR @Hp4291
160 !
170 !***** VARIABLE DECLARATIONS *****
180 DIM Cmp$(50)
190 DIM Cmp1(1:201,1:2)
200 DIM Cmp2(1:201,1:2)
210 DIM Cmp3(1:201,1:2)
220 Nop=201
230 OUTPUT @Hp4291;"SENS:SWE:POIN ";Nop
240 !
250 !***** CAL CHECK: MODULE 2-10 *****
260 OUTPUT @Hp4291;"SENS:CORR1:STAT?"
270 ENTER @Hp4291;Stat
280 IF NOT Stat THEN
290   DISP "CALIBRATION REQUIRED!"
300   STOP
310 END IF
320 !
330 !***** Fixture Compensation *****
340 OUTPUT @Hp4291;"STAT:INST:ENAB 256"
350 OUTPUT @Hp4291;"*SRE 4"
360 OUTPUT @Hp4291;"TRIG:SOUR INT"
370 ON INTR Scode GOTO Open_end
380 Cmp$="SENS:CORR2:COLL STAN1"

```

```

390 INPUT "CONNECT OPEN, THEN [RETURN]",Ans$
400 GOTO Compen
410 Open_end:!
420 ON INTR Scode GOTO Short_end
430 Cmp$="SENS:CORR2:COLL STAN2"
440 INPUT "CONNECT SHORT, THEN [RETURN]",Ans$
450 GOTO Compen
460 Short_end:!
470 ON INTR Scode GOTO Load_end
480 Cmp$="SENS:CORR2:COLL STAN3"
490 INPUT "CONNECT LOAD, THEN [RETURN]",Ans$
500 GOTO Compen
510 Load_end:!
520 !
530 OUTPUT @Hp4291;"SENS:CORR2:COLL:SAVE"
540 !
550 GOTO Skip
560 !
570 Compen:!
580 OUTPUT @Hp4291;"*CLS;*OPC?"
590 ENTER @Hp4291;Opc
600 ENABLE INTR Scode;2
610 OUTPUT @Hp4291;Cmp$
620 Waiting:GOTO Waiting
630 !
640 Skip:!
650 !
660 OUTPUT @Hp4291;"FORM:DATA REAL,64"
670 OUTPUT @Hp4291;"DATA? CMP1"
680 ENTER @Hp4291 USING "#,8A";Header$
690 ENTER @Dt;Cmp1(*)
700 ENTER @Hp4291;End$
710 !
720 OUTPUT @Hp4291;"DATA? CMP2"
730 ENTER @Hp4291 USING "#,8A";Header$
740 ENTER @Dt;Cmp2(*)
750 ENTER @Hp4291;End$
760 !
770 OUTPUT @Hp4291;"DATA? CMP3"
780 ENTER @Hp4291 USING "#,8A";Header$
790 ENTER @Dt;Cmp3(*)
800 ENTER @Hp4291;End$
810 !
820 OUTPUT @Hp4291;"FORM:DATA ASC"
830 !
840 !**** DATA SAVE TO FILE ****
850 INPUT "ENTER FILE NAME TO SAVE",File$
860 CREATE File$,Nop*3*2*8
870 ASSIGN @File TO File$;FORMAT OFF
880 OUTPUT @File;Cmp1(*),Cmp2(*),Cmp3(*)
890 ASSIGN @File TO *
900 !
910 END

```

Loading Compensation Data from Disk (CMP_LOAD)

This program loads the fixture compensation data that is saved by CMP_STOR program and enables it.

To enable the compensation data, the OPEN, SHORT, and LOAD correction must be turned on. You cannot turn them on without performing a fixture compensation process. Therefore, this program performs a dummy fixture compensation process in lines 320 to 420.

After turning on the corrections, the program loads the fixture compensation data from the disk to the fixture compensation data array. The program uses a 64-bit binary transfer to send data from the controller to the analyzer. See lines 520 to 600.

Disk



This program is included in the sample program disk. Its filename is CMP_LOAD. This program is initialized for Instrument BASIC. The program for the external controller, CMPLOAD_E, is also included in the *Sample Program Disk*.

```

100 !**** INITIALIZE: MODULE 1-1 ****
110 ASSIGN @Hp4291 TO 800
120 ASSIGN @Dt TO 800;FORMAT OFF
130 Scode=8
140 ABORT Scode
150 CLEAR @Hp4291
160 !
170 !***** VARIABLE DECLARATION *****
180 DIM Cmp1(1:201,1:2)
190 DIM Cmp2(1:201,1:2)
200 DIM Cmp3(1:201,1:2)
210 Nop=201
220 OUTPUT @Hp4291;"SENS:SWE:POIN ";Nop
230 !
240 !***** CAL CHECK: MODULE 2-10 *****
250 OUTPUT @Hp4291;"SENS:CORR1:STAT?"
260 ENTER @Hp4291;Stat
270 IF NOT Stat THEN
280   DISP "CALIBRATION REQUIRED!"
290   STOP
300 END IF
310 !
320 !***** Dummy Compensation *****
330 OUTPUT @Hp4291;"STAT:INST:ENAB 256"
340 OUTPUT @Hp4291;"*SRE 4"
350 OUTPUT @Hp4291;"TRIG:SOUR INT"
360 ON INTR Scode GOTO Open_end
370 OUTPUT @Hp4291;"*CLS;*OPC?"
380 ENTER @Hp4291;Opc
390 ENABLE INTR Scode;2
400 OUTPUT @Hp4291;"SENS:CORR2:COLL STAN1"
410 Waiting:GOTO Waiting
420 Open_end:!
430 !
440 OUTPUT @Hp4291;"SENS:CORR2:COLL:SAVE"
450 !

```

```
460 !***** COMPEN DATA INPUT *****  
470 INPUT "ENTER COMPEN DATA FILE NAME",File$  
480 ASSIGN @File TO File$  
490 ENTER @File;Cmp1(*),Cmp2(*),Cmp3(*)  
500 ASSIGN @File TO *  
510 !  
520 OUTPUT @Hp4291;"FORM:DATA REAL,64"  
530 Header$="#43216"  
540 OUTPUT @Hp4291;"DATA CMP1,";Header$;  
550 OUTPUT @Dt;Cmp1(*),END  
560 OUTPUT @Hp4291;"DATA CMP2,";Header$;  
570 OUTPUT @Dt;Cmp2(*),END  
580 OUTPUT @Hp4291;"DATA CMP3,";Header$;  
590 OUTPUT @Dt;Cmp3(*),END  
600 OUTPUT @Hp4291;"FORM:DATA ASC"  
610 !  
620 END
```

Using Status Reporting System

The analyzer has status registers that report system conditions. The register contents are changed depending on the particular condition of the analyzer. By reading this register, you can determine the specific analyzer status.

This chapter provides the following information:

- General status register model.
- Status register structure.
- How to use status register in a program.
- Sample program: performing calibration

General Status Register Model

The analyzer has a status reporting system to report the condition of the analyzer.

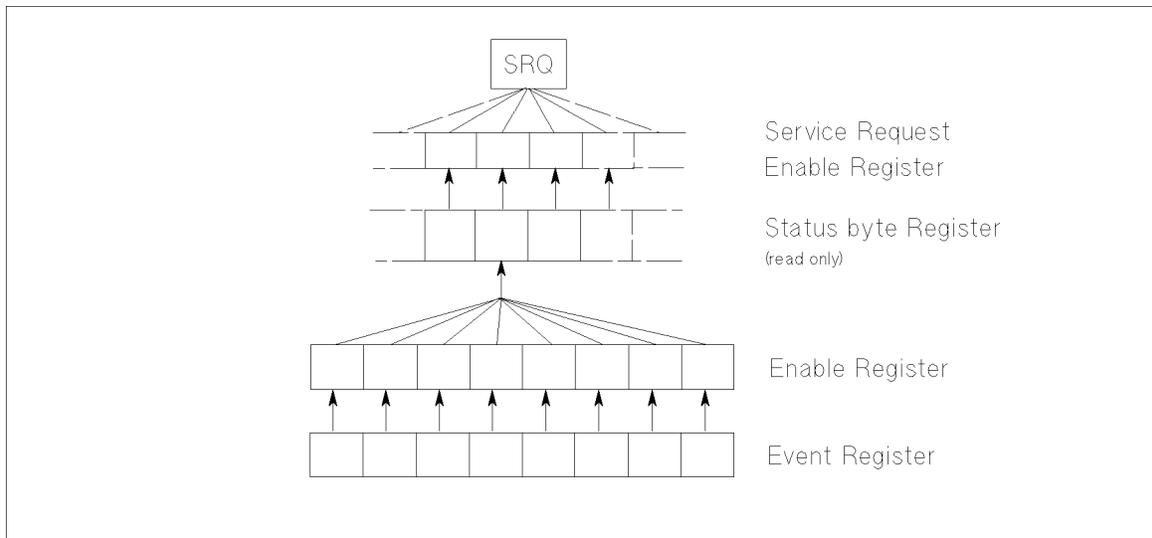


Figure 4-1. General Status Register Model

The status reporting system has a hierarchical structure as shown in Figure 4-1. When the analyzer condition satisfies the particular condition, the corresponding bit of the event register is set TRUE. Therefore, you can check the analyzer condition by reading the event register.

When the event register bit is set to TRUE, and corresponding enable register bit is also TRUE, the summary bit of the status byte register is set to TRUE. You can read the status byte register by using the serial poll.

If the corresponding bit of the service request enable register is TRUE, the service request (SRQ) is generated with the positive transition of the status byte register bit. By generating the SRQ, you can notify the controller that the analyzer is requesting service. You can program for an interruption request by the SRQ. See “SRQ and Interrupt” for more details about the program requirements.

Event Register

Reflects the correspondent analyzer condition as a bit status. These bits monitor the changing analyzer state continuously and change the bits status as required.

You cannot change bit status by GPIB command.

The analyzer has the following event registers:

- Instrument Event Status Register (see Table 4-2 for details).
- Standard Event Status Register (see Table 4-3).
- Operation Status Event Register (see Table 4-4).

Enable Register

The enable register selects which event register bits can set the bit in the summary bit of the status byte register that is connected to SRQ generation. The register bits work like mask bits. When you want to set a bit in the status byte register by a specific register condition, set the corresponding enable register to 1. This sets a 1 bit in the status byte register with a corresponding event register bit.

Use this register to select which event register bits generate the SRQ.

All event registers have a corresponding enable register for each bit.

Status Byte Register

If enabled event register is set to 1, the corresponding bit of the status byte register is set to 1. This register also indicates the output queue and SRQ status.

The value of the status byte register can be read by using the SPOLL statement or *STB? query by the controller. SPOLL reads the status byte register value directly without being set the analyzer to remote. Therefore, you can continue to operate front panel keys while a controller is reading the status byte register. On the other hand, the *STB? query sets the analyzer to remote mode. Reading the status byte register by either command does not affect the contents of the status byte register, except for the SPOLL clears RQS bit. Table 4-1 lists the contents of the status byte register.

A serial poll initiated by using the SPOLL command reads bit 6 of the status byte register as the RQS bit. The *STB? command reads bit 6 as the MSS bit.

SRQ (Service Request) can be generated the status byte register by setting the service request enable register. For more information about SRQ, see “SRQ and Interrupt” in this chapter.

Transition Filter and Condition Register

The transition filter allows to select which transitions of the analyzer condition to set a bit in the event register.

When the status register has a transition filter, there is a lower register called a condition register under the event register. The transition filter is in between the event register and condition register. The transition filter enables you to select a positive and/or negative transition of the condition register bit to set a bit in the corresponding event register. For example, if you set the negative transition filter, 1 is set in the event register by changing 1 to 0 in the event register.

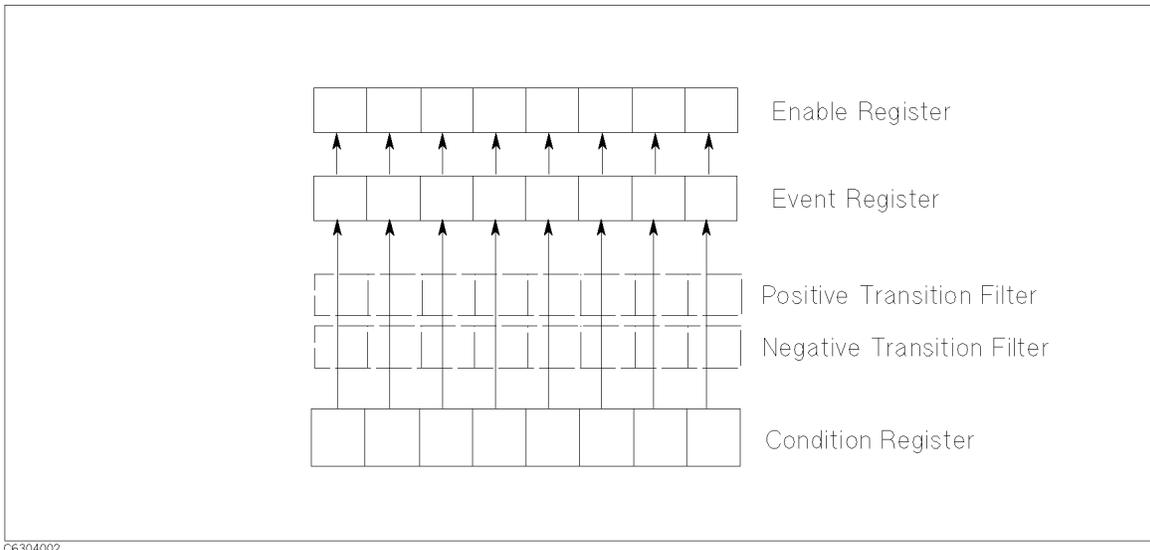


Figure 4-2. Transition Filter and Condition Register

For the 4291B, only “Program Running” bit of the operation status register has a transition filter. (See Figure 4-3.) By using the transition filter, you can generate an SRQ either at the start or the end of the program execution. See “Determining Instrument BASIC Execution State” in Chapter 8 for an example program using the transition filter.

Status Register Structure

The status register has a hierarchical structure. The status byte register summarizes the low level registers. This section shows the status register structure of the analyzer (Figure 4-3) and describes each bit of the status registers from Table 4-1 to Table 4-4.

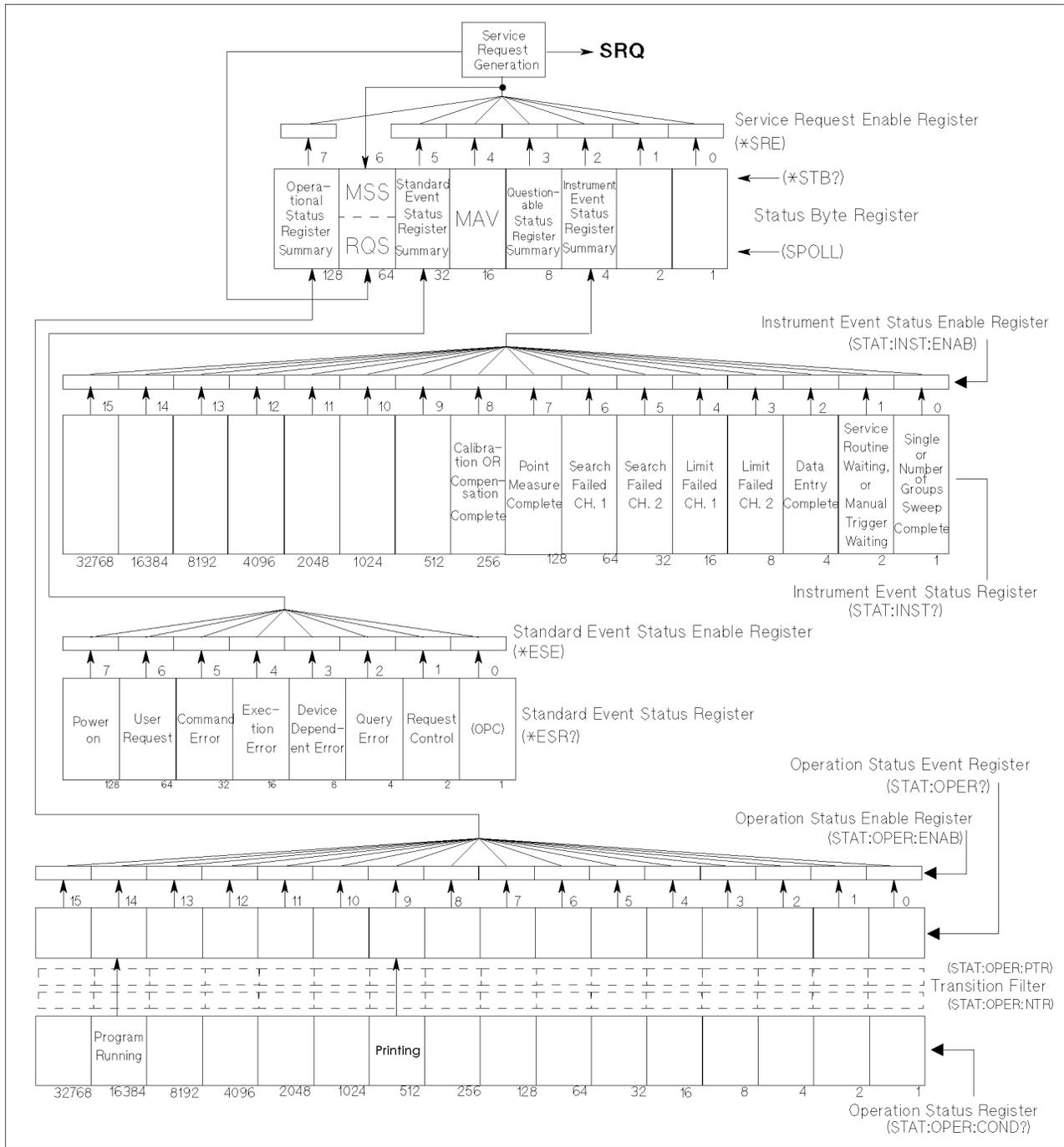


Figure 4-3. Status Register Structure

Table 4-1. Status Bit Definitions of the Status Byte Register

Bit	Name	Description
2	Instrument Event Status Register Summary Bit	One of the enabled bits in Instrument Event Status Register has been set.
3	Questionable Status Register Summary Bit	The analyzer has no operation to report the event to the Questionable Status Register group. This register is available to maintain consistency with other SCPI compatible devices.
4	MAV (Message Available)	An information has been prepared to be output, but it has not been read yet.
5	Standard Event Status Register Summary Bit	One of the enabled bits in the Standard Event Status Register has been set.
6	RQS	The analyzer generated an SRQ. When reading the status byte register by SPOLL, bit 6 is RQS bit.
	MSS	The analyzer has at least one TRUE bit in the status byte register that is enabled by the service request enable register. When reading the status byte register by *STB?, bit 6 is MSS bit.
7	Operation Status Register Summary Bit	One of the enabled bits in the Operation Status Register has been set.

A status bit is cleared when *CLS command is executed.

Table 4-2.
Status Bit Definitions of the Instrument Event Status Register

Bit	Name	Description
0	Single or Number of Groups Sweep Complete	A single sweep or group has been completed since the last read of the register.
1	Service Routine Waiting or Manual Trigger Waiting	1. An internal service routine has completed an operation or is waiting for an operator response. 2. The analyzer has set the manual trigger mode and is waiting for a manual trigger.
2	Data Entry Complete	A terminator key has been pressed.
3	Limit Failed, Ch 2	Limit test failed on channel 2.
4	Limit Failed, Ch 1	Limit test failed on channel 1.
5	Search Failed, Ch 2	A marker search was executed on channel 2, but the target value was not found.
6	Search Failed, Ch 1	A marker search was executed on channel 1, but the target value was not found.
7	Point Measurement Complete ¹	One measurement point of a sweep has been completed.
8	Calibration or Compensation Complete	The analyzer completed a calibration or compensation process including a measurement and a calculation for the coefficients, and ready to perform next process.

¹ This bit is set only when both related bits of the service request enable register and the instrument event status enable register are enabled.

Table 4-3. Status Bit Definitions of the Standard Event Status Register

Bit	Name	Description
0	Operation Complete	The analyzer has completed all pending operations and is ready to accept new commands. This bit is generated only in response to the *0PC command.
1	Request Control	The analyzer requires control of GPIB to proceed the current operation.
2	Query Error	<ol style="list-style-type: none"> 1. The analyzer has been addressed to talk, but there is nothing in the output queue to transmit. 2. Data in the Output Queue has been lost.
3	Device Dependent Error	An error, other than a command error, a query error, or an execution error has occurred.
4	Execution Error	<ol style="list-style-type: none"> 1. An GPIB command parameter exceeded its input range, or is inconsistent with the analyzer's capabilities. 2. An GPIB command could not be properly executed due to some analyzer condition.
5	Command Error	<ol style="list-style-type: none"> 1. An IEEE 488.2 syntax error has occurred. The analyzer receives a command that did not follow the syntax defined by the IEEE 488.2 standard. Possible violations include, a command parameter violated the analyzer listening formats or a command parameter type is unacceptable to the analyzer. 2. A semantic error occurred. For example, the analyzer received an incorrectly spelled command. Another example would be that the analyzer received an optional 488.2 command that is not implemented to the analyzer. 3. The analyzer received a Group Execute Trigger (GET) inside a GPIB command.
6	User Request	The operator pressed a front panel key or an optional keyboard key or turned the rotary knob.
7	Power ON	A power-on sequence has occurred since the register was last read.

Table 4-4. Status Bit Definitions of the Operation Status Register

Bit	Name	Description
9	Printing	Data is being transferred to the printer.
14	Program running	An HP instrument BASIC program is running.

How to Use the Status Registers in a Program

You can use the status registers to determine the specific analyzer status in the program. To determine the contents of the status register, the following methods are used:

- Read an event register directly.
- Use the Service Request (SRQ).

Reading an Event Register Directly

You can read the contents of the event register directly to determine the specific analyzer condition. Use this method if you do not need to know the timing of the event register changes. The following procedure reads the register directly:

1. Query the event register contents.
2. Retrieve a return value.
3. Check the bit condition using the BASIC BIT function.

OUTPUT @Hp4291;"STAT:INST?"	<i>Queries instrument event status register contents.</i>
ENTER @Hp4291;Esb	<i>Retrieve return value.</i>
IF BIT(Esb,4) THEN	<i>If bit 4 of the instrument event status register is set to 1, the limit</i>
DISP "LIMIT TEST FAILED AT Ch 1."	<i>test failed on channel 1.</i>
END IF	

Module 4-1. Reading an Event Register

- Related GPIB Commands. The following query commands can be used to read the contents of an event register directly.

*STB?	Returns Status Byte Register contents.
*ESR?	Returns Event Status Register contents.
STAT:INST?	Returns Instrument Event Status Register contents.
STAT:OPER:COND?	Returns Operation Status Register contents.

SRQ and Interrupt

You can initialize your program to enable interrupt processing by the Service Request (SRQ) from the analyzer. The analyzer generates an SRQ when the specified condition is satisfied.

The SRQ itself does not contain information on the SRQ source. However, the Request Service (RQS) bit in the Status Byte Register of the SRQ source device is set to 1. If multiple devices are connected on the bus, you can check the RQS bit (bit 6) of the analyzer by using a serial poll, SPOLL.

Use the SRQ interrupt if you want to determine when the condition changes. The following procedure is used to set the SRQ interrupt:

1. Define the branch for the interruption. (Use `ON INTR` statement.)
2. Set to 1 the enable register for the correspondent event register bit.
3. Set to 1 the service request enable register bit for the correspondent status byte register bit.
4. Clear the status register before enabling the SRQ interruption.
5. Enable the SRQ interruption. (Use `ENABLE INTR` statement.)
6. Start the event.
7. Wait for the SRQ. Usually, the program waits within an endless loop.
8. If multiple devices that can generate an SRQ exist on the GPIB, you should check bit 6 of the status byte register of the target device. If the SRQ is generated from the target device, the status byte register bit 6 is set to 1.

The following example uses an SRQ interruption for detecting the end of sweep. Bit 0 of the instrument event status register is used for this purpose.

<code>ON INTR Scode GOTO Sweep_end</code>	<i>When the SRQ has occurred, jumps to label, "Sweep_end."</i>
<code>OUTPUT @Hp4291;"STAT:INST:ENAB 1"</code>	<i>Set bit 0 of the instrument event status enable register to 1. ($2^0 = 1$)</i>
<code>OUTPUT @Hp4291;"*SRE 4"</code>	<i>Set bit 2 of the service request enable register to 1. ($2^2 = 4$)</i>
<code>OUTPUT @Hp4291;"*CLS"</code>	<i>Clears the event register.</i>
<code>OUTPUT @Hp4291;"*OPC?"</code>	<i>Confirms the clear operation is completed.</i>
<code>ENTER @Hp4291;0pc</code>	
<code>!</code>	
<code>OUTPUT @Hp4291;"TRIG:SOUR INT"</code>	<i>Selects internal trigger source.</i>
<code>OUTPUT @Hp4291;"INIT:CONT OFF"</code>	<i>Sets the continuous mode OFF.</i>
<code>OUTPUT @Hp4291;"ABOR"</code>	<i>Moves trigger sequence to the idle state.</i>
<code>ENABLE INTR Scode;2</code>	<i>Enables the SRQ interruption just before triggering.</i>
<code>OUTPUT @Hp4291;"INIT"</code>	<i>Trigger a measurement.</i>
<code>Waiting: GOTO Waiting</code>	<i>Waits until SRQ is generated.</i>
<code>Sweep_end:!</code>	<i>When on SRQ, program jumps to this label.</i>
<code>IF NOT BIT(SPOLL(@Hp4291),6) THEN</code>	<i>Check the SRQ is generated from the target device by checking status byte register bit 6.</i>
<code> ENABLE INTR Scode;2</code>	<i>If not, enable the SRQ again, then</i>
<code> GOTO Waiting</code>	<i>returns to the endless looping.</i>
<code>END IF</code>	

Module 4-2. Detecting Sweep End Using SRQ and Interrupt

Note



*CLS clears only the event registers and the status byte register. The enable register and transition filter settings are not altered by executing the *CLS command. To clear the enable register and transition filter, use `SYST:PRES` command.

Figure 4-4 shows the SRQ generation sequence of the example above.

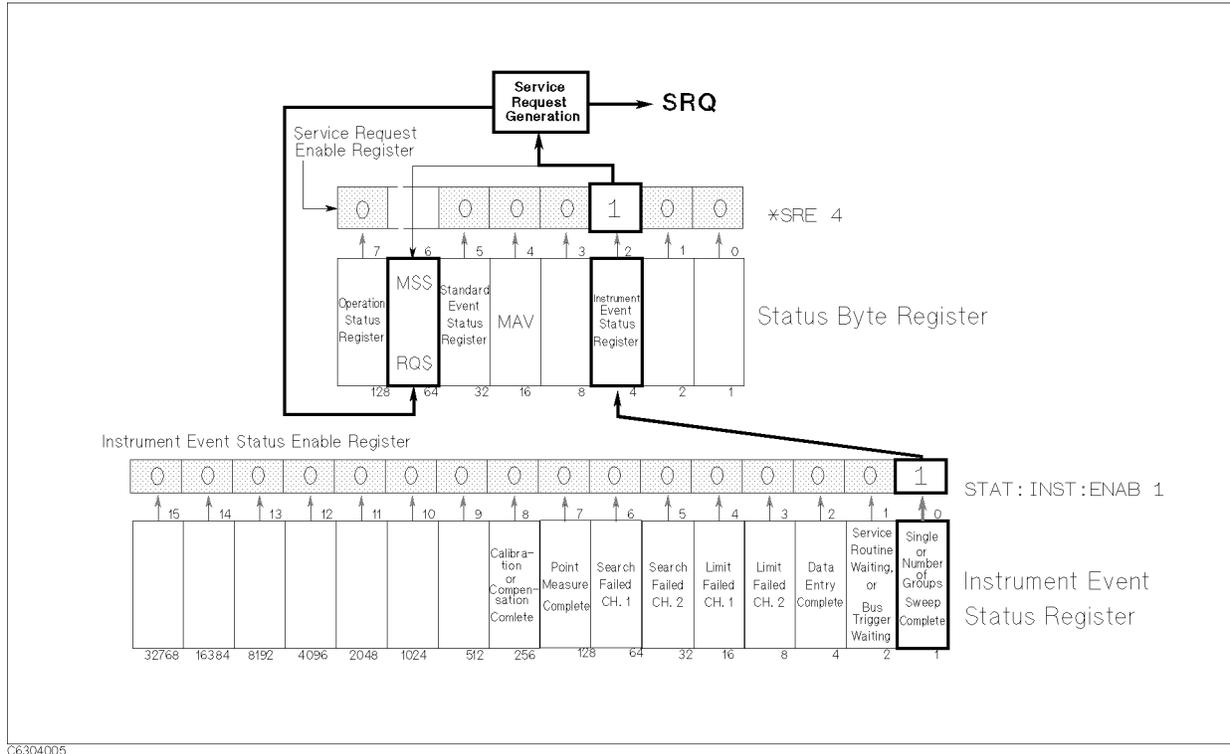


Figure 4-4. SRQ Generation Sequence

■ Related GPIB Commands

The following GPIB commands are used for setting the SRQ generation:

- *SRE *decimal*** Sets the service request enable register.
- *ESE *decimal*** Sets the enable register for event status register.
- STAT:INST:ENAB *decimal*** Sets the enable register for instrument event status register.
- STAT:OPER:ENAB *decimal*** Sets the enable register for operation status register.
- STAT:OPER:PTR *decimal*** Sets the transition filter to positive for operation status register.
- STAT:OPER:NTR *decimal*** Sets the transition filter to negative for operation status register.

Sample Program: Performing Calibration

This sample program automates the calibration process. For the same reason as the fixture compensation process that is discussed in Chapter 2, the calibration process also requires synchronizing the program with the analyzer by detecting the calibration process completion.

The calibration process measures a standard in three different settings. Bit 8 of the instrument event status register ("Calibration or Compensation Complete" bit) is set to 1 when all measurements are completed. By setting the enable registers to generate an SRQ from this bit, you can detect the end of the calibration process of each standard.

You have to use an SRQ interrupt for measuring every standard. When you enable the interrupt, do not forget to clear the register to avoid an unexpected interruption from occurring. Clearing register by *CLS does not alter the enable register setting. Therefore, setting the enable register only occurs once in this program (lines 330 and 340).

Disk



This program is included in the sample program disk. Its filename is CAL. This program is initialized for Instrument BASIC. The program for the external controller, CAL_E, is also included in the *Sample Program Disk*.

```

100 !**** INITIALIZE ****
110 ASSIGN @Hp4291 TO 800
120 Scode=8
130 CLEAR @Hp4291
140 ABORT Scode
150 !
160 !**** CALIBRATION STATE CHEKING ****
170 OUTPUT @Hp4291;"SENS:CORR1?"
180 ENTER @Hp4291;Stat
190 IF Stat THEN
200   INPUT "ARE YOU SURE TO RE-CALIBRATE NOW? [Y/N]",Ans$
210   IF Ans$"Y" THEN STOP
220 END IF
230 !
240 !**** SELECTING CAL MODE (USRER/FIXED) ****
250 INPUT "SELECT CAL MODE [F]IXED OR [U]SER. [F/U] (DEFAULT=F)",Cal_mode$
260 IF Cal_mode$="U" THEN
270   OUTPUT @Hp4291;"SENS:CORR1:COLL:FPO USER"
280 ELSE
290   OUTPUT @Hp4291;"SENS:CORR1:COLL:FPO FIX"
300 END IF
310 !
320 !**** SETTING ENABLE REGISTER FOR SRQ ****
330 OUTPUT @Hp4291;"STAT:INST:ENAB 256"
340 OUTPUT @Hp4291;"*SRE 4"
350 OUTPUT @Hp4291;"TRIG:SOUR INT"
360 !
370 !**** OPEN STD. MEASUREMENT ****
380 ON INTR Scode GOTO Open_end
390 OUTPUT @Hp4291;"*CLS;*OPC?"
400 ENTER @Hp4291;Opc
410 INPUT "CONNECT OPEN, THEN PRESS [RETURN]",A
420 ENABLE INTR Scode;2

```

```
430 OUTPUT @Hp4291;"SENS:CORR1:COLL STAN1"
440 Wait_open:GOTO Wait_open
450 Open_end:!
460 !
470 !**** SHORT STD. MEASUREMENT ****
480 ON INTR Scode GOTO Short_end
490 OUTPUT @Hp4291;"*CLS;*OPC?"
500 ENTER @Hp4291;0pc
510 INPUT "CONNECT SHORT, THEN PRESS [RETURN]",A
520 ENABLE INTR Scode;2
530 OUTPUT @Hp4291;"SENS:CORR1:COLL STAN2"
540 Wait_short:GOTO Wait_short
550 Short_end:!
560 !
570 !**** LOAD STD. MEASUREMENT ****
580 ON INTR Scode GOTO Load_end
590 OUTPUT @Hp4291;"*CLS;*OPC?"
600 ENTER @Hp4291;0pc
610 INPUT "CONNECT LOAD, THEN PRESS [RETURN]",A
620 ENABLE INTR Scode;2
630 OUTPUT @Hp4291;"SENS:CORR1:COLL STAN3"
640 Wait_load:GOTO Wait_load
650 Load_end:!
660 !
670 !**** LOW-LOSS C MEASUREMENT ****
680 INPUT "DO YOU WANT TO MESURE LOW-LOSS C? [Y/N] Default=Y",A$
690 IF A$="N" OR A$="n" THEN Skip
700 ON INTR Scode GOTO Llc_end
710 OUTPUT @Hp4291;"*CLS;*OPC?"
720 ENTER @Hp4291;0pc
730 INPUT "CONNECT LOW-LOSS C, THEN PRESS [RETURN]",A
740 ENABLE INTR Scode;2
750 OUTPUT @Hp4291;"SENS:CORR1:COLL STAN4"
760 Wait_llc:GOTO Wait_llc
770 Llc_end:!
780 !
790 Skip:!
800 OUTPUT @Hp4291;"SENS:CORR1:COLL:SAVE"
810 !
820 END
```

Using the Trigger System

This chapter provides information about the SCPI trigger system of the analyzer. SCPI defines a common trigger system for various types of instruments. This means you can control the analyzer for your measurements using operations that are common to other instruments.

The analyzer has two trigger modes: trigger on sweep mode and trigger on point mode. The trigger on sweep mode sweeps all measurement points with single trigger input. The trigger on point mode requires a trigger for every measurement point. This section describes the characteristics and differences of each mode.

Trigger System

The trigger system has three states during a measurement. Figure 5-1 shows the trigger sequence:

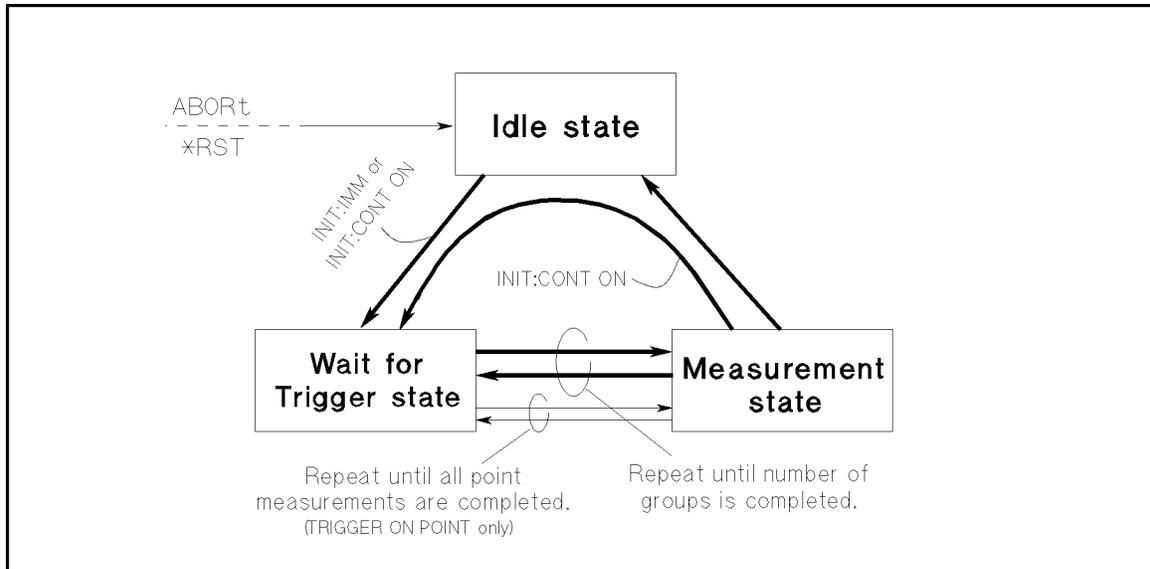


Figure 5-1. Simplified Trigger System

Idle State

The idle state is the start point of the trigger sequence. The idle state has the following characteristics:

- *RST or ABORt sets the analyzer to the idle state at anytime during the sequence.
- INIT(: IMM) moves the trigger sequence to the wait for trigger state from the idle state.
- If INIT:CONT ON is set, the trigger sequence immediately moves to the wait for trigger state.

When the analyzer in the idle state, the Hld notation is displayed on the left edge of the display.

Wait for Trigger State

In the wait for trigger state, the analyzer waits for a selected trigger event. It then moves to the measurement state.

The following trigger sources are available for the analyzer:

Trigger Source	Trigger Event
INTERNAL	Internal trigger source. This source continuously generates a trigger event. The trigger sequence immediately moves to the measurement state.
EXTERNAL	External trigger source. Apply a TTL level pulse longer than 2 μ s to the external trigger connector on the rear panel to trigger. TRIG:SLOP POS or NEG toggles the polarity of the trigger signal to positive or negative logic, respectively.
MANUAL	Manual trigger. Press MANUAL key under Trigger TRIGGER [MANUAL] to trigger.

5-2 Using the Trigger System

BUS GPIB trigger. Executing either of the following statements triggers a measurement.
OUTPUT @Hp4291;"*TRG" or TRIGGER @Hp4291

Measurement State

In this state, the analyzer makes a measurement. The measurement method depends on whether the selected measurement mode is the trigger on sweep mode or the trigger on point mode.

When the trigger on sweep mode is selected, the analyzer measures all points by a single trigger. After all the point measurements are completed, it exits the measurement state to the idle state. If the number of groups sweep is selected, it goes back to the wait for trigger state and repeats this loop until the specified number of groups is completed. It then moves to the idle state. If the single sweep is selected, the trigger sequence immediately moves to the idle state after all point measurements are completed.

If INIT:CONT ON is selected, the trigger sequence skips the idle state and goes to the wait for trigger state.

When the trigger on point mode is selected, the analyzer measures one point and then moves to the wait for trigger state. Therefore, you must apply a trigger for each point measurement. When the measurement point reaches the right edge of the display, the trigger sequence acts the same as the trigger on sweep mode.

If INIT:CONT ON is set, the trigger sequence immediately moves to the wait for trigger state after exiting from the measurement state.

Sweeping Once Using the GPIB Trigger

To write a program for a single sweep using the GPIB trigger, perform the following procedure:

1. Define the branch for the SRQ interrupt.
2. Set instrument event status enable register bit 0.
3. Set service request enable register bit 2.
4. Set the trigger source to bus trigger.
5. Enable an interrupt.
6. Send GPIB trigger.
7. Wait for completion of sweep.

You have to wait within the program during the analyzer measurement time to synchronize the program and the measurement. You can detect the completion of a measurement by using the SRQ. Use the instrument event status enable register bit 0 for detecting the sweep end. For details about the SRQ, see "SRQ and Interrupt" in Chapter 4.

ON INTR Scode GOTO Sweep_end	<i>When the sweep is completed, program jumps to Sweep_end: ! line.</i>
OUTPUT @Hp4291;"STAT:INST:ENAB 1"	<i>Enables Instrument Event Status Enable Register bit 0.</i>
OUTPUT @Hp4291;"*SRE 4"	<i>Enables Service Request Enable Register bit 2.</i>
OUTPUT @Hp4291;"*CLS;*OPC?"	<i>Clears the status register.</i>
ENTER @Hp4291;0pc	<i>Wait until *CLS operation is completed.</i>
OUTPUT @Hp4291;"TRIG:SOUR BUS"	<i>Selects GPIB trigger.</i>
OUTPUT @Hp4291;"INIT:CONT OFF"	<i>Turns off the continuous mode.</i>
OUTPUT @Hp4291;"ABOR"	<i>Moves trigger sequence to the idle state.</i>
OUTPUT @Hp4291;"INIT"	<i>Moves to wait for trigger state.</i>
WAIT .01	<i>Waits for initializing.</i>
ENABLE INTR Scode;2	<i>Enables the SRQ interrupt.</i>
TRIGGER @Hp4291	<i>Triggers a sweep.</i>
Waiting:GOTO Waiting	<i>Waits until the sweep is completed.</i>
Sweep_end: !	

Module 5-1. Sweeping Once Using the GPIB Trigger

Sweeping a Specified Number of Times

When you want to sweep a specified number of times you can use the number of groups sweep. The number of groups sweep sweeps the specified number of times and then goes to the idle state. This sweep can be used when you want to perform averaging.

The number of groups sweep requires a trigger for each sweep. Therefore, we use the internal trigger source to avoid applying a new trigger for each sweep in this sample module. When you use the internal trigger, the INIT command becomes a trigger for the measurement.

Perform the following steps to sweep the specified number of times using an internal trigger.

1. Define the branch for the SRQ interrupt.
2. Set instrument event status enable register bit 0.
3. Set service request enable register bit 2.
4. Clear the register.
5. Set the trigger source to internal trigger.
6. Specify the number of times to sweep.

7. Set INIT:CONT OFF.
8. Move the trigger sequence to the Idle state using the ABOR command.
9. Enable an interrupt.
10. Start number of groups sweep by sending INIT command.
11. Wait for completion of specified number of sweeps.

When the number of groups sweep is selected, the instrument event status register bit 0 also indicates the status of the number of groups sweep. When the specified number of sweeps are completed, this bit is set to 1.

ON INTR Scode GOTO Sweep_end	<i>When the sweep is completed, program jumps to Sweep_end: ! line.</i>
OUTPUT @Hp4291;"STAT:INST:ENAB 1"	<i>Enables Instrument Event Status Enable Register bit 0.</i>
OUTPUT @Hp4291;"*SRE 4"	<i>Enables Service Request Enable Register bit 2.</i>
OUTPUT @Hp4291;"*CLS;*OPC?"	<i>Clears the status register.</i>
ENTER @Hp4291;Opc	<i>Wait until *CLS operation is completed.</i>
OUTPUT @Hp4291;"TRIG:SOUR INT"	<i>Selects internal trigger. You can omit this line after presetting .</i>
OUTPUT @Hp4291;"INIT:CONT OFF"	<i>Turns the continuous mode off.</i>
OUTPUT @Hp4291;"SENS:SWE:COUN 16"	<i>Sets the number of sweeps to 16.</i>
OUTPUT @Hp4291;"ABOR"	<i>Moves to the idle state.</i>
ENABLE INTR Scode;2	<i>Enables the SRQ interrupt.</i>
OUTPUT @Hp4291;"INIT"	<i>Start a measurement.</i>
Waiting:GOTO Waiting	<i>Waits until the sweep is completed.</i>
Sweep_end:!	

Module 5-2. Sweeping a Specified Number of Times

■ Related GPIB Commands

The following commands set the averaging requirements:

SENS:AVER2:COUN *value*;STAT ON Specify the averaging factor and turns averaging ON.

Triggering on Each Point Using the Manual Trigger

The following program module makes a point measurement using the manual trigger and displays the measured value on the display:

1. Set instrument event status enable register bit 7.
2. Set service request enable register bit 2.
3. Set the trigger source to manual.
4. Select trigger on point mode.
5. Move to wait for trigger state.
6. Enable an interrupt.
7. Press **MANUAL** under **Trigger** TRIGGER: [FREE RUN].
8. Wait for completion of point measurement.
9. Query the measured point value.
10. Display the value.
11. Repeat 6 to 10.

Use the "Point Measure Complete" bit (Instrument event status enable register bit 7) to detect the end of the on point measurement.

OUTPUT @Hp4291;"TRIG:SOUR MAN"	<i>Selects manual trigger.</i>
OUTPUT @Hp4291;"TRIG:EVENT:TYPE POIN"	<i>Select trigger on point mode.</i>
ON INTR Scode GOTO Sweep_end	
OUTPUT @Hp4291;"STAT:INST:ENAB 128;*SRE 4"	<i>Set SRQ generation with point measure complete.</i>
OUTPUT @Hp4291;"INIT"	<i>Move to wait for trigger state.</i>
FOR I=1 TO 201	
OUTPUT @Hp4291;"*CLS;*OPC?"	
ENTER @Hp4291;0pc	
ENABLE INTR Scode;2	<i>Enable the SRQ interrupt.</i>
Waiting:GOTO Waiting	<i>Wait for trigger and measurement completion.</i>
Sweep_end:!	
OUTPUT @Hp4291;"TRAC:VAL? DTR,";I	<i>Output a measured point value.</i>
ENTER @Hp4291;Value	<i>Receive a response.</i>
PRINT Value	<i>Display received value.</i>
NEXT I	

Module 5-3. Triggering on Each Point Using the Manual Trigger

■ Related GPIB Commands

The following commands selects the trigger mode:

TRIG:EVENT:TYPE SWE Selects trigger on sweep mode.

TRIG:EVENT:TYPE POIN Selects trigger on point mode.

The following commands output a specified measurement point value:

TRAC:VAL? DTR,*point* Outputs the specified point data of the data trace of the active channel.

TRAC:VAL? MTR,*point* Outputs the specified point data of the memory trace of the active channel.

TRAC:VAL? DTRCH1,*point* Outputs the specified point data of the data trace of channel 1.

TRAC:VAL? DTRCH2,*point* Outputs the specified point data of the data trace of channel 2.

The following commands sets the averaging on point:

SENS:AVER1:COUN *value*;STAT ON Sets the point averaging factor and turns on it.

SENS:SWE:DWEL1 *value*;DWEL1:AUTO ON Sets the point delay time.

Using the I/O Port

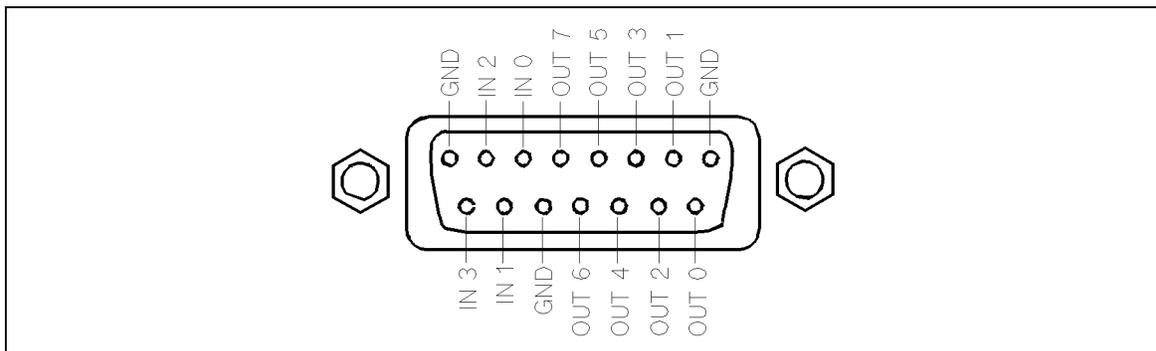
The analyzer has an 8-bit general purpose input/output port on the rear panel. You can use this port to connect an external instrument to the analyzer. This chapter describes the pin assignment, access commands and a sample program for using with a handler.

This chapter contains:

- I/O port pin assignment
- Accessing I/O port
- Sample program: BIN sorting using the I/O port

I/O Port Pin Assignment

Figure 6-1 shows the hardware pin assignment of the I/O port.



C6306001

Figure 6-1. I/O Port Pin Assignment

There are 4 pins for input and 8 pins for output. Therefore, the analyzer can handle 4 bit data for input and 8 bit data for output.

Accessing I/O Port

To read or write to the I/O port from Instrument BASIC, use the following command:

```
Dat=11
WRITEIO 15,0;Dat  Outputs "11". (bit 0,1, and 3 are ON)
```

Module 6-1. Send Data to I/O Port

You may notice that Module 6-1 does not use an OUTPUT statement to access to the I/O port. Instead it uses the WRITEIO statement. The WRITEIO statement of Instrument BASIC is provided for the purpose of accessing the I/O port. This statement allows faster access than the OUTPUT statement because it directly accesses the I/O port. Therefore, this statement can only be used with Instrument BASIC.

```
Dat=READIO(15,0)  Returns 4 bit decimal value from input of I/O port.
```

Module 6-2. Reading Data from I/O Port

Like the WRITEIO statement, READIO allows you to read data from the input of the I/O port with Instrument BASIC. READIO is a function that you can use in equations.

Access I/O Port from the External Controller

If you want to access the I/O port from an external controller, you have to use an GPIB command with the OUTPUT statement instead of using the READIO or WRITEIO commands.

To write data to the I/O port from the external controller, use the following GPIB command:

```
OUTPUT @Hp4291;"SYST:COMM:PAR:TRAN:DATA ";Dat
```

Module 6-3. Send Data to I/O Port from an External Controller

To read data from the I/O port to the external controller, use the following GPIB commands:

```
OUTPUT @Hp4291;"SYST:COMM:PAR:DATA?"
ENTER @Hp4291;Dat
```

Module 6-4. Reading Data from I/O Port to an External Controller

Note



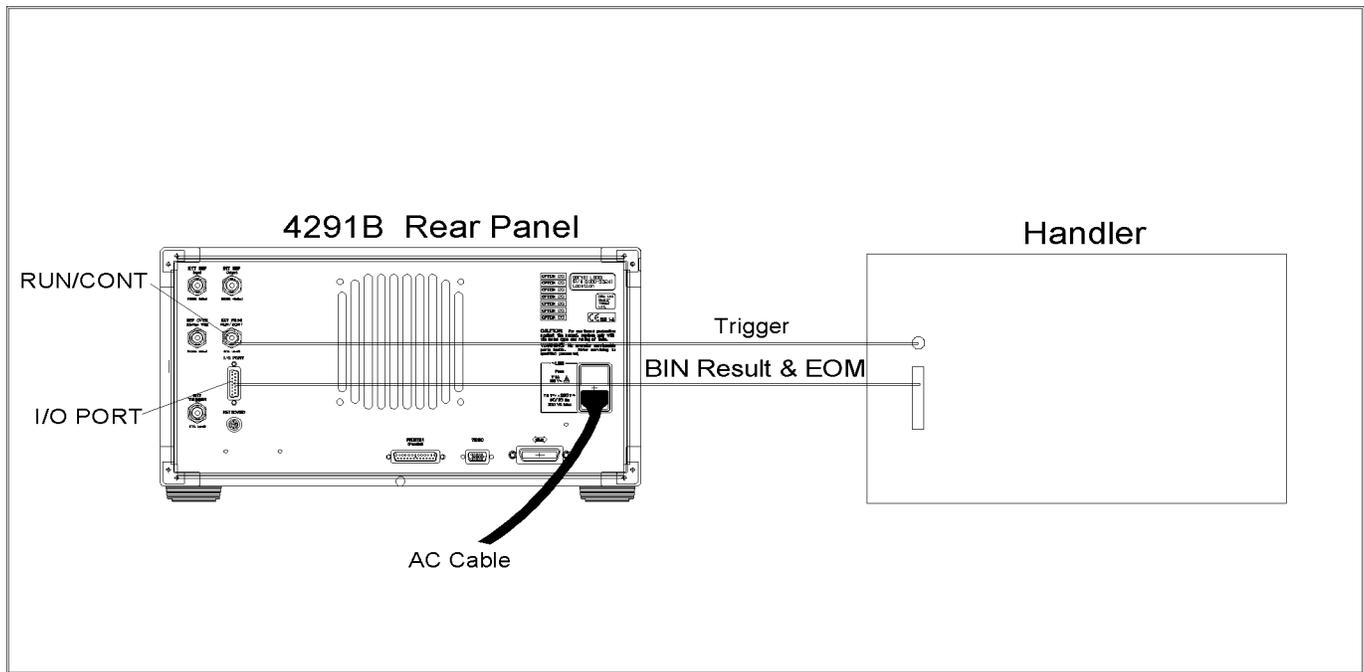
When you are using a fast external controller to control the analyzer, a continuously looping query causes the analyzer to lock up. If this happens, add a WAIT statement within the loop to slow down the query frequency.

Sample Program: BIN Sorting Using the I/O Port

This section provides a sample program that performs BIN sorting with the external handler and outputs the result via the I/O port.

This sample program sorts the DUTs into 4 bins using the difference rate from a reference value at a fixed frequency. Then it outputs the result of sorting to the I/O port.

Connect the analyzer and the external handler as shown in Figure 6-2.



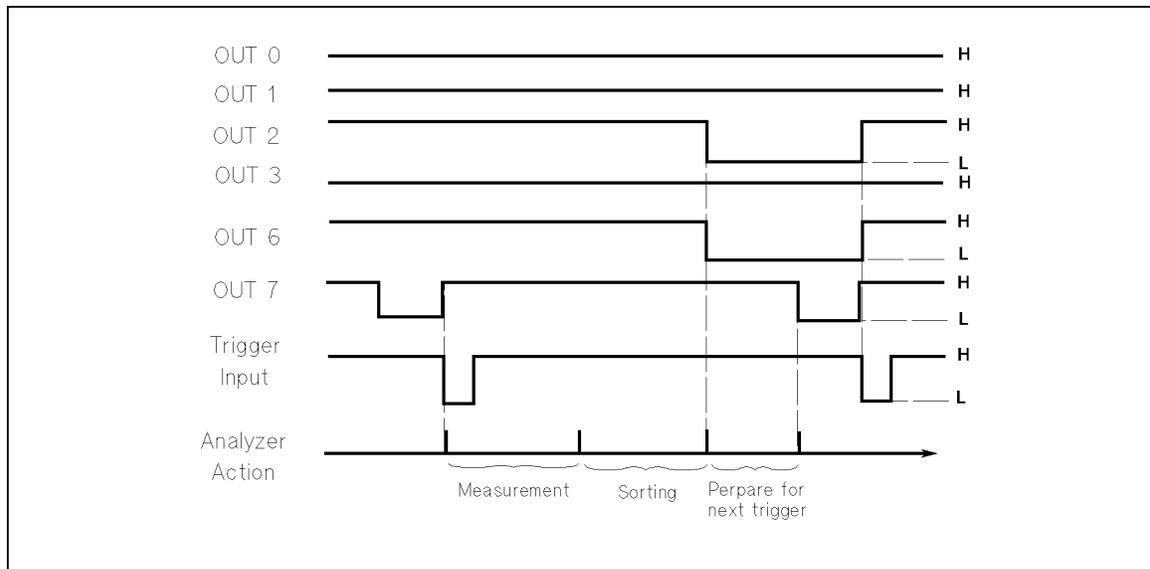
CE200602

Figure 6-2. Connecting I/O Port

Each output pin is set to TRUE if it satisfies the following conditions:

- OUT 0 Measured C is in the range of 5 % to the reference.
- OUT 1 Measured C is in the range of 10 % to the reference.
- OUT 2 Measured C is 10 % greater than the reference.
- OUT 3 Measured C is 10 % less than the reference.
- OUT 4 to 5 Not assigned
- OUT 6 Analyzer finished measurement and BIN sorting.
- OUT 7 Analyzer ready to accept a trigger.

The external handler can monitor OUT 6 and 7. When OUT 6 is set to TRUE, the handler reads the BIN result and sorts out the DUT. Then, it sets the next DUT to the measurement stage. The external handler can apply the trigger signal when OUT 7 is set to TRUE.



C6306003

Figure 6-3. Timing Chart

Figure 6-3 shows the timing chart for this program. Note that the I/O port is negative logic, therefore, TRUE is low and FALSE is high. This can be change by setting.

The flow of this program is as follows:

1. Set up the measurement.
2. Prepare the SRQ interrupt for a sweep end detection.
3. Wait for a trigger input. Sets OUT 7 to TRUE.
4. When the trigger is applied, start measuring the DUT. Set all pins to FALSE.
5. Sort the measured result.
6. Output the result of the BIN sorting and set OUT 6 to TRUE.
7. Return to 2.

This program pauses when in the wait for trigger state. Apply the RUN/CONT signal instead of the external trigger.

Disk



This program is included in the sample program disk. Its filename is BINSORT. This program is initialized for the Instrument BASIC. There is no program for the external controller in the *Sample Program Disk*.

```

100 !**** INITIALIZE ****
110 ASSIGN @Hp4291 TO 800
120 Scode=8
130 ABORT Scode
140 CLEAR @Hp4291
150 OUTPUT @Hp4291;"DISP:ALL HIHB"
160 !
170 !*** CONSTANTS ****
180 Cref1=2.E-12 ! C Reference = 10pF

```

```

190 Count=1
200 Nop=801      ! NUMBER OF POINTS
210 Cent=1.E+8  ! CENTER 100 MHz
220 Span=0      ! SPAN 0 Hz (Zero Span)
230 !
240 !*** BIN COUNTER INITIAL VALUE ****
250 Bin=0
260 Bin1=0! C:+-5%
270 Bin2=0! C:+-10%
280 Bin3=0! C:>+10%
290 Bin4=0! C:<-10%
300 !
310 !*** MEASUREMENT SETUP ****
320 OUTPUT @Hp4291;"SENS:FREQ:CENT ";Cent
330 OUTPUT @Hp4291;"SENS:FREQ:SPAN ";Span
340 OUTPUT @Hp4291;"SENS:SWE:POIN ";Nop
350 OUTPUT @Hp4291;"CALC:FORM CP"
360 !
370 !*** TRIGGER SETUP ****
380 OUTPUT @Hp4291;"TRIG:SOUR BUS"
390 OUTPUT @Hp4291;"TRIG:EVEN:TYPE POIN"
400 OUTPUT @Hp4291;"INIT:CONT ON"
410 !
420 !*** SRQ SETUP FOR TRIGGER ON POINT ****
430 ON INTR Scode GOTO Swp_end
440 OUTPUT @Hp4291;"STAT:INST:ENAB 128"
450 OUTPUT @Hp4291;"*SRE 4"
460 !
470 !*** MEASUREMENT START ****
480 DISP ""
490 Start:!
500 !
510 !***** COUNTER *****
520 Point=Count MOD Nop
530 IF Point=0 THEN Point=Nop
540 !
550 !***** TRIGGER *****
560 OUTPUT @Hp4291;"*CLS;*OPC?"
570 ENTER @Hp4291;0pc
580 WRITEIO 15,0;BINIOR(Bin,2^7)
590 PAUSE
600 ENABLE INTR Scode;2
610 WRITEIO 15,0;0
620 TRIGGER @Hp4291
630 Wait_swp:GOTO Wait_swp
640 Swp_end:!
650 !
660 !*** GET MEASURED C DATA ***
670 Data_out_q:!
680 OUTPUT @Hp4291;"TRAC:VAL? DTR,";Point
690 ENTER @Hp4291;Cmeas
700 C10=Crefl*.1
710 C5=Crefl*.05
720 Cdelta=Cmeas-Crefl
730 IF ABS(Cdelta)>C10 THEN

```

```
740     IF SGN(Cdelta) THEN
750         Bin=2^2
760         Bin3=Bin3+1
770     ELSE
780         Bin=2^3
790         Bin4=Bin4+1
800     END IF
810 ELSE
820     IF ABS(Cdelta)<=C5 THEN
830         Bin=2^0
840         Bin1=Bin1+1
850         GOTO Eom
860     ELSE
870         IF ABS(Cdelta)<=C10 THEN
880             Bin=2^1
890             Bin2=Bin2+1
900             GOTO Eom
910         END IF
920     END IF
930 END IF
940 !
950 Eom: !
960 PRINT TABXY(1,1),"COUNT:";Count
970 PRINT TABXY(1,2),"BIN:";Bin
980 Count=Count+1
990 WRITEIO 15,0;BINIOR(Bin,2^6)
1000 GOTO Start
1010 !
1020 Done: !
1030 END
```

Using the User Traces

The analyzer has a user definable trace feature. If you want to display data using custom formats or parameters that are not provided in the analyzer, you can display any x-axis or y-axis format that you specify. This chapter describes user traces and how to use them.

This chapter contains:

- What's the user trace?
- Using a user trace
- Sample program: time characteristic measurement

What's the User Trace?

The user trace function of the analyzer has the following features:

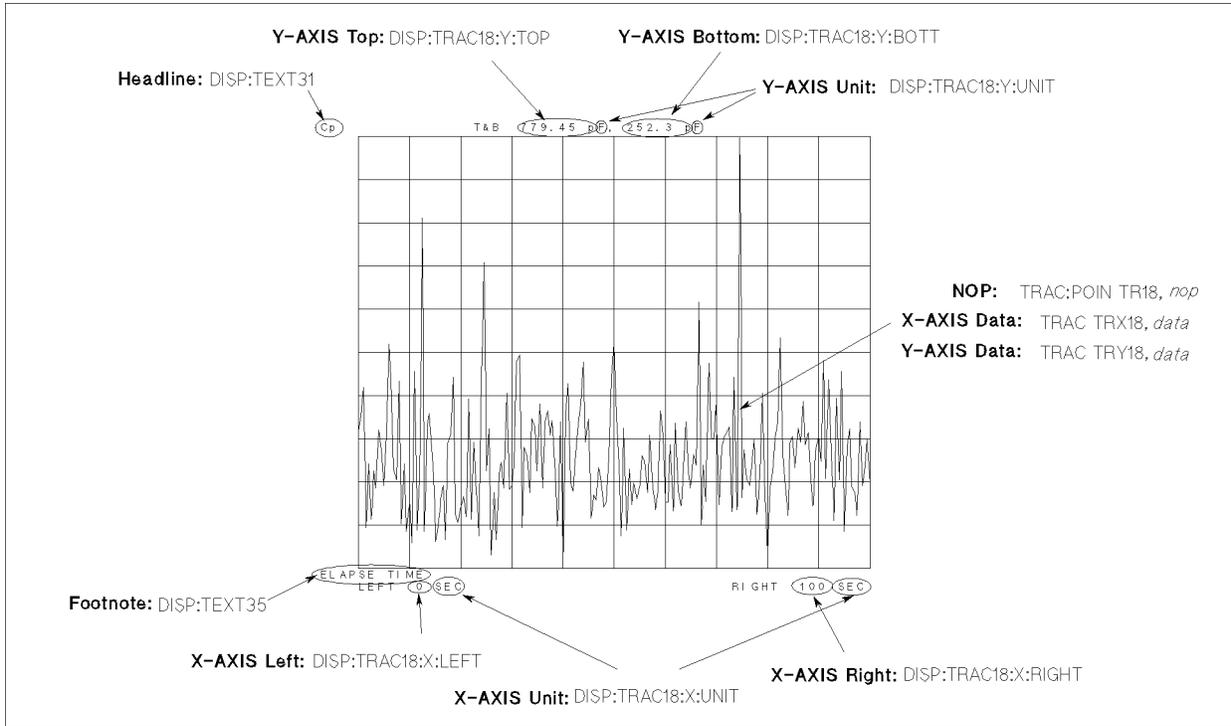
- You can display up to 4 arbitrary traces.
- You can independently define x-axis and y-axis units, labels, and scales for each trace.
- Marker functions can be used on the user traces.

For example, when you want to display the time characteristics of an impedance, you can first define the x-axis as time and the y-axis as impedance. Then display the trace in that format. This feature allows you to obtain the trace you want last on the display.

You cannot display the user trace and the normal measurement display simultaneously. When you are displaying the user trace, the measurement screen disappears, but the measurement function is still active. Therefore, you can make a measurement in the background while displaying a user trace.

Using a User Trace

Figure 7-1 shows the user trace parameters that you can specify and the related GPIB commands for user trace 1.



C6307001

Figure 7-1. User Trace

You can define four user traces as described previously. Each trace can have different settings for the unit, footnote, headline, and scale. The GPIB commands for user trace 1 are listed in Figure 7-1. Table 7-1 lists the corresponding commands for user traces 2 to 4.

Table 7-1. GPIB Commands for User Trace 1 to 4

User Trace	Unit & Scale	NOP	X-axis Data	Y-axis Data	Headline	Footnote
1	DISP:TRAC18 ...	TRAC:POIN TR18	TRAC TRX18	TRAC TRY18	DISP:TEXT31	DISP:TEXT35
2	DISP:TRAC19 ...	TRAC:POIN TR19	TRAC TRX19	TRAC TRY19	DISP:TEXT32	DISP:TEXT36
3	DISP:TRAC20 ...	TRAC:POIN TR20	TRAC TRX20	TRAC TRY20	DISP:TEXT33	DISP:TEXT37
4	DISP:TRAC21 ...	TRAC:POIN TR21	TRAC TRX21	TRAC TRY21	DISP:TEXT34	DISP:TEXT38

To display a user trace, perform the following procedure:

1. Set a grid for x-axis and y-axis.
2. Set data for the user trace.
3. Turn the user trace on.
4. When the operation is completed, delete the user trace.

7.2 Using the User Traces

Setting A Grid

First define the grid specifications for the data of the user trace. The specifications for the grid are divided into x-axis and y-axis.

For the x-axis, set the unit, the left edge value (start), the right edge value (stop), and the footnote.

OUTPUT @Hp4291;"DISP:TRAC18:X:UNIT 'SEC'"	<i>Define X-axis unit.</i>
OUTPUT @Hp4291;"DISP:TRAC18:X:LEFT 0"	<i>Define min. edge of the x-axis of user trace.</i>
OUTPUT @Hp4291;"DISP:TRAC18:X:RIGH 100"	<i>Define max. edge of the x-axis of user trace.</i>
OUTPUT @Hp4291;"DISP:TEXT35 'ELAPSE TIME'"	<i>Put a footnote.</i>

Module 7-1. Setting Grid For X-Axis

For the y-axis, set the unit, the top and bottom values for the scale, and the headline.

OUTPUT @Hp4291;"DISP:TRAC18:Y:UNIT 'F'"	<i>Define Y-axis unit.</i>
OUTPUT @Hp4291;"DISP:TRAC18:Y:BOTT 10E-12"	<i>Define min. edge of the y-axis of user trace to 10 pF.</i>
OUTPUT @Hp4291;"DISP:TRAC18:X:TOP 200E-12"	<i>Define max. edge of the y-axis of user trace to 200 pF.</i>
OUTPUT @Hp4291;"DISP:TEXT31 'Cp'"	<i>Put a headline.</i>

Module 7-2. Setting Grid For Y-Axis

■ Related GPIB Commands

You can toggle the linear or log scale for the x and y-axis:

DISP:TRAC{18-21}:Y:SPAC LIN	Set y-axis to the linear scale. (Default)
DISP:TRAC{18-21}:Y:SPAC LOG	Set y-axis to the log scale.
DISP:TRAC{18-21}:X:SPAC LIN	Set x-axis to the linear scale. (Default)
DISP:TRAC{18-21}:X:SPAC LOG	Set x-axis to the log scale.

The default grid setting of the four user traces is common. To define each trace's unit, headline, footnote, and scale individually, turn off the grid coupling by using the following command:

DISP:TRAC{18-21}:GRAT:AXIS:COUP OFF Turning off the scale and grid coupling.

Setting Data Train for The Trace

After you define a grid specification, enter the x-axis and the y-axis data into the user trace's data arrays. This enables the analyzer to display the user trace.

You must ensure that the number of points and number of x-axis and y-axis data are equal. Otherwise, if it is greater or less than equal, an GPIB error occurs.

OUTPUT @Hp4291;"TRAC:POIN TR18,201"	<i>Defines Number of points.</i>
OUTPUT @Hp4291;"TRAC TRY18,";Y_data(*)	<i>Sets Y-Axis Values.</i>
OUTPUT @Hp4291;"TRAC TRX18,";X_data(*)	<i>Sets X-Axis Values.</i>

Module 7-3. Setting Data Train for X- and Y-axis

You can change scale specifications after setting x-axis and y-axis trace data. The following module sets the optimized scale setting for the user trace.

```
OUTPUT @Hp4291;"DISP:TRAC18:Y:AUTO ONCE" Adjust the scale to fit the trace in the grid. TRAC18 is for the user trace 1.
```

Module 7-4. Automatic Scaling for User Trace

■ Related GPIB Commands

You can copy the data or memory trace data of the measurement display to the user trace array. At this time, the number of points is reset to the same setting as the measurement display.

TRAC:COPY TR{18-21},TR1 Copies data trace data, number of points, stimulus, and units for x and y-axis of the active channel to user trace 1 to 4, respectively.

TRAC:COPY TR{18-21},TR2 Copies selected memory trace data, number of points, stimulus, and units for x and y-axis of the active channel to user trace 1 to 4.

Turning ON the User Trace

When you finish setting the user trace parameters, turn it on to display the user trace.

```
OUTPUT @Hp4291;"DISP:TRAC18:STAT ON" Turn on the user trace.
```

Module 7-5. Turning ON the User Trace

Using the Marker on a User Trace

You can use the marker on a user trace just as it is used on the data trace. The following module puts the marker on the user trace 1.

```
OUTPUT @Hp4291;"CALC:EVAL:ON 'TR18'" Sets the marker to work on the user trace 1.
```

Module 7-6. Using Marker on the User Trace

Clearing a User Trace

When you finish using a user trace, you should release the allocated memory for the use of other functions. When you clear the user trace, the user trace display is automatically turned off.

```
OUTPUT @Hp4291;"DISP:TRAC18:CLE" Clear all user trace data and turn off the user trace display.
```

Module 7-7. Turning OFF and Clearing The User Trace

Sample Program: Time Characteristic Measurement

This sample program measures the time characteristics of a capacitor at a fixed frequency. It then displays the trace on a time domain grid by using the user trace function.

The flow of this program is as follows:

1. Initialize the program.
2. Define constants.
3. Prepare x-axis data array for a time domain grid of user trace.
4. Set up for measurement.
5. Set up trigger.
6. Make a time interval measurement at fixed frequency by using zero span and queries measured data.
7. Set up x-axis grid specifications for user trace.
8. Set up y-axis grid specifications for user trace.
9. Enter x and y-axis data for the user trace array.
10. Turn on the user trace.

For the x-axis of the user trace, the program puts the elapse time from 0 at intervals of 0.5 seconds (line 220 to 250).

Disk



This program is included in the sample program disk. Its filename is `USR_TRAC`. This program is initialized for Instrument BASIC. The program for the external controller, `USRTRAC_E`, is also included in the *Sample Program Disk*.

```

100 !**** INITIALIZE ****
110 ASSIGN @Hp4291 TO 800
120 Scode=8
130 CLEAR @Hp4291
140 ABORT Scode
150 !
160 !**** CONSTANTS ****
170 Tint=.5 ! INTERVAL TIME (SEC)
180 Nop=201 ! SAMPLE POINTS
190 F=1.E+8 ! MEASUREMENT FREQ. 100MHZ
200 !
210 !**** ARRAY DEFINITION ****
220 DIM X(1:201),Y(1:201)
230 FOR I=1 TO Nop ! PREPARING X-AXIS DATA
240   X(I)=(I-1)*Tint
250 NEXT I
260 !
270 !**** MEASUREMENT SETUP ****
280 OUTPUT @Hp4291;"SENS:FREQ:SPAN 0;CENT ";F
290 OUTPUT @Hp4291;"CALC:MATH:STAT OFF"
300 OUTPUT @Hp4291;"CALC:FORM CP"
310 !
320 !**** TRIGGER SETUP ****

```

```

330 OUTPUT @Hp4291;"STAT:INST:ENAB 128"
340 OUTPUT @Hp4291;"*SRE 4"
350 ON INTR Scode GOTO Sweep_end
360 OUTPUT @Hp4291;"TRIG:SOUR BUS"
370 OUTPUT @Hp4291;"INIT:CONT ON;"
380 OUTPUT @Hp4291;"TRIG:EVEN:TYPE POIN"
390 !
400 !**** TIME-INTERVAL MEASUREMENT ****
410 T1=TIMEDATE
420 FOR I=1 TO Nop
430   OUTPUT @Hp4291;"*CLS;*OPC?"
440   ENTER @Hp4291;0pc
450   ENABLE INTR Scode;2
460   TRIGGER @Hp4291
470   Waiting:GOTO Waiting
480   Sweep_end:!
490   OUTPUT @Hp4291;"TRAC:VAL? DTR,";I
500   ENTER @Hp4291;Y(I)
510   DISP (I-1)*Tint;TAB(7);" [SEC]"
520   REPEAT           ! CHECKING INTERVAL TIME.
530     T2=TIMEDATE   ! IF INTERVAL EXCEEDS
540     UNTIL T2>T1+Tint ! LOOP WILL EXIT.
550     T1=T2
560   NEXT I
570 !
580 !**** SETTING GRID FOR X-AXIS ****
590 OUTPUT @Hp4291;"DISP:TRAC18:X:UNIT 'SEC'"
600 OUTPUT @Hp4291;"DISP:TRAC18:X:LEFT ";X(1)
610 OUTPUT @Hp4291;"DISP:TRAC18:X:RIGHT ";X(Nop)
620 OUTPUT @Hp4291;"DISP:TEXT35 'ELAPSE TIME'"
630 !
640 !**** SETTING GRID FOR Y-AXIS ****
650 OUTPUT @Hp4291;"DISP:TRAC18:Y:UNIT 'F'"
660 OUTPUT @Hp4291;"DISP:TRAC18:Y:BOTT ";MIN(Y(*))
670 OUTPUT @Hp4291;"DISP:TRAC18:Y:TOP ";MAX(Y(*))
680 OUTPUT @Hp4291;"DISP:TEXT31 'Cp'"
690 !
700 !**** SETTING X AND Y-AXIS DATA ****
710 OUTPUT @Hp4291;"TRAC:POIN TR18,";Nop
720 OUTPUT @Hp4291;"TRAC TRX18,";X(*)
730 OUTPUT @Hp4291;"TRAC TRY18,";Y(*)
740 !
750 !**** TURNING ON USER TRACE ****
760 OUTPUT @Hp4291;"DISP:TRAC18:STAT ON"
770 !
780 !**** TURNING ON MARKER ****
790 OUTPUT @Hp4291;"CALC:EVAL:ON 'TR18'"
800 OUTPUT @Hp4291;"CALC:EVAL:INT OFF"
810 !
820 !**** CLEARING USER TRACE ****
830 INPUT "MOVE MARKER, OR PRESS [RETURN] TO CLEAR TRACE",Ans$
840 OUTPUT @Hp4291;"DISP:TRAC18:CLE"
850 !
860 END

```

Programming Miscellaneous

This chapter provides information not directly concerned with measurements, but useful for programming. The following topics are described:

- Using Disks
- Printing
- Controlling Instrument BASIC from an External Controller
- Debugging Program
- Processing Time Measurement
- Key Sequence Logging
- File transfer function

Using Disks

The analyzer has an floppy disk drive and a memory disk. You can use them from the controller by using one of the following:

- GPIB commands:
This method is the same as the front panel operation. You can use this method for saving or loading the analyzer settings.
- Instrument BASIC statements:
This method directly accesses the disks from the Instrument BASIC controller. This method is only available on the Instrument BASIC.

Saving the Analyzer Status

You can remotely save the analyzer status to the floppy disk or memory disk using the GPIB command, `MMEM:STOR:STAT`.

<pre>INPUT "Enter File Name (w/o ext.)",File\$ OUTPUT @Hp4291;"MMEM:STOR:STAT """;File\$;"","","DISK"";"</pre>	<p><i>Asks you to enter the filename to save.</i></p> <p><i>Saves analyzer state to the floppy disk.</i></p> <p><i>Use MEMORY instead of DISK to save to memory disk.</i></p>
--	---

Module 8-1. Saving the Analyzer Status

- Related GPIB Command

You can also save the trace data by using the following commands:

`MMEM:STOR:TRAC SEL ,filename` Save selected trace data.

`MMEM:STOR:ITEM:TRAC:SEL {RAW|DATA|DTR}` Select item for `MMEM:STOR:TRAC` .

`MMEM:STOR:ITEM:TRAC:DEL {RAW|DATA|DTR}` Deselect item for `MMEM:STOR:TRAC` .

You can load an analyzer state or data by using the following commands:

MMEM:LOAD:STAT *filename* Load the analyzer status.

MMEM:LOAD:TRAC SEL,*filename* Load selected trace data.

Note

A storage device for the GPIB commands, MMEM:LOAD or MMEM:STOR is independent of the "MSI" setting of Instrument BASIC.



Entering Trace Data From the Disk into a Program Variable

This example reads the trace data file that is saved from the front panel keys. For details about the saved file format, see the *Operation Manual*.

To prepare the trace data file:

1. Insert Disk.
2. Press **Save**.
3. Toggle to **STOR DEV [DISK]**.
4. Press **DEFINE SAVE DATA**, then turn **DATA TRACE** to **ON** and all other data are **OFF**.
5. Press **SAVE BINARY**.
6. Enter file name, then press **DONE**.

ON ERROR GOTO Err	<i>If file does not exist, go to label Err.</i>
INPUT "Enter File Name",File_name\$	<i>Asks you to enter the filename to get data.</i>
MSI ":INTERNAL"	<i>Selects floppy disk drive. To select RAM disk, use ":MEMORY".</i>
DIM Dat1(1:201,1:2),Dat2(1:201,1:2)	<i>Defines the data array for storing the trace data.</i>
ASSIGN @File TO File_name\$	<i>Open the file.</i>
ENTER @File USING "#,17X"	<i>Skip file header.</i>
!**** FOR CH1 DATA ****	
ENTER @File USING "#,6X"	<i>Skips a reserved header.</i>
ENTER @File;Dat1(*)	<i>Reads Trace Data of channel 1.</i>
ENTER @File USING "4X,#"	<i>Skips delimiter.</i>
!**** FOR CH2 DATA ****	
ENTER @File USING "#,6X"	<i>Skips a reserved header.</i>
ENTER @File;Dat2(*)	<i>Reads Trace Data of channel 2.</i>
ENTER @File USING "4X,#"	<i>Skips delimiter.</i>
ASSIGN @File TO *	<i>Closes file.</i>
GOTO Done	<i>Skip error routine.</i>
Err: PRINT "File not found."	
Done: !	

Module 8-2. Entering Data-Trace Data From Disk into Variable

Sample Program: Making HP CITIfile

The following sample program allows to create an HP CITIfile format file that is used to exchange the data between different computers and instruments. "CITIfile" stands for "Common Instrumentation Transfer and Interchange file." The HP CITIfile is a plain ASCII format that contains some instrument dependent setting information and measurement data.

The program flow is as follows:

1. Measures a reflection coefficient as S_{11} data of 1-port device.
2. Puts measured reflection data in order of real and imaginary part.
3. Identifies LIF or DOS to create a proper ASCII file on both format disk.
4. Creates a file. CREATE is used for DOS and CREATE ASCII is used for LIF.
5. Stores the header and data into the file.

Any points trace (2 to 801 points) is accepted without using dynamic re-dimensioning the array (REDIM). This is accomplished by specifying the image, "%,K", when reading a measurement data. This technique is useful especially for the Instrument BASIC that is not support REDIM statement.

The following shows the example of the HP CITIfile format file. The head 7 lines called *Header* include the setup information and CITIfile keywords. The lines after BEGIN called *Data* contain a measurement data.

```
CITIFILE A.01.00
NAME DATA
VAR FREQ MAG 201
DATA S[1,1] RI
SEG_LIST_BEGIN
SEG 1.E+8 1.E+9 201
SEG_LIST_END
BEGIN
  7.3690E-04,-2.8451E-02
  7.1558E-04,-2.7074E-02
  6.7095E-04,-2.5867E-02
  6.4325E-04,-2.4910E-02
  :
```

Figure 8-1. Example of HP CITIfile

Disk



This program is included in the sample program disk. Its filename is CITI. This program is initialized for Instrument BASIC. The program for the external controller, CITI_E, is also included in the *Sample Program Disk*. Touchstone format file generation program is also included in the sample program disk as TOUCH for Instrument BASIC and TOUCH_E for the external controller.

```
100 ASSIGN @Hp4291 TO 800
110 Scode=8
120 CLEAR @Hp4291
130 ABORT Scode
140 !
150 DIM File$[16],Cat$(1:200)[150],Str$[35]
```

```

160 DIM Refl(1:801,1:2),Freq(1:801)
170 !
180 OUTPUT @Hp4291;"CALC:MATH:NAME RCO;STAT ON"
190 DISP "Getting data..."
200 OUTPUT @Hp4291;"SENS:SWE:POIN?"
210 ENTER @Hp4291;Nop
220 PRINT "NOP=";Nop
230 OUTPUT @Hp4291;"CALC:FORM COMP"
240 OUTPUT @Hp4291;"DISP:TRAC:GRAT:FORM SMIT"
250 OUTPUT @Hp4291;"TRAC? DTR"
260 ENTER @Hp4291 USING "%,K";Refl(*)
270 OUTPUT @Hp4291;"DATA? SPAR"
280 ENTER @Hp4291 USING "%,K";Freq(*)
290 !
300 ! CHECK DISK FORMAT
310 Lif=1
320 DISP "Now Checking Disk Format..."
330 CAT TO Cat$(*)
340 IF Cat$(1)[1,9]="DIRECTORY" THEN Lif=0
350 !
360 PRINT "Write S11 data as HP CITIfile format file."
370 INPUT "Enter file name to save?",File$
380 PRINT "File: """;File$;""",
390 DISP "Creating file """;File$;"""...
400 Size=INT((Nop+2)*36/256+1) ! 33 char/line
410 IF Lif=0 THEN
420     CREATE File$,Size
430     PRINT "FORMAT:DOS"
440 ELSE
450     CREATE ASCII File$,Size ! 1 block = 256 byte
460     PRINT "FORMAT:LIF"
470 END IF
480 ASSIGN @Disk TO File$;FORMAT ON
490 OUTPUT @Disk;"CITIFILE A.01.00"
500 OUTPUT @Disk;"NAME DATA"
510 OUTPUT Str$ USING "13A,K,#";"VAR FREQ MAG ";Nop
520 OUTPUT @Disk;Str$
530 OUTPUT @Disk;"DATA S[1,1] RI" ! CITIfile DATA support only "RI" format
540 OUTPUT @Disk;"SEG_LIST_BEGIN"
550 OUTPUT Str$ USING "4A,K,A,K,A,K,#";"SEG ";Freq(1);" ";Freq(Nop);" ";Nop
560 OUTPUT @Disk;Str$
570 OUTPUT @Disk;"SEG_LIST_END"
580 OUTPUT @Disk;"BEGIN"
590 DISP "Output S11 data to disk..."
600 FOR I=1 TO Nop
610     OUTPUT Str$ USING "MD.4DESZZ,A,MD.4DESZZ,#";Refl(I,1);",";Refl(I,2)
620     OUTPUT @Disk;Str$
630 NEXT I
640 OUTPUT @Disk;"END"
650 ASSIGN @Disk TO *
660 DISP "Writing Finished."
670 BEEP
680 !
690 END

```

Printing

You can print a hard copy of the display to the printer using GPIB commands.

To Print Analyzer Display

Printer Preparation

1. Connect a printer using a parallel cable.
2. Turn the printer on.

Execute Print

To print the screen, execute the following command.

```
OUTPUT 800;"HCOP"
```

Set the GPIB address when you execute from an external controller.

To Observe Printing

The HP Basic program shown below gives an example to detect printing end by using a SRQ interrupt.

```
10  !
20  !  To Observe Printing
30  !
40  ASSIGN @Hp4291 TO 800
50  !
60  OUTPUT @Hp4291;"*CLS"
70  OUTPUT @Hp4291;"STAT:OPER:NTR 512"  !Catch High to Low Transition
80  OUTPUT @Hp4291;"STAT:OPER:PTR 0"    !Disable Low to High Transitions
90  OUTPUT @Hp4291;"STAT:OPER:ENAB 512" !Enable OS Event Reg.
100 OUTPUT @Hp4291;"*SRE 128"          !Enable OSR bit
120 ON INTR 8 GOTO La1
130 ENABLE INTR 8;2
140 OUTPUT @Hp4291;"HCOP"
150 Wait_print: GOTO Wait_print
160 La1:!
170 DISP "PRINT COMPLETE"
180 !
190 END
```

Figure 8-2. Sample Program : To Observe Printing

Controlling Instrument BASIC from an External Controller

The analyzer provides the PROGRAM subsystem command set that allows control of Instrument BASIC from an external controller. The PROGRAM subsystem can do the following:

- Read or put the variable data of an Instrument BASIC program from the external controller.
- Control Instrument BASIC execution state.
- Download or upload the Instrument BASIC program.

Reading or Putting the Variable Data

You can get from or put into the contents of a variable in the Instrument BASIC program from an external controller. This section provides the following example modules:

- Reading numeric variable of Instrument BASIC.
- Putting numeric variable of Instrument BASIC.
- Reading string variable of Instrument BASIC.
- Putting string variable of Instrument BASIC.

Reading Numeric Variable

To read the contents of a numeric variable, perform the following steps:

OUTPUT @Hp4291;"PROG:NUMB? ""Dat"";"	<i>Query contents with variable name.</i>
ENTER @Hp4291;Dat	<i>Retrieve contents of Dat.</i>

Module 8-3. Reading Numerical Variable

For an array variable, use the same process:

DIM Dat(1:201,1:2)	<i>Define the same size array as the Instrument BASIC's array.</i>
OUTPUT @Hp4291;"PROG:NUMB? ""Dat"";"	<i>Query contents with array name.</i>
ENTER @Hp4291;Dat(*)	<i>Retrieve contents of Dat.</i>

Module 8-4. Reading Numerical Array

Note You cannot read array data by the array element.



Putting Numeric Variable

You can put an arbitrary value into a variable of Instrument BASIC by specifying the variable name and sending data:

OUTPUT @Hp4291;"PROG:NUMB ""Dat"";";1	<i>Puts 1 into the Instrument BASIC variable, Dat.</i>
---------------------------------------	--

Module 8-5. Putting Numerical Data into Variable

You can also put an array data value into an array of Instrument BASIC:

```
OUTPUT @Hp4291;"PROG:NUMB ""Dat"";";Array(*)  Transfers data arrays into the Instrument BASIC array, Dat
```

Module 8-6. Putting Numerical Data into Array

Reading String Variable

You can read the string data of Instrument BASIC in a similar way.

```
OUTPUT @Hp4291;"PROG:STR? ""Dat$"";"  Query the contents of Dat$ string variable.  
ENTER @Hp4291;Dat_receive$  Retrieve string into string variable.
```

Module 8-7. Reading String Variable

Putting String Variable

You can put the string data to an Instrument BASIC variable from an external controller by specifying the string name and sending the data:

```
OUTPUT @Hp4291;"PROG:STR ""Str$"",""TEST1""  Puts string "TEST1" into Str$.
```

Module 8-8. Putting String Variable (1)

To use a string variable for sending data:

```
File$="TEST1"  
OUTPUT @Hp4291;"PROG:STR ""Str$"",""";File$;""""  Puts content of File$ into Str$.
```

Module 8-9. Putting String Variable (2)

Simultaneously Running Instrument BASIC and External Controller Programs

You may want to use both controllers simultaneously (that is, Instrument BASIC and an external controller). When running programs on both controllers, the synchronization of both programs is a serious issue. The PROGram subsystem commands that allow you to know or control the Instrument BASIC running status are RUN, PAUS, CONT, or STOP.

Controlling Instrument BASIC Execution Status

Using the PROG:STAT command, you can control the Instrument BASIC running state.

```
OUTPUT @Hp4291;"PROG:STAT RUN" Run the Instrument BASIC program.
```

Module 8-10. Controlling Instrument BASIC Execution Status

Determining Instrument BASIC Execution State

You can determine the status of the BASIC program from an external controller using the following methods:

- Query Instrument BASIC execution state using the PROG:STAT? query command

The PROG:STAT? query returns the status as strings. By checking the query response, you can determine if the program status is RUN, PAUS, or STOP.

```
OUTPUT @Hp4291;"PROG:STAT?"
ENTER @Hp4291;Stat$
DISP Stat$
```

Module 8-11. Determining Instrument BASIC Execution State

- Check the "Program Running" bit

Bit 14 of the operation status register represents the running status of Instrument BASIC. When the program is running, 1 is set in this bit. When the program is stopped, or paused, 0 is set. You can check the status of Instrument BASIC by querying this bit, or generating SRQ by the condition of this bit.

You can generate an SRQ on RUN, or PAUSE and STOP by setting the transition filter. Setting the positive transition filter generates SRQ on RUN. Setting the negative transition filter generates the SRQ on PAUSE or STOP.

```
ON INTR Scode GOTO Paused When Instrument BASIC is paused, jumps to label, Paused:.
OUTPUT @Hp4291;"*CLS;*OPC?"
ENTER @Hp4291;0pc
OUTPUT @Hp4291;"STAT:OPER:ENAB 16384;*SRE 128" Set SRQ enable register
OUTPUT @Hp4291;"STAT:OPER:PTR 0;NTR 16384" Set the negative transition filter.
ENABLE INTR Scode;2 Enable interrupt.
Wait_paused:GOTO Wait_paused
Paused:!
```

Module 8-12. Generating SRQ on PAUSED

- Use the PROG:WAIT? GPIB query command

The `PROG:WAIT?` query returns 1 when the Instrument BASIC program is stopped or paused. Retrieving the return value by using the `ENTER` statement causes the external controller execution to wait until the Instrument BASIC program is stopped or paused.

```
OUTPUT @Hp4291;"PROG:WAIT?"  
ENTER @Hp4291;Dummy
```

Module 8-13. Using `PROG:WAIT?` GPIB Query

Transferring Program Source

You can transfer the program between Instrument BASIC and the external controller by using one of the following procedures:

- Uploading a program from controller to Instrument BASIC.
- Downloading a program from Instrument BASIC to the controller.

Uploading a Program from Controller to Instrument BASIC

This example transfers the program file on the external controller disk into Instrument BASIC. This is useful for loading the same program into multiple analyzers on the GPIB.

OUTPUT @Hp4291;"PROG:DEL:ALL"	<i>Clears Instrument BASIC program before uploading.</i>
INPUT "Enter File Name to Transfer.",File\$	<i>Ask a filename of program to transfer.</i>
ASSIGN @File TO File\$;FORMAT ON	<i>Open a program file. FORMAT ON indicates a text file.</i>
ON ERROR GOTO Done	<i>When the end of file, goto label, Done:.</i>
DIM LINE\$[100]	<i>Allocate 100 characters per line.</i>
OUTPUT @Hp4291;"PROG:DEF #0"	<i>#0 indicates to send a program until EMD.</i>
LOOP	
Line\$=""	
ENTER @File;Line\$	<i>Read 1 line from file.</i>
OUTPUT @Hp4291;Line\$	<i>Send 1 line to @Hp4291.</i>
END LOOP	
Done:!	<i>Repeat until end of file.</i>
OUTPUT @Hp4291;"";END	<i>Complete transfer by sending terminator.</i>
ASSIGN @FILE TO *	

Module 8-14. Uploading

Downloading a program from Instrument BASIC to the Controller

The following example downloads the Instrument BASIC program into the string array of the external controller program.

DIM PROG\$(1:3000)[100]	<i>Allocate variable 3000 lines maximum.</i>
ON ERROR GOTO Done	<i>If queried program is smaller or larger than expected, interrupted to the label, Done:.</i>
ENABLE INTR Scode;2	<i>interrupted to the label, Done:.</i>
OUTPUT @Hp4291;"PROG:DEF?"	<i>Query Instrument BASIC program.</i>
ENTER @Hp4291 USING "#,4A";Head\$	<i>Receive header.</i>
ENTER @Hp4291;Prog\$(*)	<i>Receive program body.</i>
Done:!	<i>When all program is received, jump to this line by interrupt.</i>

Module 8-15. Downloading

Debugging Program

When you're running your program, the analyzer sometimes reports the error message on the display. The error can be caused by conflicting settings, invalid operations, or wrong commands, and so on. Your program may have a bug that causes an error to be displayed on the analyzer.

The analyzer also has an error queue that stores the error message strings and error numbers of errors that occur while operating the analyzer. If you encounter an error message like, CAUTION: xxxxxxxx, you can check the error queue to debug your program.

The error messages are stored in an FIFO (first in first out) error queue. You can read the error queue contents to determine which error occurred by using the following procedures.

The first module declares the error string variable for receiving the error message. It is recommended this module be put at the beginning of the program.

```
DIM Err$(50)           Defines String Variable for entering error string beforehand.
OUTPUT @Hp4291;"*CLS" Clears error queue.
```

Module 8-16. Checking GPIB Error (Initialize)

The second module reads and displays the error. This module should be put at the end of the program.

```
FOR I=1 TO 10
  OUTPUT @Hp4291;"SYST:ERR?" Queries the error queue.
  ENTER @Hp4291;Err$ Retrieve the error strings.
  PRINT Err Display the error strings.
NEXT I
```

Module 8-17. Checking GPIB Error (Reading Error)

This module reports the errors that are in the error queue. For example, if a wrong GPIB command is sent, the following message is displayed.

```
-113,"Undefined header"
:
:
```

You can check the meaning of the messages in **Error Messages**.

If the program is completed without any errors, the following message is displayed.

```
+0,"No error"
```

To clear the error queue, read all the error message in the error queue or execute the *CLS command.

This method reports only the first error that occurred in the program, but not its position in the program. To determine the error line, use step execution. You can find the **STEP** softkey that enables step execution under **(System) IBASIC**. By executing the program line by line with **STEP**, you can find the error line when the error message is displayed. Then, execute the following statements from the command line.

```
OUTPUT 800;"SYST:ERR?"
ENTER 800;Err$
DISP Err$
```

This displays the latest error message in the error queue. You can do this to get more specific information about an error.

If the program is too large to execute step by step, put PAUSE statements at appropriate positions in the program. If the error has not occurred before the program is paused, the program has no error before that PAUSE line. You can find the error position more efficiently using this technique.

Processing Time Measurement

You can measure the program processing time by using the real-time clock.

```
Start_time=TIMEDATE           Keep the start time.
:                             \
:                             Measurement Process
:                             /
End_time=TIMEDATE           Stores the end time.
Total=End_time-Start_time   Calculate the process time,
PRINT Total,"seconds of processing." then display it.
```

Module 8-18. Measuring Processing Time

Key Sequence Logging

The analyzer has a logging function that records the front-panel key operation and generates the equivalent program codes to the Instrument BASIC editor. While **[System] LOGGING** is set to **ON**, the Instrument BASIC statements that are equivalent to the key strokes are inserted into the Instrument BASIC editor.

Generating Equivalent Program For Empty Editor

If the Instrument BASIC editor is empty, the following statement is added to the head of the program:

```
ASSIGN @Hp4291 TO 800
```

and the **END** statement is added to the end of the program. As a result, the logging function records a working Instrument BASIC program that is equivalent to the front-panel key operation. You can run this created program by pressing the **RUN** softkey. You can also modify or save it as an Instrument BASIC program.

For example, the following procedure makes a program that sets the measurement parameter to the reflection coefficient, ($|\Gamma|$):

1. Press **[System]**, then toggle **LOGGING** to **ON** to start logging.
2. Press **[Meas] MORE ... REFL.COEF:MAG(| Γ |)** to select the reflection coefficient as a measurement parameter.
3. Press **[System]**, then toggle **LOGGING** to **OFF** to complete logging.

This operation generates the following Instrument BASIC program:

```
1 ASSIGN @Hp4291 TO 800
2 OUTPUT @Hp4291;"CALC:MATH:NAME RCO"
3 OUTPUT @Hp4291;"CALC:MATH:STAT ON"
4 OUTPUT @Hp4291;"CALC:FORM MLIN"
10 END
```

Inserting Equivalent Codes into Your Program

If there are any codes in the Instrument BASIC editor, **LOGGING ON** inserts the equivalent codes into the cursor position. This will help you develop a program. If you want to add a key operation code within your program, perform the following steps:

1. Move the cursor to the line in which you want to insert the key-operation codes.
2. Press **END EDIT** to exit from Instrument BASIC editor.
3. Press **[System]**, then toggle **LOGGING** to **ON** to start logging.
4. Perform the front-panel key operation you want to add to your program.
5. Press **[System]**, then toggle **LOGGING** to **OFF** to complete logging.

Limitations

- When an Instrument BASIC program is running, logging is ignored.
- If you make a mistake when logging is ON, the analyzer generates equivalent codes faithfully and records incorrect statement.
- The logging function does not truncate repeated nodes of the SCPI command. This means that sometimes the program is redundant.
- The logging function does not allow for timing sensitive operations such as triggering or a fixture compensation procedure. Therefore, you need to add or rewrite that part of a program to allow the time required for it to run correctly.
- If the logged codes exceed the limitation of editor memory, an error will occur.

File Transfer Function

This section describes how to use the file transfer function, showing you a sample program.

The file transfer function uses the external controller to transfer files between the selected storage device of this instrument (memory disk or diskette) and an external storage device (such as hard disk). This function allows you to:

- Directly access data you want to use on the external controller.

For example, you can transfer the file of an instrument screen to the external controller, print it on a printer connected to the external controller, and paste it onto a file in a word processor running on the external controller.

- Use external storage devices, which have larger capacity compared to the memory disk or a diskette.

For example, if there are a great number of measurement conditions which require calibration, the amount of the setting data becomes extremely large, including calibration data. In this case, it is impractical to store all of these settings on the memory disk or a single diskette at a time. However, you can realize this functionality by transferring them to the external controller and then storing them on an external storage device.

- Perform remote measurement using the external controller with a few GPIB commands for basic measurement. You do not have to memorize further details (such as GPIB commands used for detailed settings).

Preparation:

Use the keys on the front panel to establish the setting required for your measurement. Store it on the storage device of the 4291B, then transfer the file to the external controller, and store it on an external storage device. Repeat this procedure for all of the settings required for your measurement.

Measurement:

Choose a necessary setting file from those stored and transfer it to the 4291B using the external controller. Then, recall the file to set the 4291B for the measurement and perform the measurement using the GPIB commands.

The storage device of the 4291B allows you to handle files listed below in the DOS format or the LIF format. For DOS format files, both binary files and ASCII files can be transferred. For LIF format files, only binary files can be transferred.

- Binary files
 - Instrument settings and internal data array (STATE)
 - Internal data arrays (DATA ONLY binary)
 - Graphic images (GRAPHICS)
- ASCII files
 - Internal data arrays (DATA ONLY ascii)
 - HP instrument BASIC programs

File Transfer from 4291B to External Controller

This program transfers a specified file in the current directory of the 4291B to the current directory of the storage device connected to the external controller, giving a file name you desire.

When executed, this program first prompts you to enter a source file name, as shown below. Enter the name of a file you want to transfer.

```
ENTER SOURCE FILE NAME ON INSTRUMENT ?
```

Then, the program prompts you to enter a destination file name as shown below (in this example, SAMPLE.STA has been entered as the source file name). Enter the file name you want to give on the storage device. Note that a file with the same name will be overwritten, if it already exists.

```
ENTER SOURCE FILE NAME ON INSTRUMENT ?      SAMPLE.STA
ENTER DESTINATION FILE NAME ON CONTROLLER ?
```

Disk



This program is included in the sample program disk. Its filename is FM_INST. This is the program for the external controller.

```
100 DIM Src_file$[50],Dst_file$[50]
110 ASSIGN @Hp4291 TO 717
120 OUTPUT @Hp4291;"*rst"
130 !
140 PRINT "  ENTER SOURCE FILE NAME ON INSTRUMENT ?      ";
150 INPUT Src_file$
160 PRINT Src_file$
170 !
180 PRINT "  ENTER DESTINATION FILE NAME ON CONTROLLER ? ";
190 INPUT Dst_file$
200 PRINT Dst_file$
210 !
220 Copy_from_instr(@Hp4291,Src_file$,Dst_file$)
230 !
240 END
250 !
260 !  copy_from_instrument
270 !
280 SUB Copy_from_instr(@Hp4291,Src_file$,Dst_file$)
290   DIM Len$[6],Img$[32],Dmy$[2]
300   !
310   ON ERROR GOTO Skip_purge
320   PURGE Dst_file$
330 Skip_purge:  OFF ERROR
340   CREATE Dst_file$,1
350   ASSIGN @Dst_file TO Dst_file$
360   !
370   CLEAR @Hp4291
380   OUTPUT @Hp4291;"*CLS"
390   OUTPUT @Hp4291;"ROPEN """;Src_file$;""""
400   IF FNCheck_error(@Hp4291,"<CPFI: ropen>")=-1 THEN SUBEXIT
```

```

410      !
420      LOOP
430          OUTPUT @Hp4291;"READ?"
440          ENTER @Hp4291 USING "#,2A";Dmy$
450          ENTER @Hp4291 USING "#,6A";Len$
460          Block_size=VAL(Len$)
470          !
480          IF Block_size=0 THEN
490              ENTER @Hp4291 USING "%,A";Dmy$
500              ASSIGN @Dst_file TO *
510              OUTPUT @Hp4291;"CLOSE"
520              SUBEXIT
530          END IF
540          !
550          ALLOCATE Dat$[Block_size]
560          Img$="#,&VAL$(Block_size)&"A"
570          ENTER @Hp4291 USING Img$;Dat$
580          ENTER @Hp4291 USING "%,A";Dmy$
590          OUTPUT @Dst_file USING Img$;Dat$
600          DEALLOCATE Dat$
610          !
620          IF FNCheck_error(@Hp4291,"<CPFI: block read>")=-1 THEN SUBEXIT
630      END LOOP
640  SUBEND
650  !
660  ! Instrument Error Check
670  !
680  DEF FNCheck_error(@Hp4291,Str$)
690      DIM Err$[64]
700      OUTPUT @Hp4291;"SYST:ERR?"
710      ENTER @Hp4291;Err$
720      IF Err$"+0,""No error"" THEN
730          PRINT "ERROR: ";Str$;" ";Err$
740          RETURN -1
750      ELSE
760          RETURN 0
770      END IF
780  FNEND

```

Lines 140 to 200 accept the entry of the source file name and the destination file name.

Line 220 calls the subprogram to transfer a file from the 4291B to the external controller.

Lines 310 to 350 prepare for writing to the destination file.

Lines 370 to 400 prepare for reading the source file to the external controller.

Line 370 executes the query command to read data.

Lines 440 to 460 read the part indicating the length of the fixed length block data (see Figure 11-2) to obtain the length of the data to be transferred.

Lines 480 to 530 check the data length. If the data length is 0, the transfer process is terminated.

Depending on the data length obtained in lines 560 to 580, the program adjusts the format and reads the data part.

Line 590 writes the data to the destination file.

The maximum length of data transferred at a time is 16 Kbytes. Therefore, if the size of the source file is greater than 16 Kbytes, the transfer routine, lines 430 to 620, is repeated until transferring all of the data is completed.

Lines 680 to 780 provide a function to check that no error has occurred in the 4291B.

File Transfer from External Controller to 4291B

This program transfers a specified file in the current directory of the storage device connected to the external controller to the current directory of the selected storage device of the 4291B, giving a file name you desire.

This program, when executed, first prompts you to enter a source file name, as shown below. Enter the name of a file you want to transfer.

```
ENTER SOURCE FILE NAME ON CONTROLLER ?
```

Next, the program prompts you to enter the size of the source file as shown below (in this example, SAMPLE.STA has been entered as the source file name). Enter the size correctly in bytes.

```
ENTER SOURCE FILE NAME ON INSTRUMENT ?      SAMPLE.STA
ENTER SOURCE FILE SIZE ?
```

Then, the program prompts you to enter the destination file name, as shown below (in this example, the size of SAMPLE.STA is 12288 bytes). Enter the file name you want to give on the destination storage device. Note that a file with the same name will be overwritten, if it already exists.

```
ENTER SOURCE FILE NAME ON INSTRUMENT ?      SAMPLE.STA
ENTER SOURCE FILE SIZE ?                    12288
ENTER DESTINATION FILE NAME ON CONTROLLER ?
```

Disk



This program is included in the sample program disk. Its filename is TO_INST. This is the program for the external controller.

```
100 DIM Src_file$[50],Dst_file$[50]
110 ASSIGN @Hp4291 TO 717
120 OUTPUT @Hp4291;"*rst"
130 !
140 PRINT " ENTER SOURCE FILE NAME ON CONTROLLER ? ";
150 INPUT Src_file$
160 PRINT Src_file$
170 !
180 PRINT " ENTER SOURCE FILE SIZE ? ";
190 INPUT Src_size
200 PRINT Src_size
210 !
220 PRINT " ENTER DESTINATION FILE NAME ON INSTRUMENT ? ";
230 INPUT Dst_file$
240 PRINT Dst_file$
250 !
260 Copy_to_instr(@Hp4291,Src_file$,Src_size,Dst_file$)
270 !
280 END
290 !
300 ! copy_to_instrument
310 !
320 SUB Copy_to_instr(@Hp4291,Src_file$,Src_size,Dst_file$)
330 DIM Img$[32]
```

```

340     Max_bsize=16384
350     !
360     ASSIGN @Src_file TO Src_file$
370     !
380     CLEAR @Hp4291
390     OUTPUT @Hp4291;"*CLS"
400     OUTPUT @Hp4291;"WOPEN """;Dst_file$;"""
410     IF FNCheck_error(@Hp4291," <CPTI: wopen>")=-1 THEN SUBEXIT
420     Xfr_done=0
430     !
440     LOOP
450         SELECT (Src_size-Xfr_done)
460             CASE >Max_bsize
470                 Block_size=Max_bsize
480             CASE 0
490                 ASSIGN @Src_file TO *
500                 OUTPUT @Hp4291;"CLOSE"
510                 SUBEXIT
520             CASE ELSE
530                 Block_size=(Src_size-Xfr_done)
540         END SELECT
550         Xfr_done=Xfr_done+Block_size
560         !
570         ALLOCATE Dat$[Block_size]
580         !
590         Img$="#",&VAL$(Block_size)&"A"
600         ENTER @Src_file USING Img$;Dat$
610         !
620         Img$="8A,ZZZZZZ",&VAL$(Block_size)&"A"
630         OUTPUT @Hp4291 USING Img$;"WRITE #6",Block_size,Dat$,END
640         DEALLOCATE Dat$
650         IF FNCheck_error(@Hp4291," <CPTI: block write>")=-1 THEN SUBEXIT
660     END LOOP
670 SUBEND
680 !
690 ! Instrument Error Check
700 !
710 DEF FNCheck_error(@Hp4291,Str$)
720     DIM Err$[64]
730     OUTPUT @Hp4291;"SYST:ERR?"
740     ENTER @Hp4291;Err$
750     IF Err$"+0,""No error"" THEN
760         PRINT "ERROR: ";Str$;" ";Err$
770         RETURN -1
780     ELSE
790         RETURN 0
800     END IF
810 FNEND

```

Lines 140 to 240 accept the entry of the source file name and its size and the destination file name.

Line 260 calls the subprogram to transfer a file from the external controller to the 4291B.

Lines 400 to 410 prepare for writing the file to the destination storage device.

Lines 450 to 540 calculate the length of the data that has not been transferred based on the source file size previously entered and the length of the data that has been already transferred.

If the length of the remaining data does not exceed 16 Kbytes, it is set as the transfer data length; otherwise, 16 Kbytes is set as the transfer data length. Note that, if the length of the data not transferred is 0 at this time, the transfer process is terminated.

Lines 590 to 600 read data, whose amount is specified by the transfer data length, from the source file.

Lines 620 to 630 write data to the destination file in the fixed length block format (see Figure 11-2).

The maximum length of data transferred at a time is 16 Kbytes. Therefore, if the size of the source file is greater than 16 Kbytes, the transfer routine, lines 450 to 650, is repeated until transferring all of the data is completed.

Lines 710 to 810 provide a function to check that no error has occurred in the 4291B.

Note



To transfer a file from the external storage device to the 4291B, you must check the file size (number of bytes) in advance .

Displaying List of Files in Current Directory

This program displays the list of the files in the current directory.

Disk



This program is included in the sample program disk. Its filename is LIST_DIR. This program is for the external controller.

```

100 ASSIGN @@Hp4291 TO 717
110 OUTPUT @@Hp4291;"*rst"
120 !
130 Dir_instr(@@Hp4291)
140 !
150 END
160 !
170 ! Dir_instr
180 !
190 SUB Dir_instr(@@Hp4291)
200   DIM Stor_dev$[5],Curr_dir$[50],File_name$[13]
210   !
220   OUTPUT @Hp4291;"STODMEMO?"
230   ENTER @Hp4291;A
240   IF A=1 THEN
250     Stor_dev$="MEMO"
260   ELSE
270     Stor_dev$="DISK"
280   END IF
290   OUTPUT @Hp4291;"CWD?"
300   ENTER @Hp4291;Curr_dir$
310   PRINT "["&Stor_dev$&"]": "&Curr_dir$"
320   PRINT "Size[byte]      File Name"
330   PRINT "-----"
340   OUTPUT @Hp4291;"FNUM?"
350   ENTER @Hp4291;File_count

```

```

360     IF File_count>=1 THEN
370         FOR I=1 TO File_count
380             OUTPUT @Hp4291;"FNAME? ";I
390             ENTER @Hp4291;File_name$
400             OUTPUT @Hp4291;"FSIZE? ""&File_name$&""
410             ENTER @Hp4291;File_size
420             PRINT USING "XX,DDDDDD,XXXX,K";File_size,File_name$
430         NEXT I
440     END IF
450 SUBEND

```

Line 130 calls the subprogram to display the list of the files in the current directory.

Lines 220 to 310 check the storage device currently selected and its current directory name, and then display the result.

Lines 340 to 350 check the number of the files in the current directory.

If there are any files in the current directory, lines 360 to 440 check the name and size of every file and display them.

The following is the output result of the program, assuming that the selected storage device is the memory disk and the current directory, \TEST, contains 2 files, FILE1.STA (size: 24576 bytes) and FILE2.TIF (size: 16384 bytes) and 1 directory, DIR1. For size of a directory, -1 is displayed. To view the list of the files in DIR1, use the MMEM:CDIR command to change the current directory to DIR1 and then execute this program again.

```

[MEM0]: \TEST
Size[byte]    File Name
-----
      -1      ..\
      -1      DIR1\
    24576      FILE1.STA
    16384      FILE2.TIF

```


Facilitating Program Execution and Utilizing Storage Devices

You can easily execute Instrument BASIC programs previously saved on a storage device (floppy disk or memory disk). This means these two functions,

- Running a program through the softkey interface (Facilitating Program Execution)
- Automatically execute a certain program whenever the power is turned ON (AUTOST)

Topics covered include:

- How to save programs of Instrument BASIC
- Running a program through the softkey interface
- AUTOST
- Using storage devices

How to Save Programs of Instrument BASIC

To be able to use the function given on the front of this chapter, you have to save the program in 4291B's storage devices. 4291B has two storage devices; floppy disk drive and memory disk.

These devices have each specific benefit; the floppy disk has bigger memory and use easier, and the memory disk can read program faster.

The Procedure to Save Programs

This section shows the procedure to save programs in the memory disk.

1. Before all the other procedures, for example Measurement, you should set memory partition, if you need to change it. The key procedure is like this; **(System)** **MEMORY PARTITION**, Select memory partition, **DONE** **CHANGE** **YES**.

Note This operation let the prodno; reboot.



2. If you need to format the memory disk, press **(Save)** **STOR DEV [DISK]** keys and confirm the format and indicated as **STOR DEV [MEMORY]** and press **INITIALIZE INITIALIZE MEMORY: YES** keys.
3. To select the mass storage device to the memory disk, press **(System)** **IBASIC** **MORE** **MSI [INTERNAL]** keys.
4. Create a program.
5. Save the program using with keyboard; the command is SAVE "file-name" or SAVE "AUTOST".

6. To confirm the saved program saved, check the program name is showed or not using with this procedure: press **(System) PROGRAM MENU** keys. If you can't find the program name, press **STOR DEV [DISK]** key to switch the strage device from floppy disk drive to memory disk.
7. Press **(Save) BACK UP MEMO DISK** to keep the memory disk data as a backup.

Note

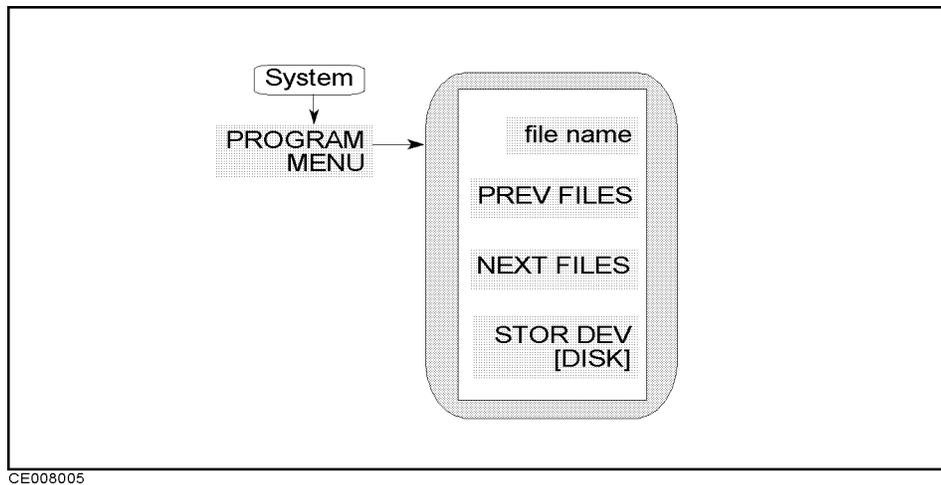


- When the 4291B is turned OFF, the data residing on the memory disk is lost. You can create a backup copy of the memory disk so that the memory disk is automatically restored from the backup copy next time the 4291B is turned on. To back up the memory disk, press **(Save) BACK UP MEMO DISK**. Alternatively, you can use the **STORMDISK** command over the GPIB.
- The memory disk can endure approximately 100,000 cycles of backup operation. This should be more than needed for the lifetime of the product, but you should avoid backing up the memory disk more frequently than actually needed.

Running a Program through the Softkey Interface

With the 4291B, you can run a program by just choosing the softkey associated with the program. To do so, follow these steps:

1. Open the Program Menu as illustrated below:



2. To tell the 4291B where program files are located, toggle the **STOR DEV** softkey as follows:

Storage Device	Toggle To
Floppy disk drive	STOR DEV [DISK]
Memory disk	STOR DEV [MEMORY]

- The 4291B displays a menu of softkeys that correspond to available program files residing on the selected storage device. Select your desired program by choosing the associated softkey. Then the 4291B executes the selected program.

Note

The 4291B may fail to recognize a program file that was created on an external controller such as a PC. If this is the case, use Instrument BASIC to load and re-save the program without adding any file name extension.

Automatically Starting a Program at Power-ON (AUTOST)

You can have the 4291B automatically execute a particular program whenever it is turned ON. To use this feature, save the program under the name of “AUTOST” on a floppy disk (note that the floppy disk containing the “AUTOST” program must be kept in the disk drive for the autostart feature to work).

To use the autostart feature, choose the following softkeys in order:

SYSTEM **PROGRAM MENU** **STOR DEV [DISK]**

Alternatively, you can save an “AUTOST” program file on the memory disk, and back up the memory disk as explained in “Memory Disk”. In this case, the “AUTOST” program will be automatically executed from the backup copy of the memory disk next time you turn ON the 4291B.

Note

If the autostart feature does not work, make sure that your selected storage device contains an “AUTOST” program.

About AUTOREC

Only saving the 4291B’s configuration as the file AUTOREC, 4291B can set as the information saved in AUTOREC file when 4291B is turned on. To use AUTOREC with AUTOST, you can simplify AUTOST program.

When the power is turned ON, the 4291B checks for any “AUTOREC” file. If an “AUTOREC” file exists, it reads the information contained in the file, and then loads and executes the “AUTOST” program.

To know more, see *Operation Manual* chapter eight.

Using Storage Devices

This section describes additional informations about strage devices.

BASIC Commands for Setting up the Storage Devices

MSI " :INTERNAL" and MSI " :INTERNAL,4,0" are commands to select the floppy disk drive.

MSI " :MEMORY" and MSI " :MEMORY,0,0" are commands to select the memory disk.

Floppy Disk Drive

- The INITIALIZE command of HP instrument BASIC can format a disk into the 1.44 MB (2HD) format only; it does not support the 720 KB (2DD) and 270 KB formats.
- The INITIALIZE command accepts only the default format option parameter (0), which provides 256 byte sectors.
- The 4291B can read and write a DOS format disk that meets the following specifications:
 - 1.44 MB, 80 tracks, double-sided, 18 sectors/track
- The CREATE and CREATE DIR commands are not available for a LIF format disk.

Note



The 4291B edition of HP instrument BASIC does not support an external storage device. The MASS STORAGE IS(MSI) command accepts either "INTERNAL,4" (floppy disk drive) or "MEMORY,0" (memory disk).

Memory Disk

The 4291B allows you to use as volatile memory disk. The memory disk must be formatted into either DOS or LIF. If you have backed up the memory disk, turning on the power causes the 4291B to restore the memory disk from the backup copy retaining the original format, so you don't need to reformat the memory disk.

Note



Backup is also important as a means of recovering your data in the event of power interruption or operator error. For example, even if you inadvertently formatted the 4291B memory disk, you could easily recover the data from the backup copy; all you have to do is turn OFF and ON the 4291B or issue the RESTMDISK command over GPIB.

Transferring Data between Floppy Disk and Memory Disk

The GPIB MMEM: COPY command allows you to copy files between the floppy disk drive and memory disk.

Note



MMEM: COPY does not support copying files between different data formats (LIF and DOS).

Introducing HP instrument BASIC System

This chapter introduces the analyzer's HP instrument BASIC (IBASIC) and describes how to connect and use a keyboard. Read this chapter before using HP instrument BASIC with the analyzer for the first time. The topics covered in this chapter are:

- Overview of HP instrument BASIC
- Controlling the analyzer
- Using HP instrument BASIC for the first time
- Entering BASIC Statements from the front panel keys
- Getting into/out of the EDIT mode
- Editing programs in the EDIT mode
- Listing programs
- Saving programs
- Listing file names
- Getting programs
- On Key Label function
- Increasing program speed
- Pass Control Between the External Controller
- External RUN/CONTINUE connector
- Graphics
- Softkeys used for HP instrument BASIC operation

Overview of HP instrument BASIC

HP instrument BASIC (IBASIC) can be used for a wide range of applications from simple recording and playback of measurement sequences to remote control of other instruments.

HP instrument BASIC is a complete system controller residing inside your analyzer. It communicates with your analyzer via GPIB commands and can also communicate with other instruments, computers, and peripherals over the GPIB interface.

The HP instrument BASIC's programming interface includes an editor and a set of programming utilities. The utilities allow you to perform disk I/O, renumber, secure, or delete all or part of your program.

The HP instrument BASIC command set is similar to the command set of HP 9000 Series 200/300 BASIC. Therefore, HP instrument BASIC programs can be run on any HP BASIC workstation with few if any changes. Porting information can be found in the *HP instrument BASIC Programming Techniques* of the *HP instrument BASIC Users Handbook*.

Controlling the Analyzer

HP instrument BASIC can control the analyzer (itself) through the “internal” GPIB bus. This means that an analyzer with HP instrument BASIC includes both a controller and an analyzer in the same instrument. They are connected through an internal GPIB bus.

Note



The select code of the internal GPIB interface is 8, and the GPIB address of the analyzer can be any number from 0 to 30. In this manual, we use “800” for the device selector of the analyzer.

For more information on GPIB addresses and device selectors, see “Device Selectors” in the *HP instrument BASIC Interfacing Techniques* of the *HP instrument BASIC Users Handbook* and “Available I/O Interfaces and Select Codes”.

Using HP instrument BASIC for the First Time

Allocating Screen Area for HP instrument BASIC

Because all of the analyzer’s screen is allocated for analyzer operation after power ON, you must allocate screen area for HP instrument BASIC when you want to use it. The analyzer provides four display allocation types. Select one of them using `DISPLAY ALLOCATION` under `Display`.

Let’s try

1. Press the following key and softkeys:

`Display` `DISPLAY ALLOCATION`

2. Press the following softkey.

`ALL BASIC`

The screen is cleared and all of the screen area is allocated for HP instrument BASIC.

3. Press the following softkey.

`ALL INSTRUMENT`

The total screen area is reallocated as the analyzer display.

4. Press the following softkey:

`HALF INSTR HALF BASIC`

The screen area is allocated so that the upper half of the screen is used for the analyzer operation and the lower half is used for HP instrument BASIC.

5. Press the following softkey:

`BASIC STATUS`

Three blank lines appear at the display line (lower area of the screen). This area is used by HP instrument BASIC to input commands and to display messages.

Entering BASIC Statements from the Front Panel Keys

The analyzer's HP instrument BASIC allows you to enter and execute statements from the front panel keys (if the external mini-DIN keyboard is not connected).

Press the following key and softkeys from the front panel:

(System) IBASIC MORE [1/3] MORE [2/3] COMMAND ENTRY

The Command Entry menu is displayed on the softkey menu area, and the active entry area displays the letters, the digits 0 through 9, and some special characters including mathematical symbols. Three sets of letters can be scrolled using the step keys, **(↑)** and **(↓)**. To enter a statement, press the step keys for the desired letter set, rotate the knob until the arrow "↑" points at the first letter, then press **SELECT LETTER**. Repeat this until the complete statement is entered, then press **DONE** to execute the statement.

Getting into/out of the EDIT Mode

Pressing the following key and softkey allows you to enter the EDIT mode immediately, irrespective of Display Allocation.

Getting into the EDIT Mode

Press the following key and softkeys from the front panel:

(System) IBASIC Edit

Entering the EDIT Mode from the Keyboard

Use the following keys to enter the EDIT mode with the cursor positioned at the specified line number. The *line_number* can be omitted. Press the following key among the 3 menus which leads to the **(Shift) - (F9)** key.

EDIT *line_number* **(Enter)**

or type as follows:

EDIT *line_number* **(Enter)**

To use the keyboard, the Keyboard Input Line must be allocated on the screen. If it is not, press **(Display)** **DISPLAY ALLOCATION** and select any allocation except ALL INSTRUMENT.

Getting Out of the EDIT Mode

The EDIT mode is exited by pressing **(Shift) - (Alt) - (F4)**, **(ESC)**, and **(Home)** from the keyboard (or by pressing the **END EDIT** softkey).

Editing Programs in the EDIT Mode

This section describes how to edit a program while in the EDIT mode, the topics are:

- Deleting characters
- Inserting characters
- Moving the cursor
- Scrolling lines and pages
- Jumping lines
- Inserting/deleting/recalling lines
- Clearing lines

See “The Keyboard” for more information on functions of each key.

Deleting Characters

There are two functions you can use to delete characters: “Back space” and “Delete characters.”

Back Space

Pressing **(Back space)** on the front panel (or on the keyboard) erases the character to the left of the cursor and moves the cursor left to the position of the erased character.

Deleting Characters

Pressing **(Delete char)** from the keyboard deletes the character at the cursor’s position.

Inserting Characters

The EDIT mode is always in the insert mode. Characters you type at the keyboard are inserted before the current cursor position. (Pressing **(insert)** performs no function.)

Moving the Cursor

The following key operations allow you to move the cursor horizontally along a line:

From the front panel	From the keyboard
Turning the knob	Pressing (←) and (→)

Scrolling Lines and Pages

Scrolling Lines

The following key operations enable you to scroll lines up and down:

From the front panel	From the keyboard
Pressing (↑) and (↓)	pressing (▲) and (▼)

Scrolling Pages

Pressing **(Page Up)** and **(Page Down)** from the keyboard causes the display to scroll up and down in one page increments.

Jumping from the Current Line

Jumping to a Specified Line

You can specify a line by using a line number or a label name when jumping from the current line as follows:

`GOTO LINE line_number`

or

`GOTO LINE label_name`

If the label specified is not defined in the program, an error will occur.

Jumping to the Top/Bottom of a Program

Pressing the following keys allows you to jump to top or bottom of the program:

-

-

Inserting/Deleting/Recalling Lines

- inserts a new line above the current cursor position.

- deletes the line at which the cursor is.

`RECALL LINE` recalls the last deleted line.

Clearing Line

Pressing - clears a line from the current cursor position to the end of the line.

Renumbering Program Line Numbers

The REN command allows you to renumber the program currently in memory. You should execute the REN command after exiting the EDIT mode. Press the following key among the 3 menus which leads to the - key.

`RENumber`

or

`REN`

You can specify the starting value, increment value, beginning line number, and the ending line number when renumbering a program as follows:

`RENumber starting_value, increment IN beginning_line_number, ending_line_number`

or type as follows:

`REN starting_value, increment IN beginning_line_number, ending_line_number`

line_label can be also use instead of *line_number*. For more information, see the *HP instrument BASIC Language Reference* of the *HP instrument BASIC Users Handbook*.

Listing Programs

The system can list the program on the screen and to a printer.

Listing on the Screen

You can list a program on the screen as follows:

1. Because the system lists a program in the print area, the Print Area must be allocated on the screen. For example:

```
(Display) DISPLAY ALLOCATE ALL BASIC
```

All of the screen area is allocated for the print area.

2. Type as follows:

```
LIST (Enter)
```

Listing to the Printer

Note For hard copy output, an parallel cable must connect the analyzer to the printer.



1. Set the output device to a printer as follows:

```
PRINTER IS PRT (Enter)
```

2. Type and press as follows:

```
LIST (Enter)
```

The program is listed on the printer.

3. Set the output device to LCD as follows:

```
PRINTER IS LCD (Enter)
```

Saving Programs (SAVE)

1. To use the built-in disk drive, insert a 2DD disk or 2HD disk into the disk drive.
2. If you are using a flexible disk for the first time, set the disk format to LIF or DOS and initialize the disk. See "To Save and Recall" in *Operation Manual* for the procedure.
3. If the display allocation is ALL INSTRUMENT, change the allocation to either HALF INSTRUMENT HALF BASIC or ALL BASIC. For example:

```
(Display) DISP ALLOCATION ALL BASIC
```

4. Select the storage units: the built-in flexible disk drive and the RAM disk memory.

For the built-in disk drive, enter

```
MSI ":INTERNAL" (Enter) or MSI ":INTERNAL,4,0" (Enter)
```

For the RAM disk memory, enter

```
MSI ":MEMORY,0" (Enter) or MSI ":MEMORY,0,0" (Enter)
```

5. Press the following key among the 3 menus which leads to the **(Shift) - (F9)** key. And type in the filename to which you will store the program as follows:

```
SAVE file_name (Enter)
```

You can also save the file from the keyboard. Type and press as follows:

```
SAVE file_name (Enter)
```

The program is stored on the disk.

Note



If you get the error -257, "File name error", a file on the disk already has the name you are trying to use. In this case, you have three choices:

- Pick a new file name that doesn't already exist. To determine which file names are already being used, use the "CAT" command (see below).
 - Replace an existing file, use the "RE-SAVE" statement.
 - Purge the old file using the PURGE command, then save the new one.
-

Listing File Names (CAT)

Listing to Screen

Press the following key and softkeys:

1. If the display allocation is ALL INSTRUMENT or BASIC STATUS, change the allocation to either HALF INSTRUMENT HALF BASIC or ALL BASIC. For example:

```
(Display) DISP ALLOCATION ALL BASIC
```

2. Press the following key among the 3 menus which leads to the **(Shift) - (F9)** key:

```
CAT (Enter)
```

You can list from the keyboard as follows:

```
CAT (Enter)
```

The file names stored on the disk are listed on the screen.

Note



Because the CAT statement outputs 80 columns to a line and the maximum number of columns to a screen is 61, each line is wrapped at the 62th column. If you do not want the list to wrap around, execute the following statement before executing the CAT command.

```
PRINTER IS CRT;WIDTH 80
```

Listing to Printer

Note



For hard copy output, a parallel cable must connect the analyzer to the printer.

1. Set the output device to be a printer as follows:

```
PRINTER IS PRT;WIDTH 80 
```

Type and press as follows:

```
CAT 
```

The program is listed on the printer.

2. Get the output device back to LCD:

```
PRINTER IS CRT 
```

Getting Programs (GET)

You can retrieve a program from the disk as follows:

1. If the display allocation is ALL INSTRUMENT, change the allocation to either HALF INSTRUMENT HALF BASIC or ALL BASIC. For example:

```
 DISP ALLOCATION ALL BASIC
```

2. Press the following key among the 3 menus which leads to the - key and type the filename you want to retrieve:

```
GET file_name 
```

You can get the file from the keyboard.

On Key Label Function

The HP instrument BASIC allows you to define softkeys from within a program. The softkey labels you define will appear when pressing or the - key on the Keyboard. The labels are displayed while running the program.

Example:

```
.....
100 ON KEY 1 GOTO 150
110 ON KEY 2 LABEL "Print" GOSUB Report
.....
```

You can also use the KEY statement to automatically display the label. This prevents you from pressing - or choosing while the user program is running. The KEY statement is used to display the softkey labels defined. The following set of statements is the same as the key strokes :

```
OUTPUT @Hp4291;"USKEY"
```

or

```

.....
200 OUTPUT @Hp4291;"KEY 47"    ! SYSTEM key
210 OUTPUT @Hp4291;"KEY 0"    ! IBASIC softkey
220 OUTPUT @Hp4291;"KEY 7"    ! ON KEY LABELS softkey
.....

```

For more information on the ON KEY statement, see the *HP instrument BASIC Language Reference* of the *HP instrument BASIC Users Handbook*.

Pass Control Between the External Controller

This section describes how to pass control between Instrument BASIC and the controller.

Pass Control

To pass active control to HP instrument BASIC:

```
PASS CONTROL 717 
```

Pass Control (On External Controller)

While the 4291B has control, it is free to address devices to talk and listen as needed. As the active controller, the 4291B can send messages to and read replies back from printers and plotters.

Note



The ability to assert the GPIB interface clear line (IFC) and remote enable line (REN) are reserved for the system controller. Even when HP instrument BASIC has active control, it is denied these functions.

```

ABORT 7      assert the interface clear line (IFC)
REMOTE 7     assert the remote enable line (REN)

```

To return active control to the system controller:

```
PASS CONTROL 721 
```

Return Control (On HP instrument BASIC)

Or, you can return control to the external controller by resetting the GPIB as follows:

```
ABORT 7 
```

Return Control (On External Controller)

To Execute an HP instrument BASIC Command from the External Controller

```
10 !
20 ! To Transfer the Program to iBASIC (on External Controller)
30 !
40   ABORT 7
50   ASSIGN @Hp4291 TO 717
60   INPUT "FILENAME?",File_name$
70   OUTPUT @Hp4291;"PROG:DEL:ALL"
80   OUTPUT @Hp4291;"PROG:DEF #0"
90   ASSIGN @File TO File_name$
100  ON ERROR GOTO Done
110   DIM Line$[1024]
120   LOOP
130     Line$=""
140     ENTER @File;Line$
150     OUTPUT @Hp4291;Line$
160   END LOOP
170 Done: !
180   OFF ERROR
190   OUTPUT @Hp4291;" " END
200 !
210 OUTPUT @Hp4291;"PROG:EXEC ""RUN""
220 END
```

Figure 10-1.

Sample Program : To Transfer the Program to IBASIC (on External Controller)

This Program transfers the program file in the mass storage of the external controller.

Lines 70 to 80 scratch any program that currently exists in the tester's HP instrument BASIC editor and open the editor.

Lines 90 to 160 transfer the program by line to the analyzer.

Line 190 closes the HP instrument BASIC Editor.

Line 210 executes the transferred program.

To Load an Array in an HP instrument BASIC Program to the External Controller

```

10 !
20 ! To Load iBASIC Program Array (on External Controller)
30 !
40 ABORT 7
50 ASSIGN @Hp4291 TO 717
60 DIM Passed(1:801,1:2)
70 OUTPUT @Hp4291;"PROG:NUMB? ""Dat"";"
80 ENTER @Hp4291;Passed(*)
90 END

```

Figure 10-2.

Sample Program : To Load HP instrument BASIC Program Array (on External Controller)

This program retrieves the array `Dat` generated in a program in HP instrument BASIC. This information is transferred to the external controller.

Lines 70 to 80 returns the program array `Dat(1:801,1:2)` generated in a Instrument BASIC program using `PROG:NUMB? "Dat"` query. The array is entered into `Passed(1:801,1:2)`.

Available I/O Interfaces and Select Codes

Available interfaces and their select codes in the analyzer's HP Instrument BASIC are listed in the following table:

Select Codes	Devices
1	LCD
2	Keyboard
7	External GPIB interface
8	Internal GPIB interface

External RUN/CONTINUE Connector

You can trigger `RUN` or `CONT` of the HP instrument BASIC program externally by applying a TTL signal through the `RUN/CONT` connector on the rear panel of the 4291B. The signal should be more than $20\mu\text{sec}$ in width and follow the negative logic. The program is triggered at the trailing edge of the pulse.

Graphics

HP instrument BASIC adds graphics capability to the analyzer. You can draw pictures on the LCD independent of the grids and traces.

The analyzer has two screens, the instrument screen and the graphics screen. These two screens are always displayed together on the LCD and are not separately selectable. The instrument screen consists of a trace display area and a softkey label area. The HP instrument BASIC editor is also displayed on the trace display area. The graphics screen covers the entire instrument screen as shown in Figure 10-3. The graphics screen is like an independent transparent overlay in front of the instrument screen. Therefore, you can draw figures in both the trace display and softkey label areas.

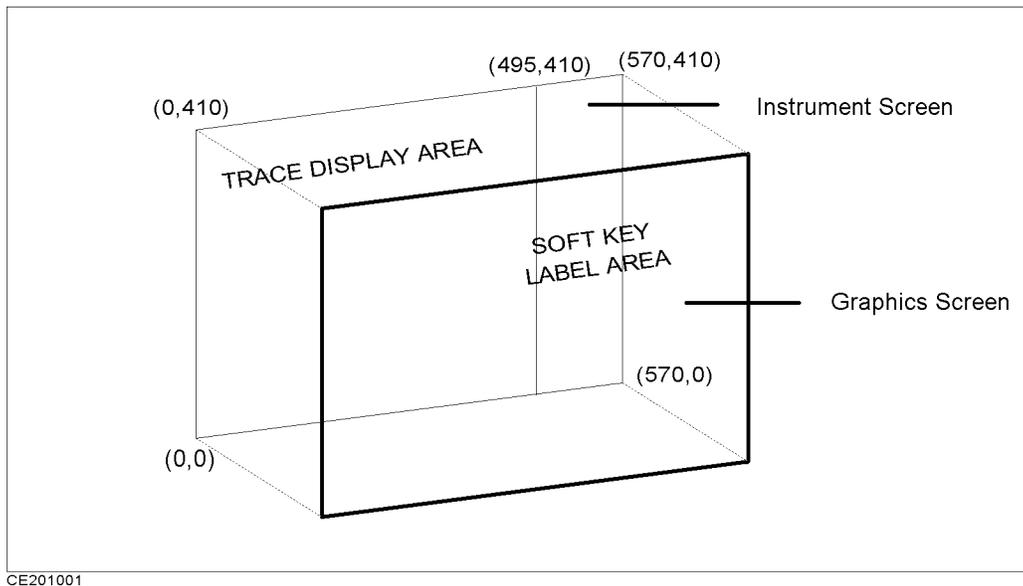


Figure 10-3. Screen Structure

Each point on the graphics screen is addressable using a coordinate address as shown in Figure 10-3. The bottom left corner is the origin (0,0) and the top right corner is the maximum horizontal and vertical end points (570,410). The MOVE and DRAW statement parameters are specified using these coordinate values. Because the aspect ratio of a graphics screen is 1, you need not adjust the aspect ratio when drawing figures.

HP instrument BASIC Graphics Commands

The analyzer's HP instrument BASIC has three graphics commands; MOVE, DRAW, and GCLEAR.

MOVE	Moves the pen from its current position to the specified coordinates.
DRAW	Draws a line from the current pen position to the specified coordinates.
GCLEAR	Clears the graphics screen, moves the pen from its current position to the origin (0,0), and selects pen 1.

Note

The total times of executing the MOVE and DRAW commands is up to 1933, even if the pen position is not changed.

Hard Copies

Graphics hard copies can be obtained with the printing function. Select **PRINT [STANDARD]** under **(Copy)**.

Initial settings

When power is turned ON, the default settings are as follows:

- MOVE 0,0

Example of Graphics Programming

This section describes an example of a simple program for drawing lines on the graphics screen.

Drawing a Straight Line

The following HP instrument BASIC program will draw a line from coordinate (50,200) to coordinate (300,200) on the display.

```
GCLEAR          ! INITIALIZE GRAPHICS MODE
MOVE 50,200     ! MOVE PEN TO COORDINATE (50,200)
DRAW 300,200    ! DRAW A LINE TO COORDINATE (300,200)
END
```

Drawing a Circle

Trying to express all graphical images using only straight lines is tedious, slow, and difficult. This example describes a subprogram you can use to draw a circle. It can draw a circle by passing the center coordinates and the radius as arguments to the following subroutine. This subroutine can be used as a base for drawing arcs, setting different values for Theta, etc.

```
SUB Drawcircle(Centx,Centy,R)  !
  DEG                          ! USE DEGREES FOR ANGLE EXPRESSIONS
  X=Centx+R                     !
  Y=Centy                       !
  MOVE X,Y                     ! MOVE PEN TO INITIAL POINT
  For Theta=1 to 360           !
    X=INT(COS(Theta)*R+Centx)  ! NEXT X COORDINATE ON CIRCLE
    Y=INT(SIN(Theta)*R+Centy)  ! NEXT Y COORDINATE ON CIRCLE
    DRAW X,Y                   ! UNTIL STARTING POINT IS REACHED
  NEXT Theta                   !
SUBEND                          !
```

The Keyboard

This section provides the key binding information of the keyboard.

Character Entry Keys

The character entry keys are arranged in the familiar QWERTY typewriter layout, but with additional features.

- Caps** Sets the unshifted keyboard to either upper-case (which is the default after power ON) or lower-case (normal typewriter operation).
- Shift** You can enter standard upper-case and lower-case letters, using the **Shift** key to access the alternate case.
- Enter** Has three functions:
- When a running program prompts you for data, respond by typing in the requested data and then press **Enter**. This signals the program that you have provided the data and that it can resume execution.
 - When typing in program source code, the **Enter** key is used to store each line of program code.
 - After typing in a command, the **Enter** key causes the command to be executed.
- CTRL** In the EDIT mode, **CTRL** allows you to control the editor in the same as using the cursor-control, display-control, and editing keys. For more detail, see “Using **CTRL** Key in Edit Mode”.
- Backspace** Erases the character to the left of the cursor and moves the cursor to the erased character’s position on the line.
- Tab** Performs no function.

Cursor-Control and Display-Control Keys

- ▲ ▼** Allow you to scroll lines up and down in the print display area. Shifted, these keys cause the display to scroll toward the top or bottom of the display.
- ▶ ◀** Allow you to move horizontally along a line. Shifted, these keys allow you to “jump” to the left and right limits of the current line.
- Page Up Page Down** Cause the display to scroll up or down in one page increments.
- Home** Performs no function.

Numeric Keypad

The numerical keypad provides a convenient way to enter numbers and perform arithmetic operations. Just type in the arithmetic expression you want to evaluate, then press **Enter**. The result is displayed in the lower-left corner of the screen.

- Enter** Performs the same function as the **Enter** key. The numerical keypad serves the same function as the numerical keypad on the front panel of the analyzer.
- Num Lock** Performs no function. Pressing the **Num Lock** key causes the LED ON/OFF, but the keys are performed as the numerical keypad only.

Editing Keys

Insert	Performs no function. The HP instrument BASIC is always in the insert mode. The characters you type are always inserted to the left of the cursor.
Shift - Insert	Inserts a new line above the cursor's current position (edit mode only).
Delete	Deletes the character at the cursor's position.
Shift - Delete	Deletes the line containing the cursor (edit mode only).
End	Delete the line containing the cursor except the line number.
Shift - End	Clears from the current cursor position to the end of the line.
Home	Clears the entire alpha screen. In EDIT mode, this exits the EDIT mode.

Program Control Keys

The following keys allow you to control execution of the program stored in the analyzer's memory.

Pause	Pause or Alt - F4 pauses program execution after the current line. Pressing Continue in the System menu resumes program execution from the point where it paused.
	Shift - Alt - F4 stops program execution after the current line. To restart the program, press Run in the System menu.
	When in the editor mode, Shift - Alt - F4 exits the edit mode.
Ctrl - Break	Ctrl - Break resets program execution immediately without erasing the program from memory (BASIC RESET).
	Pauses program execution when the computer is performing or trying to perform an I/O operation. Press Alt - F5 instead of Pause or Alt - F4 when the computer is hung up during an I/O operation, because Pause or Alt - F4 works only after the computer finishes the current program line.

System Control Keys

Shift - Page Up (Recall)	Shift - Page Up (Recall) recalls the last line the you entered, executed, or deleted. Several previous lines can be recalled this way. Recall is particularly handy to use when you mistype a line. Instead of retyping the entire line, you can recall it, edit it using the editing keys, and enter or execute it again.
	Shift - Page Down moves forward through the recall stack.
Alt - F3 (Run)	Starts a program running from the beginning.
Alt - F2 (Continue)	Resumes program execution from the point where it paused.
F12 (IBASIC)	Allows you to type BASIC commands on Keyboard Input Line. If Display Allocation is ALL INSTRUMENT, pressing this key changes the Display Allocation to BASIC STATUS.
	Shift - F12 changes Display Allocation to ALL INSTRUMENT.

Softkeys and Softkey Control

There are eight softkeys (labeled **F1** through **F8**). The softkey labels are indicated on the right of the analyzer's screen.

Softkey Control Keys

Pressing the following:

- F9** Leads to the IBASIC menu, which controls programs and the editor.
- Shift** - **F9** leads to the BASIC menu from which to control a BASIC program. This menu is the same menu displayed when pressing **SYSTEM** **IBASIC** from the front panel.
- In the edit mode, pressing **F10** leads to the Edit System menu, which provides softkeys to conveniently enter BASIC commands.
- Shift** - **F10** (User) leads to the ON KEY LABEL menu, which are user defined softkeys in a BASIC program. (For information on getting to this menu through the HP instrument BASIC, see "On Key Label Function".)

Softkeys

F9 and **F10** keys leads to the IBASIC menu. Pressing a softkey performs the command labeled or produces a sequence of characters on the keyboard input line (or on the "current line" in the EDIT mode).

Pressing the softkeys on the front panel of the analyzer performs the same functions as pressing the **F1** through **F8** function keys.

Softkeys Accessed form **F10** Key

F10 key allows you to access three different softkey flows dependent on conditions as follows:

- Pressing **F10** accesses the Program Control menu
- In editor mode, pressing **F10** accesses the Edit System menu
- Pressing **Shift**-**F10** accesses the On Key Label menu.

The menus listed above are described in "Instrument BASIC Menu" in the *Operation Manual*.

Using **CTRL** Key in Edit Mode

In the edit mode, pressing **CTRL**, holding it down and pressing another key, allows you to control the editor in the same way as pressing control keys such as **▲**, **▼**, **Insert line**, etc.

If you press ...	It performs ...
CTRL -a	Moves the cursor to beginning of line, (the same function as Shift - ←).
CTRL -b	Moves cursor backward one character, (the same function as ←).
CTRL -d	Deletes a character, (the same function as Delete).
CTRL -e	Moves the cursor to end of the line, (the same function as Shift - →).
CTRL -f	Moves cursor forward character along a line, (the same function as →).
CTRL -g	Allows you to move the cursor to any line number or label, after press CTRL - g , type a line number or label name and press Enter , the cursor moves to the specified line, (the same function as GOTO LINE).
CTRL -h	Deletes backward one character, (the same function as Back Space).
CTRL -j	Performs the same function as Enter .
CTRL -k	Deletes a line from the cursor's current position to the end of the line.
CTRL -m	Performs the same function as Enter .
CTRL -n	Moves the cursor to the next line, (the same function as ▼).
CTRL -o	Inserts a new line above the cursor's current position, (the same function as Shift - Insert).
CTRL -p	Moves the cursor to the previous line, (the same function as ▲).

Run Light Indications

- (blank) Program stopped; can execute commands; CONTINUE not allowed.
- Program paused; can execute commands; CONTINUE is allowed.
- ? BASIC program waiting for input from keyboard; cannot execute commands.
- * This indication has two possible meanings:
 - Program running; CANNOT execute commands. CONTINUE not allowed.
 - System executing command entered from keyboard; CANNOT enter commands.


```

Ioport=READIO(15,0)      ! Substitutue data from the I/O port to Ioport

100 EXECUTE "OUTPRES0?"  ! Execute OUTPRES0? query
110 Za=READIO(8,0)      ! Substitutue the first return value from the register 0 to Za
120 Fa=READIO(8,1)      ! Substitutue the second return value from the register 1 to Fa
130 Zr=READIO(8,2)      ! Substitutue the third return value from the register 2 to Zr
140 Fr=READIO(8,3)      ! Substitutue the fourth return value from the register 3 to Fr

```

SET TIME

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This command resets the time-of-day given by the real-time clock.

■ Example Commands

```

SET TIME TIME("22:00:30")      ! Set the real time clock to 22:00:30
SET TIME Hours*3600+Minutes*60 ! Set the real time clock to Hours:Minutes

```

SET TIMEDATE

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This command resets the absolute seconds (time and day) given by the real-time clock.

■ Example Commands

```

SET TIMEDATE DATE("4 JAN 1993")+TIME("10:00:00") ! Set the real time clock to
                                                    ! 10:00 Jan/4/1993
SET TIMEDATE TIMEDATE+86400                       ! Set the real time clock 1 day ahead

```

TIME

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This command converts data formatted as time of day (HH:MM:SS), into the number of seconds past midnight.

■ Example Commands

```

Seconds=TIME("8:37:20")      ! Seconds passed since 0:00 until 8:37:20
SET TIME TIME("8:37:20")      ! Set the real time cloc to 8:37:20
ON TIME TIME("12:10") GOSUB Lunch ! Go to "Lunch" at 12:10

```

TIME\$

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This command converts the number of seconds past midnight into a string representing the time of day (HH:MM:SS).

■ Example Commands

```

DISP "The time is: ";TIME$(TIMEDATE) ! Shows the current time based on the real time clock
PRINT TIME$(45296)                  ! Shows the time that passed 45296 seconds since 0:00

```

WRITEIO

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This statement writes register data in decimal notation to a specified EXECUTE command parameter register or to a specified I/O port.

Item	Description	Range
select code	numeric expression	8: EXECUTE register 15: I/O port
register number	numeric expression	0 to 800 (Select code 8) 0: I/O port
register data	numeric expression	-2147483648 to +2147483647 0 to 255: I/O port

■ Example Commands

```

WRITEIO 15,0;12                ! Writes 12 on the I/O port

100 WRITEIO 8,0; 100E6         ! Writes the first argument on the register 0
110 WRITEIO 8,1; 200E6         ! Writes the second argument on the register 1

```

Command Reference

This chapter provides a reference for the GPIB commands of the analyzer. Use this information as a reference to the syntax requirements and general function of the individual commands.

The chapter is organized as follows:

- Instrument control commands
- Common commands
- Simple commands

The instrument control commands, common commands, and simple commands are listed in alphabetical order. See Appendix C for a functional list of the commands.

See the *Operating Manual* for the details of each function.

Conventions and Definitions

The following conventions and definitions are used to describe the commands.

① → CALCulate Subsystem

The CALCulate subsystem controls postacquisition data processing as follows:

1. Controls the marker and equivalent circuit evaluation function (CALCulate:EVALuate).
2. Selects the measurement parameter (CALCulate:FORMat, CALCulate:FORMat:UNIT:ANGLE, CALCulate:MATH1).
3. Controls the limit test function (CALCulate:LIMit).
4. Controls trace math function (CALCulate:MATH2).

② → CALCulate:EVALuate Subsystem

This subsystem is used to set the marker and equivalent circuit evaluation function.

Related Command DISPlay[:WINDow]:TRACe{1-17}:MARKer{1-8}

③ → :BAND:FULL[:STATe] {OFF|ON|0|1}

④ → CALCulate:EVALuate:BAND:FULL[:STATe] {OFF|ON|0|1}

⑤ → Sets the partial search of the marker search function ON or OFF. (PART SRCH ON off under Search)

Parameter	Description
OFF or 0	Partial search ON.
ON or 1	Partial search OFF.

Query Response is {0|1}. (<NR1> format)

①	<p>Section header</p> <p>The GPIB commands are divided into the following sections.</p> <p>Sections which are top of the command tree (The sections are provided in alphabetical order.)</p> <p>Common commands section</p> <p>Simple commands section</p>
②	<p>Lower level subsystem and its description</p> <p>Describes a lower level subsystem. This subsystem contains functions that are grouped together.</p>
③	<p>Command nodes and required parameter</p> <p>Upper case bold characters represent the command that must appear exactly as shown with no embedded spaces. Lower case bold characters can be omitted.</p> <p>Words or characters enclosed in < > brackets are used to symbolize a parameter type.</p> <p>Words enclosed in the [] brackets can be omitted.</p> <p>Several codes are enclosed in the { } brackets and one of these codes can be selected. For example, {OFF ON 0 1} means OFF, ON, 0, or 1. {1-4} means 1, 2, 3, or 4.</p> <p>For example, this command usage is:¹</p> <pre>OUTPUT @Hp4291;"CALC:EVAL:BAND:FULL ON"</pre>
④	<p>Command spelled out from the root</p>
⑤	<p>Command description and parameter description</p>

¹ In this manual, example statements use @Hp4291 as the GPIB address. Therefore, @Hp4291 must be assigned as the analyzer GPIB address in a preceding program statement.

ABORt

ABORt

Reset the trigger system and place all trigger sequences in the idle state. (**SWEEP: HOLD** under **Trigger**);No query)

For more information about the trigger, see *Programming Manual*.

Related Command **INITiate:CONTInuous {OFF|ON|0|1}**
INITiate[:IMMediate]

Example To abort the measurement and hold the sweep immediately,

```
OUTPUT @Hp4291"INIT:CONT OFF"
OUTPUT @Hp4291"ABORt"
```

BLIGHT {**OFF|ON|0|1**}

(*Simple Command*) See “Simple Commands” later in this chapter.

CALCulate Subsystem

The CALCulate subsystem controls post acquisition data processing as follows:

1. Controls the marker and equivalent circuit evaluation function (CALCulate:EVALuate).
2. Selects the measurement parameter (CALCulate:FORMat, CALCulate:FORMat:UNIT:ANGLE, CALCulate:MATH1).
3. Controls the limit test function (CALCulate:LIMit).
4. Controls data math function (CALCulate:MATH2).

CALCulate:EVALuate Subsystem

This subsystem is used to set the marker and equivalent circuit evaluation function.

Related Command DISPlay[:WINDow]:TRACe{1-17}:MARKer{1-8}

:BAND:FULL[:STATe] {OFF|ON|0|1}

CALCulate:EVALuate:BAND:FULL[:STATe] {OFF|ON|0|1}

Sets the partial search of the marker search function ON or OFF. (PART SRCH ON off under (Search))

Parameter	Description
OFF or 0	Partial search ON.
ON or 1	Partial search OFF.

Query Response is {0|1}.

Example OUTPUT @Hp4291;"CALC:EVAL:BAND:FULL OFF"
 OUTPUT @Hp4291;"CALC:EVAL:BAND:FULL?"
 ENTER @Hp4291;A

:BAND:SPAN DMARker

CALCulate:EVALuate:BAND:SPAN DMARker

Sets the partial search range to the range between the marker and the Δmarker. (MKRΔ→SEARCH RNG under (Search); No query)

Example OUTPUT @Hp4291;"CALC:EVAL:BAND:SPAN DMAR"

:BAND:START MARKer

CALCulate:EVALuate:BAND:START MARKer

Sets the left (lower) border of the partial search range at the current position of the marker. (MKR→LEFT RNG under (Search); No query)

:BAND:STOP MARKer

CALCulate:EVALuate: BAND:STOP MARKer

Sets the right (higher) border of the partial search range at the current position of the marker. (MKR→RIGHT RNG under **Search**); No query)

:COUple {OFF|ON|0|1}

CALCulate:EVALuate: COUple {OFF|ON|0|1}

Sets the coupled or uncoupled marker mode. (MKR [UNCOUPLE] , MKR [COUPLE] under **Marker**)

Parameter	Description
OFF or 0	Uncoupled marker mode.
ON or 1	Coupled marker mode.

Query Response is {0|1}.

Example OUTPUT @Hp4291;"CALC:EVAL:COUP ON"
 OUTPUT @Hp4291;"CALC:EVAL:COUP?"
 ENTER @Hp4291;A

:EFFect:ON {1|2}

CALCulate:EVALuate: EFFect:ON {1|2}

Selects the destination channel of the marker→ functions. When a marker→ function is performed, the sweep parameter or amplitude value of the destination channel is changed. (CROSS CHAN ON off under **Marker→**)

Parameter	Description
OFF or 0	Current active channel as the destination channel
ON or 1	Current inactive channel as the destination channel ¹

¹ Can be selected only when the dual channel function is ON.

■ Query Response

{0|1} <new line><END>

■ Equivalent SCPI Command

:CALCulate:EVALuate:EFFect:ON{1|2}

When channel 1 is active,

Parameter	Description
1	The channel currently active is selected.
2	The channel currently not active is selected.

When channel 2 is active,

CALCulate:EVALuate

Parameter	Description
1	The channel currently not active is selected.
2	The channel currently active is selected.

:EPARameters

CALCulate:EVALuate:EPARameters

Calculates and displays the equivalent circuit parameters. (CALCULATE EQV PARAMS under Display); No query)

:EPARameters:CIRCuit {A|B|C|D|E}

CALCulate:EVALuate:EPARameters:CIRCuit {A|B|C|D|E}

Selects the equivalent circuit. (SELECT EQV CKT [] under Display)

:EPARameters:SIMulation

CALCulate:EVALuate:EPARameters:SIMulation

Simulates frequency response of the equivalent circuit. (SIMULATE F-CHRST under Display); No query)

:INTerpolate {OFF|ON|0|1}

CALCulate:EVALuate:INTerpolate {OFF|ON|0|1}

Sets the continuous or discontinuous marker mode. (MKR [CONT], MKR [DISCRETE] under Marker)

Parameter	Description
OFF or 0	Discontinuous marker mode.
ON or 1	Continuous marker mode.

Query Response is {0|1}.

```
Example          OUTPUT @Hp4291;"CALC:EVAL:INT ON"
                  OUTPUT @Hp4291;"CALC:EVAL:INT?"
                  ENTER @Hp4291;A
```

:MSTatistics {OFF|ON|0|1}

CALCulate:EVALuate:MSTatistics {OFF|ON|0|1}

See "CALCulate:EVALuate:MSTatistics[:STATE] {OFF|ON|0|1}".

:MStatistics:DATA?

CALCulate:EVALuate:MStatistics:DATA?

Queries the marker statistics that are calculated by
 CALCulate:EVALuate:MStatistics[:STATE] ON. (Query only)

Query Response is $\langle numeric(mean) \rangle, \langle numeric(s.dev) \rangle, \langle numeric(p-p) \rangle$.

Where,

$\langle numeric(mean) \rangle$: mean
 $\langle numeric(s.dev) \rangle$: standard deviation
 $\langle numeric(p-p) \rangle$: peak-to-peak value

:MStatistics[:STATE] {OFF|ON|0|1}

CALCulate:EVALuate:MStatistics[:STATE] {OFF|ON|0|1}

Calculates the marker statistics (the mean, standard deviation, and peak-to-peak values) in the portion of the displayed trace that is in the search range.

(STATICS ON off under [Utility](#))

Parameter	Description
OFF or 0	Does not display the marker statistics.
ON or 1	Displays the marker statistics.

Query Response is {0|1}.

Related Command CALCulate:EVALuate:MStatistics:DATA?

Example OUTPUT @Hp4291;"CALC:EVAL:MST ON"
 OUTPUT @Hp4291;"CALC:EVAL:MST?"
 ENTER @Hp4291;A

:ON[1] "TR{1-21}"

CALCulate:EVALuate:ON1 "TR{1-21}"

Displays the marker and selects the trace on which the marker functions are used.

(MKR ON [] under [Marker](#))

Parameter	Description
"TR1"	data trace
"TR{2-17}"	memory traces
"TR{18-21}"	user traces

Query Response is "TR{1-21}".

Example OUTPUT @Hp4291;"CALC:EVAL:ON ""TR1""
 OUTPUT @Hp4291;"CALC:EVAL:ON?"
 ENTER @Hp4291;A\$

:ON2 {**"OFF"**|**"ACV"**|**"ACC"**|**"DCV"**|**"DCC"**}

CALCulate:EVALuate:ON2 {**"OFF"**|**"ACV"**|**"ACC"**|**"DCV"**|**"DCC"**}

Displays the voltage or current level applied to the DUT at the marker point.

(**LEVEL MON** [] under **Utility**)

Parameter	Description
OFF	Displays the measurement data (OFF ; level monitor OFF).
ACV	Displays the AC voltage level (ACV).
ACC	Displays the AC current level (ACC).
DCV	Displays the DC voltage level (DCV ; option 001 only).
DCC	Displays the DC current level (DCC ; option 001 only).

Query response is {"OFF"|"ACV"|"ACC"|"DCV"|"DCC"}.

Related Command DATA[:DATA]? MON

:PEAK:EXCursion {<numeric>|**DMARker**}

CALCulate:EVALuate:PEAK:EXCursion {<numeric>|**DMARker**}

See "CALCulate:EVALuate:PEAK:EXCursion[:Y] {<numeric>|**DMARker**"}.

:PEAK:EXCursion:X {<numeric>|**DMARker**}

CALCulate:EVALuate:PEAK:EXCursion:X {<numeric>|**DMARker**}

Sets the peak ΔX value that is used to define the peak. (**PEAK DEF: ΔX** under **Search**)

Parameter	Range or Description	Unit
<numeric>	0 to 8×10^9	Hz (Frequency sweep)
	0 to 8	V (OSC level sweep)
	0 to 80	V (DC-V sweep)
	0 to 0.8	A (DC-I sweep)
DMARker	The smaller value of the difference of amplitude values between the present marker position and both side display points of the marker. (MKR→PEAK DELTA: ΔX under Search); No query)	

Query response is <numeric>.

Example OUTPUT @Hp4291;"CALC:EVAL:PEAK:EXC:X DMAR"

:PEAK:EXCursion[:Y] {<numeric>|DMARker}

CALCulate:EVALuate:PEAK:EXCursion[:Y] {<numeric>|DMARker}

Sets the peak ΔY value that is used to define the peak. (PEAK DEF: ΔY under [Search](#))

Parameter	Range or Description	Unit
<numeric>	0 to 100×10^6	y-axis unit
DMARker	The smaller value of the difference of amplitude values between the present marker position and both side display points of the marker. (MKR→PEAK DELTA: ΔY under Search); No query)	

Query response is <numeric>.

Example OUTPUT @Hp4291;"CALC:EVAL:PEAK:EXC DMAR"

:PEAK:POLarity {POSitive|NEGative}

CALCulate:EVALuate:PEAK:POLarity {POSitive|NEGative}

Sets the peak polarity for the marker search functions. (PEAK PLRTY POS neg under [Search](#))

Parameter	Description
POSitive	Positive peak
NEGative	Negative peak

Query response is {POS|NEG}.

Example OUTPUT @Hp4291;"CALC:EVAL:PEAK:POL POS"

OUTPUT @Hp4291;"CALC:EVAL:PEAK:POL?"

ENTER @Hp4291;A\$

:PEAK:THReshold {<numeric>|MARKer}

CALCulate:EVALuate:PEAK:THReshold {<numeric>|MARKer}

Sets the threshold values. (THRESHOLD VALUE under [Search](#))

Parameter	Range or Description	Unit
<numeric>	-100×10^6 to 100×10^6	y-axis unit
MARKer	The amplitude value of the present marker position. (MKR→THRESHOLD under Search)	

Query response is <numeric>.

Example OUTPUT @Hp4291;"CALC:EVAL:PEAK:THR 0"

OUTPUT @Hp4291;"CALC:EVAL:PEAK:THR?"

ENTER @Hp4291;A

:PEAK:THReshold:STATe {OFF|ON|0|1}

CALCulate:EVALuate:PEAK:THReshold:STATe {OFF|ON|0|1}

Sets the threshold ON or OFF. (THRESHOLD ON off under [Search](#))

Parameter	Description
OFF or 0	Threshold OFF.
ON or 1	Threshold ON.

Query response is {0|1}.

Example

```
OUTPUT @Hp4291;"CALC:EVAL:PEAK:THR:STAT ON"
OUTPUT @Hp4291;"CALC:EVAL:PEAK:THR:STAT?"
ENTER @Hp4291;A
```

:R:FORMat {RIMaginary|MLIPhase|MLOPhase|RX|GB|SWRPhase}

CALCulate:EVALuate:R:FORMat {RIMaginary|MLIPhase|MLOPhase|RX|GB|SWRPhase}

Selects a format to read out the value of a Smith, polar, or admittance chart using markers. (REAL IMAG, LIN MAG PHASE, LOG MAG PHASE, R+jX, G+jB, SWR PHASE under [Utility](#))

Parameter	Description
RIMaginary	Real and imaginary form.
MLIPhase	Linear magnitude and phase form.
MLOPhase	Log magnitude and phase form.
RX	Complex impedance form (R+jX).
GB	Complex admittance form (G+jB).
SWRPhase	SWR and phase form.

Query Response is {RIM|MLIP|MLOP|RX|GB|SWRP}.

Example

```
OUTPUT @Hp4291;"CALC:EVAL:R:FORM GB"
OUTPUT @Hp4291;"CALC:EVAL:R:FORM?"
ENTER @Hp4291;A$
```

:REFerence:DATA?

CALCulate:EVALuate:REFerence:DATA?

Returns the Δ marker value. (Query only)

Query response is <numeric(val1)>,<numeric(val2)>,<numeric(stimulus)>.

Where,

<numeric(val1)> : Amplitude value
 <numeric(cent)> : Auxilliary amplitude value
 <numeric(stimulus)> : Stimulus value

Example

```
OUTPUT @Hp4291;"CALC:EVAL:REF:DATA?"
ENTER @Hp4291;A,B,C
```

:REference:X <numeric>

CALCulate:EVALuate:REference:X <numeric>

Sets the stimulus value of the Δ marker. (**ΔMKR STIMULUS** under **(Marker)**)

Parameter	Range	Unit
<numeric>	Stimulus start value to stop value.	Hz (frequency) V (OSC level or DC-V) A (DC-I)

Query Response is <numeric>.

Example OUTPUT @Hp4291;"CALC:EVAL:REF:X 200MAHZ"
 OUTPUT @Hp4291;"CALC:EVAL:REF:X?"
 ENTER @Hp4291;A

:REference:Y[1] <numeric>

CALCulate:EVALuate:REference:Y[1] <numeric>

Sets the amplitude value of the fixed Δ marker. (**FIXED ΔMKR VALUE** under **(Marker)**)

Parameter	Range	Unit
<numeric>	-1×10^9 to 1×10^9	Amplitude unit

Query Response is <numeric>.

Related Command

```
DISPlay[:WINDow]:TRACe{1-21}:MARKer[1]:RELative
DISPlay[:WINDow]:TRACe{1-21}:MARKer[1]:RELative:REference
```

Example OUTPUT @Hp4291;"CALC:EVAL:REF:Y1 0"
 OUTPUT @Hp4291;"CALC:EVAL:REF:Y1?"
 ENTER @Hp4291;A

:REference:Y2 <numeric>

CALCulate:EVALuate:REference:Y2 <numeric>

Sets the auxiliary amplitude value of the fixed Δ marker. This command is used with a polar, Smith, or admittance chart. (**FIXED ΔMKR AUX VALUE** under **(Marker)**)

Parameter	Range	Unit
<numeric>	-1×10^9 to 1×10^9	Auxiliary amplitude unit

Query Response is <numeric>.

Example OUTPUT @Hp4291;"CALC:EVAL:REF:Y2 0"
 OUTPUT @Hp4291;"CALC:EVAL:REF:Y2?"
 ENTER @Hp4291;A

:WIDTh:DATA?

CALCulate:EVALuate:WIDTh:DATA?

Queries the bandwidth parameters. (Query only)

Query response is

<numeric(width)>,<numeric(cent)>,<numeric(Q)>,<numeric(peak)>,<numeric(ΔL)>,<numeric(ΔR)>.

Where,

<numeric(width)> : bandwidth value
 <numeric(cent)> : bandwidth center value
 <numeric(Q)> : Q value
 <numeric(peak)> : bandwidth peak value
 <numeric(ΔL)> : ΔL value
 <numeric(ΔR)> : ΔR value

Example

```
OUTPUT @Hp4291;"CALC:EVAL:WIDT:DATA?"
ENTER @Hp4291;A,B,C,D,E,F
```

:WIDTh:STATe {OFF|ON|0|1}

CALCulate:EVALuate:WIDTh:STATe {OFF|ON|0|1}

Sets the bandwidth function ON or OFF. (WIDTH [ON], WIDTH [OFF] under)

Parameter	Description
OFF or 0	Marker width OFF.
ON or 1	Marker width ON.

Query Response is {0|1}.

Example

```
OUTPUT @Hp4291;"CALC:EVAL:WIDT:STAT ON"
OUTPUT @Hp4291;"CALC:EVAL:WIDT:STAT?"
ENTER @Hp4291;A
```

:WIDTh:XPOStion:IN

CALCulate:EVALuate:WIDTh:XPOStion:IN

Searches for another bandwidth cutoff point that is within the current cutoff point.

(SEARCH IN under ;No query)**:WIDTh:XPOStion:OUT**

CALCulate:EVALuate:WIDTh:XPOStion:OUT

Searches for another bandwidth cutoff point that is outside of the current cutoff point.

(SEARCH OUT under ;No query)

:WIDTh:Y {DIVS2|MULS2|DIV2|FIXed[,<numeric>]}

CALCulate:EVALuate:WIDTh:Y {DIVS2|MULS2|DIV2|FIXed[,<numeric>]}

Selects the bandwidth cutoff point value as shown in the parameter table.

(WIDTH VALUE under [Search](#));No query)

Parameter	Description
DIVS2	The value equal to (marker value) $\times 1/\sqrt{2}$. (MKRVAL/(\sqrt{2}))
MULS2	The value equal to (marker value) $\times \sqrt{2}$. (MKRVAL*(\sqrt{2}))
DIV2	The value equal to (marker value) $\times 1/2$. (MKRVAL/2)
FIXed,[<numeric>]	The value equal to (marker value) \times <numeric>. If the <numeric> is omitted, the current setting value is used. (FIXED VALUE)

Query Response is {DIVS2|MULS2|DIV2|FIX,<numeric>}¹

¹ FIX,<numeric> is returned in character format. For example, if you set CALC:EVAL:WIDT:Y FIX,-10, the query response is FIX,-1.000000e+01.

Example

```
OUTPUT @Hp4291;"CALC:EVAL:WIDT:Y DIVS2"
ENTER @Hp4291;A$
```

```
OUTPUT @Hp4291;"CALC:EVAL:WIDT:Y?"
ENTER @Hp4291;A$
```

:Y{1-8}:DATA? [CH1|CH2]

CALCulate:EVALuate:Y{1-8}:DATA? [CH1|CH2]

Queries the measurement values and stimulus value at the marker position. (Query only)

Node	Description
:Y1	Marker.
:Y{2-8}	Sub-marker 1 to 7.

Parameter	Description
CH1	Queries the marker value of the channel 1.
CH2	Queries the marker value of the channel 2.
None	Queries the marker value of the active channel.

Query response is <numeric(val1)> <numeric(val2)> <numeric(stimulus)>.

Where,

<numeric(val1)> : Amplitude value

<numeric(val2)> : Auxiliary amplitude value

<numeric(stimulus)> : Stimulus value

Example

```
OUTPUT @Hp4291;"CALC:EVAL:Y:DATA? CH1"
ENTER @Hp4291;A,B,C
```


:Y[1]:XPOSITION:LTARget

CALCulate:EVALuate:Y[1]:XPOSITION:LTARget

Moves the marker to the next occurrence of the target value to the left of the present marker position. (SEARCH LEFT under Search); No query)

:Y[1]:XPOSITION:MAXimum

CALCulate:EVALuate:Y:XPOSITION:MAXimum

Moves the marker to the maximum point. (SEARCH: MAX under Search); No query)

:Y[1]:XPOSITION:MINimum

CALCulate:EVALuate:Y:XPOSITION:MINimum

Moves the marker to the minimum point. (SEARCH: MIN under Search); No query)

:Y[1]:XPOSITION:NPEak

CALCulate:EVALuate:Y:XPOSITION:NPEak

Moves the marker to the next peak. (NEXT PEAK under Search); No query)

:Y[1]:XPOSITION:PEAK

CALCulate:EVALuate:Y[1]:XPOSITION:PEAK

Moves the marker to peak point. (SEARCH: PEAK under Search); No query)

:Y{1-8}:XPOSITION:POINT <numeric>

CALCulate:EVALuate:Y{1-8}:XPOSITION:POINT <numeric>

Moves the marker to the specified data point number.

Parameter	Description
<numeric>	1 to number of points.

Query response is <numeric>.

Example

```
OUTPUT @Hp4291;"CALC:EVAL:Y:XPOS:POIN 1"
```

```
OUTPUT @Hp4291;"CALC:EVAL:Y:XPOS:POIN?"
```

```
ENTER @Hp4291;A
```

:Y[1]:XPOSITION:RPEak

CALCulate:EVALuate:Y[1]:XPOSITION:RPEak

Moves the marker to the peak to the right of the present marker position.

(NEXT PEAK RIGHT under **Search**); No query)

:Y[1]:XPOSITION:RTARget

CALCulate:EVALuate:Y[1]:XPOSITION:RTARget

Moves the marker to the next occurrence of the target value to the right of the present marker position. (SEARCH RIGHT under **Search**); No Query)

:Y[1]:XPOSITION:TARGet <numeric>

CALCulate:EVALuate:Y[1]:XPOSITION:TARGet <numeric>

Sets the target value. The target search function moves the marker to a specified target point on the trace. (TARGET under **Search**)

Parameter	Range	Unit
<numeric>	-100×10^6 to 100×10^6	y-axis unit

Query Response is <numeric>.

Related Command CALCulate:EVALuate:Y[1]:XPOSITION:LTARget
CALCulate:EVALuate:Y[1]:XPOSITION:RTARget

Example OUTPUT @Hp4291;"CALC:EVAL:Y:XPOS:TARG 0"
OUTPUT @Hp4291;"CALC:EVAL:Y:XPOS:TARG?"
ENTER @Hp4291;A

:Y[1]:XPOSITION:TRACk {MAXimum|MINimum|TARGet|PEAK|OFF}

CALCulate:EVALuate:Y[1]:XPOSITION:TRACk {MAXimum|MINimum|TARGet|PEAK|OFF}

Sets the search tracking function. (SRCH TRACK ON off SEARCH: MAX, MIN, TARGET, PEAK under **Search**)

Parameter	Description
MAXimum	Maximum search (SRCH TRACK ON off SEARCH: MAX)
MINimum	Minimum search (SRCH TRACK ON off MIN)
TARGet	Target search (SRCH TRACK ON off TARGET)
PEAK	Peak search (SRCH TRACK ON off PEAK)
OFF	Search tracking function OFF (SRCH TRACK on OFF)

Query Response is {MAX|MIN|TARG|PEAK|OFF}.

```

Example          OUTPUT @Hp4291;"CALC:EVAL:Y:XPOS:TRAC MAX"
                 OUTPUT @Hp4291;"CALC:EVAL:Y:XPOS:TRAC?"
                 ENTER @Hp4291;A$

```

CALCulate:FORMat {MLINear|PHASe|UPHase|REAL|IMAGinary|LFACTor|LTANGent|CP|CS|LP|LS|D|Q|RP|RS|COMPLex}

CALCulate:FORMat {MLINear|PHASe|UPHase|REAL|IMAGinary|LFACTor|LTANGent|CP|CS|LP|LS|D|Q|RP|RS|COMPLex}

Set the measurement parameter. It formats vector measurement data, Z (impedance), Y (admittance), Γ (reflection coefficient), into scalar measurement parameters, such as $|Z|$, θ , inductance, capacitance, and so on. For overall flow of data formatting, see Appendix E.

Parameter	Description
MLINear	Linear magnitude
PHASe	Phase
UPHase	Expanded phase (unwrapped phase)
REAL	Real part
IMAGinary	Imaginary part
LFACTor	Loss factor ($\epsilon r''$ and $\mu r''$; Option 002 only)
LTANGent	Loss tangent ($\tan \delta$; Option 002 only)
CP	Equivalent parallel capacitance
CS	Equivalent series capacitance
LP	Equivalent parallel inductance
LS	Equivalent series inductance
D	Dissipation factor
Q	Quality factor
RP	Equivalent parallel resistance
RS	Equivalent series resistance
COMPLex	Complex plane

Query Response is {MLIN|PHAS|UPH|REAL|IMAG|CP|CS|LP|LS|D|Q|RP|RS|COMP}.

```

Related Command  CALCulate:MATH1[:EXPRession]:NAME
                 CALCulate:MATH1:STATe
                 DISPlay[:WINDow]:TRACe{1-21}:GRATICule:FORMat
                 DISPlay[:WINDow]:TRACe{1-21}:GRATICule:Y:SPACing

```

```

Example          To display the admittance chart,
                 OUTPUT @Hp4291;"CALC:MATH1:NAME ADM;STAT ON"
                 OUTPUT @Hp4291;"CALC:FORM COMP"
                 OUTPUT @Hp4291;"DISP:TRAC:GRAT:FORM ADM"

```

Query,

```

                 OUTPUT @Hp4291;"CALC:FORM?"
                 ENTER @Hp4291;A$

```

CALCulate:FORMat:UNIT:ANGLE {DEG|RAD}

CALCulate:FORMat:UNIT:ANGLE {DEG|RAD}

Selects the unit of phase format. (PHASE UNIT [] under (Format))

Parameter	Description
DEG	Degree.
RAD	Radian.

Query response is {DEG|RAD}.

CALCulate:LIMit Subsystem

This subsystem is used to set the limit test function.

:BEEPer {OFF|ON|0|1}

CALCulate:LIMit:BEEPer {OFF|ON|0|1}

See "CALCulate:LIMit:BEEPer[:STATe] {OFF|ON|0|1}" command.

:BEEPer:CONDition {PASS|FAIL}

CALCulate:LIMit:BEEPer:CONDition {PASS|FAIL}

Sets limit test beeper. (BEEP: OFF, PASS, FAIL under (System)BEEP [])

Parameter	Description
PASS	Beeps when limit test result passes. (PASS)
FAIL	Beeps when limit test result fails. (FAIL)

Query response is {PASS|FAIL}.

Related Command CALCulate:LIMit:BEEPer[:STATe]

Example

```

OUTPUT @Hp4291;"CALC:LIM:BEEP:COND PASS"
OUTPUT @Hp4291;"CALC:LIM:BEEP ON"

OUTPUT @Hp4291;"CALC:LIM:BEEP:COND?"
ENTER @Hp4291;A,B

```

:BEEPer[:STATe] {OFF|ON|0|1}

CALCulate:LIMit:BEEPer[:STATe] {OFF|ON|0|1}

Turns the limit test beeper ON or OFF. When limit testing is ON and the beeper is ON, a beep is emitted each time a limit test is performed and the result defined by CALCulate:LIMit:BEEPer:CONDition is detected. (BEEP [] under (System) LIMIT MENU)

Parameter	Description
OFF or 0	Limit test beeper OFF.
ON or 1	Limit test beeper ON.

Query Response is {0|1}.

Related Command CALCulate:LIMit:BEEPer:CONDition

:CLEAr

CALCulate:LIMit:CLEAr

Clears all segments in the limit line. (CLEAR LIST YES under (System); No query)

:CONTRol:OFFSet <numeric>

CALCulate:LIMit:CONTRol:OFFSet <numeric>

Adds or subtracts an offset from the stimulus value of the limit line.
(STIMULUS OFFSET under (System) LIMIT MENU)

Parameter	Range	Unit
<numeric>	-1.8×10^9 (= -1.8 G) to 1.8×10^9 (= 1.8 G)	Hz (frequency sweep)
	-1 to 1	V (OSC level)
	-40 to 40	V (DC-V sweep)
	-100×10^{-3} to 100×10^{-3}	A (DC-I sweep)

Query Response is <numeric>.

Example OUTPUT @Hp4291;"CALC:LIM:CONT:OFFS 0"
 OUTPUT @Hp4291;"CALC:LIM:CONT:OFFS?"
 ENTER @Hp4291;A

:LINE {OFF|ON|0|1}

CALCulate:LIMit:LINE {OFF|ON|0|1}

Displays limit lines. (LIMIT LINE ON off under (System))

Parameter	Description
OFF or 0	Do not display limit lines.
ON or 1	Display limit lines.

Query Response is {0|1}.

Example OUTPUT @Hp4291;"CALC:LIM:LINE ON"
 OUTPUT @Hp4291;"CALC:LIM:LINE?"
 ENTER @Hp4291;A

:OFFSet {<numeric>|MARKer}

CALCulate:LIMit:OFFSet {<numeric>|MARKer}

Adds or subtracts an offset from the amplitude value of the limit line.

(AMPLITUDE OFFSET under (System) LIMIT MENU)

Parameter	Range or Description	Unit
<numeric>	-1×10^9 to 1×10^9	y-axis unit
MARKer	Sets the offset value to the marker value. (MKR → AMP. OFS)	

Query Response is <numeric>.

Example

```
OUTPUT @Hp4291;"CALC:LIM:OFFS 0"
OUTPUT @Hp4291;"CALC:LIM:OFFS?"
ENTER @Hp4291;A
```

:SAVE

CALCulate:LIMit:SAVE

Completes editing the limit table. (DONE under (System); No query)

:SEGment <numeric>

CALCulate:LIMit:SEGment <numeric>

Specifies which limit segment in the table to edit. (SEGMENT under (System))

Parameter	Description
<numeric>	Segment number, 1 to 18.

Query Response is <numeric>.

Related Command CALCulate:LIMit:SEGment:EDIT

Example

To edit a segment,

```
OUTPUT @Hp4291;"CALC:LIM:SEGM 1"
OUTPUT @Hp4291;"CALC:LIM:SEGM:EDIT"
OUTPUT @Hp4291;"CALC:LIM:SEGM:CONT 10MAHZ"
OUTPUT @Hp4291;"CALC:LIM:SEGM:UPP 100"
OUTPUT @Hp4291;"CALC:LIM:SEGM:LOW 80"
OUTPUT @Hp4291;"CALC:LIM:SEGM:SAVE"
```

Query,

```
OUTPUT @Hp4291;"CALC:LIM:SEG?"
ENTER @Hp4291;A
```

:SEGment:ADD

CALCulate:LIMit:SEGment:ADD

Adds a new segment to the end of the limit table. (**ADD** under **System**); No query)

:SEGment:CONTRol[:DATA] <numeric>

CALCulate:LIMit:SEGment:CONTRol[:DATA] <numeric>

Sets the stimulus value of a segment. (**STIMULUS** under **System**)

Parameter	Range	Unit
<numeric>	1000000 (-1 M) to 1.8×10^9 (-1.8 G)	Hz (frequency sweep)
	200×10^{-6} to 1	V (OSC level)
	-40 to 40	V (DC-V sweep)
	-100×10^{-3} to 100×10^{-3}	A (DC-I sweep)

Query Response is <numeric>.

```
Example      OUTPUT @Hp4291;"CALC:LIM:SEGM:CONT 1MAHZ"
              OUTPUT @Hp4291;"CALC:LIM:SEGM:CONT?"
              ENTER @Hp4291;A
```

:SEGment:DELeTe

CALCulate:LIMit:SEGment:DELeTe

Deletes a limit testing segment. (**DELETE** under **System**); No query)

:SEGment:DELTA <numeric>

CALCulate:LIMit:SEGment:DELTA <numeric>

Sets the limits an equal amount above and below a specified middle value, instead of setting upper and lower limits separately. (**DELTA LIMITS** under **System**)

Parameter	Range	Unit
<numeric>	0 to 1×10^9	y-axis unit

Query Response is <numeric>.

```
Example      OUTPUT @Hp4291;"CALC:LIM:SEGM:DELT 0"
              OUTPUT @Hp4291;"CALC:LIM:SEGM:DELT?"
              ENTER @Hp4291;A
```

:SEGMENT:EDIT

CALCulate:LIMit:SEGMENT:EDIT

Start editing the segment. (**EDIT** under **(System)**); No query)

Related Command CALCulate:LIMit:SEGMENT

:SEGMENT:LOWer <numeric>

CALCulate:LIMit:SEGMENT:LOWer <numeric>

Sets the lower limit value for the segment. (**LOWER LIMIT** under **(System)**)

Parameter	Range	Unit
<numeric>	-1×10^9 to 1×10^9	y-axis unit

Query Response is <numeric>.

Example OUTPUT @Hp4291;"CALC:LIM:SEGM:LOW 0"
 OUTPUT @Hp4291;"CALC:LIM:SEGM:LOW?"
 ENTER @Hp4291;A

:SEGMENT:MIDDLE {<numeric>|MARKer}

CALCulate:LIMit:SEGMENT:MIDDLE {<numeric>|MARKer}

Sets the midpoint for delta limits. (**MIDDLE VALUE** under **(System)**)

Parameter	Range or Description	Unit
<numeric>	-1×10^9 to 1×10^9	y-axis unit
MARKer	Using the marker to set the middle amplitude value of a limit segment.	

Query Response is <numeric>.

Example OUTPUT @Hp4291;"CALC:LIM:SEGM:MIDD 0"
 OUTPUT @Hp4291;"CALC:LIM:SEGM:MIDD?"
 ENTER @Hp4291;A

:SEGMENT:SAVE

CALCulate:LIMit:SEGMENT:SAVE

Terminates a limit segment definition. (**DONE** under **(System)**); No query)

Related Command CALCulate:LIMit:SEGMENT
 CALCulate:LIMit:SEGMENT:EDIT

:SEGment:UPPer <numeric>

CALCulate:LIMit:SEGment:UPPer

Sets the upper limit value for a limit testing segment. (**UPPER LIMIT** under **(System)**)

Parameter	Range	Unit
<numeric>	-1×10 ⁹ to 1×10 ⁹	y-axis unit

Query Response is <numeric>.

Example

```
OUTPUT @Hp4291;"CALC:LIM:SEGM:UPP 0"
OUTPUT @Hp4291;"CALC:LIM:SEGM:UPP?"
ENTER @Hp4291;A
```

:STATe {OFF|ON|0|1}

CALCulate:LIMit:STATe {OFF|ON|0|1}

Sets the limit testing ON or OFF. (**LIMIT TEST ON off** under **(System)**)

Parameter	Description
OFF or 0	Limit testing OFF
ON or 1	Limit testing ON

Query Response is {0|1}.

Example

```
OUTPUT @Hp4291;"CALC:LIM:STAT ON"
OUTPUT @Hp4291;"CALC:LIM:STAT?"
ENTER @Hp4291;A
```

CALCulate:MATH1 Subsystem

This subsystem is used to control conversion, which calculates raw data into the measurement parameter, such as impedance, admittance, or reflection coefficient. For an overall flow of data conversion, see Appendix E.

Related Command CALCulate:FORMat
CALCulate:FORMat:UNIT:ANGLE

:CATalog?

CALCulate:MATH1:CATalog?

See "CALCulate:MATH1[:EXPReSSion]:CATalog?" command.

Parameter	Description
ADM	Admittance parameters
DCO	Permittivity (dielectric constant)
PER	Permeability
RCO	Reflection coefficient parameters

Query Response is {ADM|DCO|PER|RCO}.

Related Command CALCulate:FORMat
CALCulate:MATH1:STATe

Example To select ADMITTANCE: MAG(|Y|),

```
OUTPUT @Hp4291;"CALC:MATH1:NAME ADM"
OUTPUT @Hp4291;"CALC:MATH1:STAT ON"
OUTPUT @Hp4291;"CALC:FORM MLIN"
```

Query,

```
OUTPUT @Hp4291;"CALC:MATH1:NAME?"
ENTER @Hp4291;A$
```

:STATe {OFF|ON|0|1}

CALCulate:MATH1:STATe {OFF|ON|0|1}

Sets the measurement parameter of the active channel, converting impedance to an admittance or reflection coefficient. For an overall flow of the measurement parameter settings, see Appendix E.

Parameter	Description
OFF or 0	Conversion OFF (Impedance measurement)
ON or 1	Conversion ON (Enable CALCulate:MATH1[:EXPRession]:NAME)

Query Response is {0|1}.

Related Command CALCulate:FORMat
CALCulate:MATH1[:EXPRession]:NAME

CALCulate:MATH2 Subsystem

This subsystem is used to control the data math function.

[:EXPRession]:CATalog?

CALCulate:MATH2[:EXPRession]:CATalog?

Returns the available parameters that can be used with the CALCulate:MATH2[:EXPRession]:NAME command. (Query only)

Query response is "ADD", "SUB", "DIV", "MUL".

Example OUTPUT @Hp4291;"CALC:MATH2:CAT?"
ENTER @Hp4291;A\$

[:EXPRession] :NAME { SUB | ADD | DIV | MUL }

CALCulate:MATH2[:EXPRession]:NAME {SUB|ADD|DIV|MUL}

Sets the data math operation. (DATA-MEM , DATA+MEM , DATA/MEM , DATA*MEM under

 DATA MATH [])

Parameter	Description
ADD	Adds the memory trace to the data trace. (DATA+MEM)
DIV	Divides the data trace by the memory trace. (DATA/MEM)
SUB	Subtracts the memory trace from the data trace. (DATA-MEM)
MUL	Multiplies the data trace by the memory trace. (DATA*MEM)

Query Response is {ADD|DIV|SUB|MUL}.

Related Command CALCulate:MATH2:STATe

Example OUTPUT @Hp4291;"CALC:MATH2:NAME SUB;STAT ON"
 OUTPUT @Hp4291;"CALC:MATH2:NAME?"
 ENTER @Hp4291;A\$

:STATe { OFF | ON | 0 | 1 }

CALCulate:MATH2:STATe {OFF|ON|0|1}

Sets data math function ON or OFF.

Parameter	Description
OFF or 0	Data math function OFF
ON or 1	Data math function ON

Query Response is {0|1}.

Related Command CALCulate:MATH2[:EXPRession]:NAME

CENT <numeric>, CHAN{1|2}, CLEM, CONT

(*Simple Command*) See “Simple Command” later in this chapter.

DATA Subsystem

The DATA subsystem is used to send or get the following values:

- The parameters used for display offset.
- The parameters used for equivalent circuit simulations.
- The raw data arrays, data arrays, memory arrays, and fixture compensation standard arrays, which are complex data.
- The limit test results.
- The level monitor data array.
- The stimulus array.
- The stimulus, data, and memory of the specified point.

DATA[:DATA] {AOFF|GAIN|MZAP},<numeric>

DATA[:DATA] {AOFF|GAIN|MZAP},<numeric>

Defines the following values.

Parameter	Description
AOFF	Defines the auxiliary part of the offset value when using the Smith, polar, and admittance chart format. (AUX OFFSET VALUE under Display)
GAIN	Defines the gain value of the data math function. (GAIN under Display)
MZAP	Defines the zooming aperture value as a percentage of the span. (ZOOMING APERTURE under Marker→)

Parameter	Range	Unit
<numeric> (AOFF)	-100×10^6 to 100×10^6	—
(GAIN)	-1×10^6 to 1×10^6	—
(MZAP)	0 to 100	—

Query response is <numeric>.

```
Example      OUTPUT @Hp4291;"DATA AOFF,0"
             OUTPUT @Hp4291;"DATA? AOFF"
             ENTER @Hp4291;A
```

DATA[:DATA] OFFS, {<numeric>|MARKer}

DATA[:DATA] OFFS, {<numeric>|MARKer}

Defines the offset value. (OFFSET under Display)

Parameter	Range or Description	Unit
<numeric>	-100×10^6 to 100×10^6	—
MARKer	Sets the marker's amplitude value into the offset value. (MKR→OFFSET under Display)	

Query response is <numeric>.

Example OUTPUT @Hp4291;"DATA GAIN,1"
 OUTPUT @Hp4291;"DATA OFFS,0"

DATA[:DATA] <array>, {<block>|<numeric 11>, <numeric 12>, ... , <numeric n2>}

DATA[:DATA] <array>, {<block>|<numeric 11>, <numeric 12>, ..., <numeric n1>, <numeric n2>}

Send data arrays.

Parameter	Description
<array>	CCO{11-33}, CMP{1-3}, DATA, RAW, OADM, SIMP, LIMP, MON

Where,

CCO11, CCO12, CCO13 : Calibration coefficient arrays, A_1 , B_1 , C_1 at normal-vl range (real-imaginary)
 CCO21, CCO22, CCO23 : Calibration coefficient arrays, A_1 , B_1 , C_1 at normal-vh range (real-imaginary)
 CCO31, CCO32, CCO33 : Calibration coefficient arrays, A_1 , B_1 , C_1 at expanded range (real-imaginary)
 CMP1, CMP2, CMP3 : Compensation coefficient arrays, A_3 , B_3 , C_3 (real-imaginary)¹
 DATA : Data array (real-imaginary)
 RAW : Raw data array (real-imaginary)
 OADM : Fixture compensation open standard array (real-imaginary)
 SIMP : Fixture compensation short standard array (real-imaginary)
 LIMP : Fixture compensation load standard array (real-imaginary)
 MON : Level monitor array (scalar)²

1 Before executing the query command, set the compensation function ON.

2 When the level monitor function is OFF, the value is not updated. Before query, set the level monitor ON.

Parameter	Description
<block>	Binary data transfer format
<numeric 11>, ... , <numeric n2>	ASCII data transfer format (real-imaginary)
<numeric 1>, ... , <numeric n>	ASCII data transfer format (scalar)

The array dimension is defined as:

CCO{11-33}, CMP{1-3}, DATA, RAW : n (the number of measurement points) × 2 (real-imaginary)
 OADM, SIMP, LIMP : n (the number defined by DATA:DEFine) × 1 (scalar)
 MON : n (the number of measurement points) × 1 (scalar)

Query response is:

{<block>|<numeric 11>, <numeric 12>, ... , <numeric n1>, <numeric n2>} (real-imaginary).
 {<block>|<numeric 1>, ... , <numeric n>} (scalar).

Example DIM A(1:201,1:2)
 OUTPUT @Hp4291;"DATA? CCO11"
 ENTER @Hp4291;A(*)

DATA[:DATA] {EQC0|EQC1|EQL1|EQR1}, <numeric>

DATA[:DATA] {EQC0|EQC1|EQL1|EQR1}, <numeric>

Defines the specified equivalent circuit parameter. (PARAMETER R1, C1, L1, CO under Display)

Parameter	Description
EQC0	Parameter C0
EQC1	Parameter C1
EQL1	Parameter L1
EQR1	Parameter R1

Parameter	Range	Unit
<numeric>	-1×10^{18} to 1×10^{18}	F (EQC0, EQC1) H (EQL1) Ω (EQR1)

Query response is <numeric>.

```
Example          OUTPUT @Hp4291;"DATA EQC0,1PF"
                  OUTPUT @Hp4291;"DATA? EQC0"
                  ENTER @Hp4291;A
```

DATA[:DATA]? LFA

DATA[:DATA]? LFA

Returns the limit test result of the fail points. (Query only)

Query response is <numeric1(stim)>,0,<numeric1(upp)>,<numeric1(low)>,<numeric2(stim)>, ... ,<numeric n(low)>. (ASCII data transfer format)

Where,

<numeric1(stim)>, ... ,<numeric n(stim)> : Stimulus at each fail point.
 <numeric1(upp)>, ... ,<numeric n(upp)> : Upper limit value at each fail point.
 <numeric1(low)>, ... ,<numeric n(low)> : Lower limit value at each fail point.

The array dimension is defined as,

n (the number of limit test fail points) \times 4 (stimulus, 0, upper limit value, lower limit value)

DATA[:DATA]? LLIS

DATA[:DATA]? LLIS

Returns the limit test result of all measurement points. (Query only)

Query response is <numeric1(stim)>,{-1|0|1},<numeric1(upp)>,<numeric1(low)>,<numeric2(stim)>, ... ,<numeric n(low)>. (ASCII data transfer format)

Where,

<numeric1(stim)>, ... ,<numeric n(stim)> : Stimulus at each measurement point.
 {-1|0|1} : -1 means NO TEST, 0 means FAIL, and 1 means PASS.
 <numeric1(upp)>, ... ,<numeric n(upp)> : Upper limit value at each point.
 <numeric1(low)>, ... ,<numeric n(low)> : Lower limit value at each point.

The array dimension is defined as,

n (the number of the measurement points) \times 4 (stimulus, {-1|0|1}, upper limit value, lower limit value)

DATA[:DATA]? LMAR

DATA[:DATA]? LMAR

Returns the limit test result at the marker. (Query only)

Query response is $\langle \text{numeric}(\text{stim}) \rangle, \{-1 | 0 | 1\}, \langle \text{numeric}(\text{upp}) \rangle, \langle \text{numeric}(\text{low}) \rangle$. (ASCII data transfer format)

Where,

$\langle \text{numeric}(\text{stim}) \rangle$: stimulus
 $\{-1 | 0 | 1\}$: -1 means NO TEST, 0 means FAIL, and 1 means PASS.
 $\langle \text{numeric}(\text{upp}) \rangle$: upper limit value
 $\langle \text{numeric}(\text{low}) \rangle$: lower limit value

DATA[:DATA]? MEM

DATA[:DATA]? MEM

Returns the selected memory array. (Query only)

Query response is $\{\langle \text{block} \rangle | \langle \text{numeric}1 \rangle, \langle \text{numeric}2 \rangle, \dots, \langle \text{numeric } n1 \rangle, \langle \text{numeric } n2 \rangle\}$.

Where,

$\langle \text{block} \rangle$: Binary data transfer format
 $\langle \text{numeric}1 \rangle, \dots, \langle \text{numeric } n2 \rangle$: ASCII data transfer format

The array dimension is defined as,
 $n(\text{number of measurement points}) \times 2$ (real-imaginary).

Related Command `DISPlay[:WINDow]:TRACe{1-21}:STATe`**DATA[:DATA]? SPAR**

DATA[:DATA]? {MON|SPAR}

Returns the stimulus array (scalar). (Query only)

Query response is $\{\langle \text{block} \rangle | \langle \text{numeric}1 \rangle, \langle \text{numeric}2 \rangle, \dots, \langle \text{numeric } n \rangle\}$.

Where,

$\langle \text{block} \rangle$: Binary data transfer format
 $\langle \text{numeric}1 \rangle, \dots, \langle \text{numeric } n \rangle$: ASCII data transfer format

The array dimension is defined as,
 $n(\text{the number of the measurement points}) \times 1$

DATA[:DATA]:VALue? {SPAR|DATA|MEM|MON}, <numeric>

DATA[:DATA]:VALue? {SPAR|DATA|MEM}, <numeric>

Returns the value at a specified point. (Query only)

Parameter	Range or Description
SPAR	Stimulus
DATA	Measurement data of active channel
MEM	Selected memory of active channel
$\langle \text{numeric} \rangle$	1 to the number of points.

Query response is $\langle \text{numeric} \rangle$.

DATA:DEFine {OADM|SIMP|LIMP},{<numeric>|DATA|DTR|TR1}

DATA:DEFine {OADM|SIMP|LIMP},{<numeric>|DATA|DTR|TR1}

Sets the fixture compensation standard arrays.

Parameter	Description
OADM	Fixture compensation open standard array
SIMP	Fixture compensation short standard array
LIMP	Fixture compensation load standard array

Parameter	Range or Description
<numeric>	1 to number of measurement points. Sets the number of data points that are sent by DATA[:DATA] {OADM SIMP LIMP}. The fixture compensation standard array is set by DATA[:DATA] {OADM SIMP LIMP}.
DATA	Copies data array to the fixture compensation standard arrays.
DTR	Copies data trace array to the fixture compensation standard arrays.
TR1	Copies data trace array to the fixture compensation standard arrays.

Query response is <numeric>.

```

Example          OUTPUT @Hp4291;"DATA:DEF OADM,DATA"
                  OUTPUT @Hp4291;"DATA:DEF? OADM"
                  ENTER @Hp4291;A

```

DATA:DELeTe {OADM|SIMP|LIMP}

DATA:DELeTe {OADM|SIMP|LIMP}

Clears the fixture compensation standard array. (No query)

After clear, SENSE:CORREction2:CKIT[1]:STANDARD{1-3}[:SElect] LPARAmeter is selected.

Parameter	Description
OADM	Clears the fixture compensation open standard array.
SIMP	Clears the fixture compensation short standard array.
LIMP	Clears the fixture compensation load standard array.

DATA:POINT? LFA

DATAPOINT? LFA

Returns the number of the limit test fail points. (Query only)

Query response is <numeric>.

DATMEM

(*Simple Command*) See “Simple Commands” later in this chapter.

DIAGnostic Subsystem

This subsystem provides the analyzer’s hardware and firmware tests, and the service related functions for maintenance and repair.

DIAGnostic:EREFerence:STATe?

DIAGnostic:EREFerence:STATe?

Queries whether the external reference signal is connected to the external reference input on the rear panel. (Query only)

Query response is {0|1}.

Where,

0 : Not connected.
1 : Connected.

Example OUTPUT @Hp4291;"DIAG:EREF:STAT?"
 ENTER @Hp4291;A

DIAGnostic:FREVision?

DIAGnostic:FREVision?

Returns the model number (HP4291B) and the firmware revision (for example, REV1.00). (Query only)

Query response is <string>.

Example OUTPUT @Hp4291;"DIAG:FREV?"
 ENTER @Hp4291;A\$

DIAGnostic:INIT:RESult?

DIAGnostic:INIT:RESult?

Returns the power-on test result. (Query only)

Query response is {PASS|FAIL}.

Where,

PASS : Power-on test passed.
FAIL : Power-on test failed.

Example OUTPUT @Hp4291;"DIAG:INIT:RES?"
 ENTER @Hp4291;A\$

DIAGnostic:SERVice Subsystem

This subsystem provides the required functions for maintenance and repair. This manual does not describe the DIAGnostic:SERVice subsystem. See the *Service Manual* for the command and function descriptions. (SERVICE under (System) SERVICE MENU)

DIAGnostic:TEST Subsystem

This subsystem provides the self-tests for maintenance and repair. This manual does not describe the DIAGnostic:TEST subsystem. See the *Service Manual* for the command and function descriptions. (TESTS under (System) SERVICE MENU)

DFLT, DISP {DATA|MEMO|DATM}

(*Simple Command*) See “Simple Commands” later in this chapter.

DISPlay Subsystem**DISPlay:ANNotation:FREQuency {OFF|ON|0|1}**

DISPlay:ANNotation:FREQuency OFF

Blanks the frequency notation for security purposes. Frequency notation is not displayed after the command shown in example is executed.

The frequency notation can only be restored by sending the :SYSTEM:PRESet ((Preset)) or *RST command, or by turning the power OFF and ON. (FREQUENCY BLANK under Display)

Parameter	Description
OFF or 0	Frequency notation OFF.
ON or 1	Frequency notation ON.

Query response is {0|1}.

Related Command SYSTEM:SECurity[:STATe]

Example To blank the frequency notation,

```

OUTPUT @Hp4291;"DISP:ANN:FREQ OFF"
OUTPUT @Hp4291;"SYST:SEC ON"

Query,

OUTPUT @Hp4291;"DISP:ANN:FREQ?"
ENTER @Hp4291;A

```

DISPlay:BACKlight {OFF|ON|0|1}

DISPlay:BACKlight {OFF|ON|0|1}

Sets backlighting the LCD ON or OFF.

Parameter	Description
OFF or 0	Backlighting OFF
ON or 1	Backlighting ON

Query response is {0|1}.

Example OUTPUT @Hp4291;"DISP:BACK OFF"

```

OUTPUT @Hp4291;"DISP:BACK?"
ENTER @Hp4291;A

```

DISPlay:BRIGhtness <numeric>

DISPlay:BRIGhtness <numeric>

Sets the display intensity as a percent of the brightest setting. (INTENSITY under Display)

Parameter	Range	Unit
<numeric>	0 to 1	—

Query Response is <numeric>.

Example

```
OUTPUT @Hp4291;"DISP:BRIG 0.83"  
OUTPUT @Hp4291;"DISP:BRIG?"  
ENTER @Hp4291;A
```

DISPlay:BRIGhtness**DISPlay:CMAP Subsystem**

This subsystem is used to adjust and control the colors of the display.

:COLor{1-14}:DEFault

DISPlay:CMAP:COLor{1-14}:DEFault

Resets the color being modified to the default color. (**RESET COLOR** under **Display**); No query)

Node	Description
COLor1	Channel 1 data.
COLor2	Channel 1 memory and limit lines.
COLor3	Channel 2 data.
COLor4	Channel 2 memory and limit lines.
COLor5	Graticule and a portion of softkey text.
COLor6	Warning annotation.
COLor7	All the non-data text.
COLor8	Text on the IBASIC screen.
COLor{9-14}	Pen 1-6.

:COLor{1-14}:HSL <numeric(Hue)>,<numeric(Sat)>,<numeric(Lum)>

DISPlay:CMAP:COLor{1-14}:HSL <numeric(Hue)>,<numeric(Sat)>,<numeric(Lum)>

Change color of the display element. (**MODIFY COLORS** under **Display**)

The :COLor{1-14} numbering is the same as the "DISPlay:CMAP:COLor{1-14}:DEFault" command numbering.

Parameter	Range or Description	Unit
<numeric(hue)>	(Hue) 0 to 1, circular, with a value of 0 resulting in the same hue as a value of 1. The approximate color is (starting at 0): red, orange, yellow, green, cyan, blue, magenta, and back to red.	—
<numeric(sat)>	(Saturation) 0 to 1, with 0 specifying no color (only white or gray, depending on intensity) and 1 specifying no white.	—
<numeric(lum)>	(Luminance) 0 to 1, with 0 resulting in black and 1 resulting in the brightest color available.	—

Query response is <numeric(hue)>,<numeric(sat)>,<numeric(lum)>.

Example OUTPUT @Hp4291;"DISP:CMAP:COL:HSL 0.2,1,1"
 OUTPUT @Hp4291;"DISP:CMAP:COL:HSL?"
 ENTER @Hp4291;A,B,C

:DEFault

DISPlay:CMAP:DEFault

Sets all the color settings back to the default values. (DEFAULT COLORS under Display); No query)

:LOAD

DISPlay:CMAP:LOAD

Recalls the previously saved color settings from the non-volatile memory. (RECALL COLORS under Display); No query)

:STORe

DISPlay:CMAP:STORe

Saves the current color setting to the non-volatile memory. (SAVE COLORS under Display); No query)

DISPlay:CONTRast <numeric>

DISPlay:CONTRast <numeric>

Sets the background intensity of the display as a percent of the white level. (BACKGROUND INTENSITY under Display)

Parameter	Range	Unit
<numeric>	0 to 1	—

Query response is <numeric>.

```
Example          OUTPUT @Hp4291;"DISP:CONT 0"
                  OUTPUT @Hp4291;"DISP:CONT?"
                  ENTER @Hp4291;A
```

DISPlay[:WINDow]:ALLocation {INSTrument|HIHB|BASic|BSTatus}

DISPlay[:WINDow]:ALLocation {INSTrument|HIHB|BASic|BSTatus}

Selects the display allocation mode. (Option 1C2 only) (DISPLAY ALLOCATION under Display)

Parameter	Description
INSTrument	All instrument.
HIHB	Half Instrument and half HP instrument BASIC.
BASic	All HP instrument BASIC.
BSTatus	HP instrument BASIC status.

Query response is {INST|HIHB|BAS|BST}.

```
Example          OUTPUT @Hp4291;"DISP:ALL HIHB"
                  OUTPUT @Hp4291;"DISP:ALL?"
                  ENTER @Hp4291;A$
```

DISPlay[:WINDow]:FORMat {FBACk|ULOWer}

DISPlay[:WINDow]:FORMat {FBACk|ULOWer}

Sets the full-screen or split display when the dual channel mode. (**SPLIT DISP ON off** under **Display**)

Parameter	Description
FBACk	Full-screen single graticule display.
ULOWer	Split display with two half-screen graticules.

Query response is {FBAC|ULOW}.

Example OUTPUT @Hp4291;"DISP:FORM ULOW"
 OUTPUT @Hp4291;"DISP:FORM?"
 ENTER @Hp4291;A\$

DISPlay[:WINDow]:GRAPhics:STATe {OFF|ON|0|1}

DISPlay[:WINDow]:GRAPhics:STATe {OFF|ON|0|1}

Sets the HP instrument BASIC graphic function ON or OFF. If the HP instrument BASIC graphic function is ON, extra (more than one) memory traces cannot be displayed.

(**GRAPHIC: BASIC DRAM** under **Display**)

Parameter	Description
OFF or 0	HP instrument BASIC graphic OFF. Display all memory traces. (ALL MEMORY TRACE)
ON or 1	HP instrument BASIC Graphic ON. Display a selected memory trace. (GRAPHIC: BASIC DRAM)

Query response is {0|1}.

DISPlay[:WINDow]:TEXT{1-40} Subsystem

This subsystem is used to display the following information.

Node	Description
:TEXT1	Tabular listing of all the measured data points and their current values.
:TEXT2	Tabular listing of the key parameters for both channels.
:TEXT3	Tabular listing of the calibration kit definition.
:TEXT4	Tabular listing of the compensation standard definitions.
:TEXT5	List sweep table, by the "start and stop" format.
:TEXT6	List sweep table, by the "center and span" format.
:TEXT7	Limit testing table, by the "Upper and Lower" format.
:TEXT8	Limit testing table, by the "Mid and Delta" format.
:TEXT9	Maker list.
:TEXT10	Title.
:TEXT{11-30}	Label.
:TEXT{31-34}	User trace headline.
:TEXT{35-38}	User trace footnote.
:TEXT39	Equivalent circuit parameter.
:TEXT40	Equivalent circuit model.

:CLEar

DISPlay[:WINDow]:TEXT{11-30}:CLEar

Erases all labels. This command clears all labels at once whatever :TEXT{11-30} node is selected. (CLEAR ALL LABEL under (Display);No query)

Related Command DISPlay[:WINDow]:TEXT{11-30}[:DATA]

:COLor <numeric>

DISPlay[:WINDow]:TEXT{11-30}:COLor <numeric>

Selects the color of the specified label. (COLOR under (Display))

Node	Description
:TEXT{11-30}	the label 1 to 10

Parameter	Description
<numeric>	0 to 5, integer

Where,

0 : yellow
 1 : green
 2 : blue
 3 : orange
 4 : white
 5 : red

Query response is <numeric>.

[:DATA] <string>

DISPlay[:WINDow]:TEXT{10-38}[:DATA] <string>

Send the string to the following display area. (LABEL, TITLE, HEADLINE, FOOTNOTE under (System))

Node	Description
:TEXT10	The title area on the display of destination channel (TITLE)
:TEXT{11-30}	The label, common to channel 1,2 (LABEL)
:TEXT{31-34}	The user trace headline (HEADLINE)
:TEXT{35-38}	The user trace footnote (FOOTNOTE)

Parameter	Description
<string>	up to 53 characters

Query response is <string>.

Example OUTPUT @Hp4291;"DISP:TEXT17 ""COMMENT"""
 OUTPUT @Hp4291;"DISP:TEXT17?"
 ENTER @Hp4291;A\$

:LOCate <numeric(x)>[,<numeric(y)>]

DISPlay[:WINDow]:TEXT{11-30}:LOCate <numeric(x)>[,<numeric(y)>]

Defines where the specified label appears. (X POS, Y POS under (Display))

Parameter	Description
<numeric(x)>	X (horizontal), 0(left-most) to 609(right).
<numeric(y)>	Y (vertical), 0(bottom) to 421(top).

Query response is <numeric(x)>,<numeric(y)>.

:PAGE {UP|DOWN|<numeric>}

DISPlay[:WINDow]:TEXT{1-8}:PAGE {UP|DOWN|<numeric>}

Selects a page of a tabular list, which is defined as follows:

Node	Description
:TEXT1	Displays a tabular listing of all the measured data points and their current values. (LIST VALUES under (Copy))
:TEXT2	Provides a tabular listing on the display of the key parameters for both channels. (OPERATING PARAMETERS under (Copy))
:TEXT3	Shows the tabular listing of the calibration kit definition. (CAL KIT DEFINITION under (Copy))
:TEXT4	Provides the tabular listing of the compensation standard definitions. (COMPEN KIT DEFINITION under (Copy))
:TEXT5	Displays the list sweep table by the "Start & Stop" format. (DISPLAY: LIST under (Copy) LIST SWEEP TABLE)
:TEXT6	Displays the list sweep table by the "Center & Span" format. (DISPLAY: LIST under (Copy) LIST SWEEP TABLE)
:TEXT7	Displays the limit testing table by the "Upper & Lower" format. (DISPLAY: LIST under (Copy) LIMIT TEST TABLE)
:TEXT8	Displays the limit testing table by the "Mid & Delta" format. (DISPLAY: LIST under (Copy) LIMIT TEST TABLE)

Parameter	Description
UP	Displays the next page of information in a tabular listing. (NEXT PAGE under (Copy))
DOWN	Displays the previous page of information in a tabular listing. (PREV PAGE under (Copy))
<numeric>	Defines the displayed page number

Query response is <numeric>.

Related Command DISPlay[:WINDow]:TEXT{1-8}:STATe

Example

```
OUTPUT @Hp4291;"DISP:TEXT3:PAGE 1"
OUTPUT @Hp4291;"DISP:TEXT3:STAT ON"
OUTPUT @Hp4291;"DISP:TEXT3:PAGE UP"
```

:STATe {OFF|ON|0|1}

DISPlay[:WINDow]:TEXT{1-9|39|40}:STATe {OFF|ON|0|1}

Displays a tabular list.

Node	Description
:TEXT{1-8}	See "DISPlay[:WINDow]:TEXT{1-8}:PAGE {UP DOWN <numeric>}" command.
:TEXT9	Displays the maker list. (MKR LIST ON off under Utility)
:TEXT39	Displays the equivalent circuit parameter. (DISP EQV PARM [] under Display)
:TEXT40	Displays the equivalent circuit model. (EQIV CKT MENU under Display)

Parameter	Description
OFF or 0	Do not display the item.
ON or 1	Displays the item.

Query response is {0|1}.

DISPlay[:WINDow]:TRACe{1-21} Subsystem

This subsystem is used to control the displayed trace and the marker.

Node	Description
:TRACe1	Data trace
:TRACe{2-17}	Memory traces
:TRACe{18-21}	User traces

:CLEAr

DISPlay[:WINDow]:TRACe{2-21}:CLEAr

Clears memory traces or user traces. (CLEAR MEMORIES, CLEAR ALL UTRC under **Display**);No query)

Node	Description
:TRACe{2-17}	Memory traces (Clears all memory traces at once whatever :TRACe{2-17} node is selected; CLEAR MEMORIES)
:TRACe{18-21}	User traces (Clears all user traces at once whatever :TRACe{18-21} node is selected; CLEAR ALL UTRC)

Query response is {0|1}.

:GRATicule:AXIS:COUPlE {OFF|ON|0|1}

DISPlay[:WINDow]:TRACe{18-21}:GRATicule:AXIS:COUPlE {OFF|ON|0|1}

Sets the four user traces axis to coupled or uncoupled. (AXIS [] under (System))

Parameter	Description
OFF or 0	Uncouple the axis
ON or 1	Couple the axis

Query response is {0|1}.

Example

```
OUTPUT @Hp4291;"DISP:TRAC18:GRAT:AXIS:COUP ON"
OUTPUT @Hp4291;"DISP:TRAC18:GRAT:AXIS:COUP?"
ENTER @Hp4291;A
```

:GRATicule:FORMat {RECTangle|POLar|SMITH|ADMittance|CPLane}

DISPlay[:WINDow]:TRACe1:GRATicule:FORMat {RECTangle|POLar|SMITH|ADMittance|CPLane}

Selects the display format. (LIN Y-AXIS, LOG Y-AXIS, POLAR CHART, SMITH CHART, ADMITTANCE CHART, COMPLEX PLANE under (Format))

Parameter	Description
RECTangle	Rectangular format
SMITH	Smith chart format
POLar	Polar chart format
ADMittance	Admittance chart format
CPLane	Complex plane (In the permittivity or permeability measurement mode, the X-axis represents ϵ'_{r} or μ'_{r} , Y-axis represents ϵ''_{r} or μ''_{r} .)

Query response is {RECT|SMIT|POL|ADM|CPL}.

Related Command DISPlay[:WINDow]:TRACe{1-21}:Y:SPACing

Example To select LOG Y-AXIS,

```
OUTPUT @Hp4291;"DISP:TRAC:GRAT:FORM RECT"
OUTPUT @Hp4291;"DISP:TRAC:Y:SPAC LOG"
```

Query,

```
OUTPUT @Hp4291;"DISP:TRAC:GRAT:FORM?"
ENTER @Hp4291;A$
```

:GRATiCule:GRID[:STATe] {OFF|ON|0|1}

DISPlay[:WINDow]:TRACe{1-17}:GRATiCule:GRID[:STATe] {OFF|ON|0|1}

Sets the grid on the display of the selected channel ON or OFF. This command works the same whatever :TRACe{1-17} number is selected. (GRATICULE ON off under (Display))

Parameter	Description
OFF or 0	Grid OFF
ON or 1	Grid ON

Query response is {0|1}.

Example OUTPUT @Hp4291;"DISP:TRAC:GRAT:GRID ON"
 OUTPUT @Hp4291;"DISP:TRAC18:GRAT:GRID?"
 ENTER @Hp4291;A

:MARKer[1]:ALL DEFault

DISPlay[:WINDow]:TRACe{1-21}:MARKer[1]:ALL DEFault

Turns off all markers and cancels all settings of the marker functions. This command works the same whatever :TRACe{1-21} node is selected. (PRESET MKRS under (Marker); No query)

:MARKer[1]:ALL:STATe {OFF|ON|0|1}

DISPlay[:WINDow]:TRACe{1-21}:MARKer[1]:ALL:STATe {OFF|ON|0|1}

Turns the marker ON or OFF. This command works the same whatever :TRACe{1-21} node is selected.

Parameter	Description
OFF or 0	Turns the marker and all sub-markers OFF and preset the marker. (PRESET MKRS under (Display))
ON or 1	Turns ON the marker. (Marker)

Example OUTPUT @Hp4291;"DISP:TRAC:MARK:ALL:STAT ON"
 OUTPUT @Hp4291;"DISP:TRAC:MARK:ALL:STAT?"
 ENTER @Hp4291;A

:MARKer[1]:RELative {OFF|ON|0|1}

DISPlay[:WINDow]:TRACe{1-21}:MARKer[1]:RELative {OFF|ON|0|1}

Displays the Δ marker at the point of the marker and the marker mode changes to the Δ mode. This command works the same whatever :TRACe{1-21} node is selected. (Δ MKR, Δ MODE OFF under (Marker))

DISPlay[:WINDow]:TRACe{1-21}

Parameter	Description
OFF or 0	Turns off the Δ mode. (Δ MODE OFF)
ON or 1	Puts the Δ marker on the current position of the marker. (Δ MKR)

Query response is {0|1}.

Related Command

DISPlay[:WINDow]:TRACe{1-21}:MARKer1:RELative:REFerence

:MARKer[1]:RELative:REFerence {FIXed|MARKer|TRACked}

DISPlay[:WINDow]:TRACe{1-21}:MARKer[1]:RELative:REFerence {FIXed|MARKer|TRACked}

Changes the marker mode to the fixed, or tracking Δ mode. This command works the same whatever :TRACe{1-21} node is selected. (Δ MKR, FIXED Δ MKR, TRACKING Δ MKR), under (Marker)

Parameter	Description
FIXed	Sets a user-specified fixed reference marker. (FIXED Δ MKR)
MARKer	Puts the Δ marker on the current position of the marker. (Δ MKR)
TRACked	Puts a Δ marker at the present active marker position and turns on the tracking Δ marker. (TRACKING Δ MKR)

Query response is {FIX|MARK|TRAC}.

Example

```
OUTPUT @Hp4291;"DISP:TRAC:MARK:REL ON"
OUTPUT @Hp4291;"DISP:TRAC:MARK:REL:REF MARK"

OUTPUT @Hp4291;"DISP:TRAC:MARK:REL?"
ENTER @Hp4291;A
OUTPUT @Hp4291;"DISP:TRAC:MARK:REL:REF?"
ENTER @Hp4291;A$
```

:MARKer{2-8}:STATe {OFF|ON|0|1}

DISPlay[:WINDow]:TRACe{1-21}:MARKer{2-8}:STATe {OFF|ON|0|1}

Displays specified sub-marker at the point of the marker (ON). This command works the same whatever :TRACe{1-21} node is selected. (SUB MKR {1-7}) under (Marker)

Node	Description
MARKer{2-8}	Sub-marker1 to 7

Parameter	Description
OFF or 0	Erases specified sub-marker.
ON or 1	Displays specified sub-marker.

Query response is {0|1}.

Example

```
OUTPUT @Hp4291;"DISP:TRAC:MARK2:STAT ON"
OUTPUT @Hp4291;"DISP:TRAC:MARK2:STAT?"
ENTER @Hp4291;A
```

:MARKer{1-8}:UNIT {SPARameter|TIME|IOMega}

DISPlay[:WINDow]:TRACe{1-17}:MARKer{1-8}:UNIT {SPARameter|TIME|IOMega}

Selects the marker X-axis unit. This command works the same whatever :TRACe{1-17}, and :MARKer{1-8} node is selected. (MKR X AXIS [] under **Utility**)

Parameter	Description
SPARameter	Stimulus value (MKR X AXIS [STIM])
TIME	Time scale (MKR X AXIS [TIME])
IOMega	Angular velocity (MKR X AXIS [1/(2πF)])

Query response is {SPAR|TIME|IOM}.

:STATe {OFF|ON|0|1}

DISPlay[:WINDow]:TRACe{1-21}:STATe {OFF|ON|0|1}

Selects the trace to be displayed. (DISPLAY: DATA , MEMORY , DATA and MEMORY , SEL'D UTRC ON off under **Display**)

Node	Description
:TRACe1	Data trace
:TRACe{2-17}	Memory traces
:TRACe{18-21}	User traces

Parameter	Description
OFF or 0	Trace OFF.
ON or 1	Trace ON.

Query response is {0|1}.

Example

To display the data trace and memory trace1,

```
OUTPUT @Hp4291;"DISP:TRAC1:STAT ON"
OUTPUT @Hp4291;"DISP:TRAC2:STAT ON"
OUTPUT @Hp4291;"DISP:TRAC3:STAT OFF"
:
OUTPUT @Hp4291;"DISP:TRAC17:STAT OFF"
```

To display the user trace1,

```
OUTPUT @Hp4291;"DISP:TRAC1:STAT OFF"
:
OUTPUT @Hp4291;"DISP:TRAC17:STAT OFF"
OUTPUT @Hp4291;"DISP:TRAC18:STAT ON"
OUTPUT @Hp4291;"DISP:TRAC19:STAT OFF"
OUTPUT @Hp4291;"DISP:TRAC20:STAT OFF"
OUTPUT @Hp4291;"DISP:TRAC21:STAT OFF"
```

:X[:SCALe]:LEFT <numeric>

DISPlay[:WINDow]:TRACe{18-21}:X[:SCALe]:LEFT <numeric>

Defines the left edge value of the X-axis of the graticule for the user trace.

(LEFT VALUE under (System))

Node	Description
:TRACe{18-21}	User traces

Parameter	Range	Unit
<numeric>	-8.0×10^9 to 8.0×10^9	—

Query response is <numeric>.

:X[:SCALe]:RIGHT <numeric>

DISPlay[:WINDow]:TRACe{18-21}:X[:SCALe]:RIGHT <numeric>

Defines the right edge value of the X-axis of the graticule for the user trace.

(RIGHT VALUE under (System))

Node	Description
:TRACe{18-21}	User traces

Parameter	Range	Unit
<numeric>	-8.0×10^9 to 8.0×10^9	—

Query response is <numeric>.

:X[:SCALe]:RLEVel <numeric>

DISPlay[:WINDow]:TRACe{1-17}:X[:SCALe]:RLEVel <numeric>

When the measurement format is set to the complex plane, this command, sets the value of the x-axis reference line, moving the measurement trace correspondingly.

(REFERENCE X VALUE under (Scale Ref))

Node	Description
:TRACe1	Data trace
:TRACe{2-17}	Memory traces

Parameter	Range	Unit
<numeric>	-5.0×10^8 to 5.0×10^8	U

Query response is <numeric>.

:X:SPACing {LINear|LOGarithmic|OBASe}

DISPlay[:WINDow]:TRACe{1-21}:X:SPACing {LINear|LOGarithmic|OBASe}

This command has the following three functions:

1. Selects the sweep type of Data & Memory trace, LINear or LOGarithmic. This function works with the SENSE:SWEep:SPACing, SOURCE{1|2}:SWEep:SPACing.

(SWP TYPE: LIN , LOG under (Sweep))

2. Selects the scale of the user trace, LINear or LOGarithmic. (X-AXIS: LIN , LOG under (Format))

3. Selects the order base (OBASe) on frequency base list sweep.

(LIST DISP: ODER BASE under (Sweep))

Node	Description
:TRACe{1-17}	Data trace(:TRACe1) and memory trace(:TRACe{2-17}). The command works the same whatever :TRACe{1-17} is selected.
:TRACe{18-21}	User traces.

Parameter	Description
LINear	Linear sweep mode (data & memory trace) when :TRACe{1-17} is selected. Linear X axis (user trace) when :TRACe{18-21} is selected.
LOGarithmic	Logarithmic sweep mode (data & memory trace) when :TRACe{1-17} is selected. Logarithmic X axis (user trace) when :TRACe{18-21} is selected.
OBASe	Order base list sweep (cannot be used with :TRACe{18-21}).

Query response is {LIN|LOG|OBAS}.

Related Command SENSE:SWEep:SPACing
SOURCE{1|2}:SWEep:SPACing

Example To set sweep type to linear in frequency sweep,

```
OUTPUT @Hp4291;"SENS:SWE:SPAC LIN"
OUTPUT @Hp4291;"DISP:TRAC:X:SPAC LIN"
```

To set sweep type to logarithmic in OSC level sweep,

```
OUTPUT @Hp4291;"SOUR:SWE:SPAC LOG"
OUTPUT @Hp4291;"DISP:TRAC:X:SPAC LOG"
```

:X:UNIT <string>

DISPlay[:WINDow]:TRACe{18-21}:X:UNIT <string>

Defines the X-axis unit of the graticule for the user trace. (X UNIT LABEL under (Display))

Node	Description
:TRACe{18-21}	User traces

Parameter	Description
<string>	Unit to display, up to 4 characters

Query response is <string>.

DISPlay[:WINDow]:TRACe{1-21}**:Y[:SCALE]:AUTO ONCE**

DISPlay[:WINDow]:TRACe{1-21}:Y[:SCALE]:AUTO ONCE

Adjusts the display scale and brings the trace data, defined by the TRACe{1-21} node, in view on the display. (**AUTO SCALE** under **(Scale Ref)**; No query)

Node	Description
:TRACe1	Data trace
:TRACe{2-17}	Memory traces
:TRACe{18-21}	User traces

:Y[:SCALE]:BOTTom <numeric>

DISPlay[:WINDow]:TRACe{1-21}:Y[:SCALE]:BOTTom <numeric>

Defines the bottom border of the display and adjusts the scale value. (**BOTTOM VALUE** under **(Scale Ref)**)

Node	Description
:TRACe1	Data trace
:TRACe{2-17}	Memory traces
:TRACe{18-21}	User traces

Parameter	Range	Unit
<numeric>	-500×10 ⁶ to 500×10 ⁶	y-axis unit

Query response is <numeric>.

:Y[:SCALE]:COUPlE {OFF|ON|0|1}

DISPlay[:WINDow]:TRACe{1-17}:Y[:SCALE]:COUPlE {OFF|ON|0|1}

Couples or uncouples the DATA and MEMORY traces to be scaled. This command works the same whatever :TRACe{1-17} is selected. (**D&M SCALE []** under **(Scale Ref)**)

Parameter	Description
OFF or 0	Uncouples the "DATA" and "MEMORY" traces.
ON or 1	Couples the "DATA" and "MEMORY" traces.

Query response is {0|1}.

Example

```

OUTPUT @Hp4291;"DISP:TRAC1:Y:COUP ON"
OUTPUT @Hp4291;"DISP:TRAC1:Y:COUP?"
ENTER @Hp4291;A

```

:Y[:SCALE]:PDIVision <numeric>

DISPlay[:WINDow]:TRACe{1-17}:Y[:SCALE]:PDIVision <numeric>

Sets the response value scale per graticule trace. (SCALE/DIV under **Scale Ref**)

Node	Description
:TRACe1	Data trace
:TRACe{2-17}	Memory traces

Parameter	Range	Unit
<numeric>	-1×10^{-15} to 1×10^{-15}	y-axis unit

Query response is <numeric>.

Example

```
OUTPUT @Hp4291;"DISP:TRAC1:Y:PDIV 1"
OUTPUT @Hp4291;"DISP:TRAC1:Y:PDIV?"
ENTER @Hp4291;A
```

:Y[:SCALE]:RLEVel {<numeric>|MARKer}

DISPlay[:WINDow]:TRACe{1-17}:Y[:SCALE]:RLEVel {<numeric>|MARKer}

Sets the value of the reference line, moving the measurement trace correspondingly.

(REFERENCE VALUE, REFERENCE Y VALUE under **Scale Ref**)

Node	Description
:TRACe1	Data trace
:TRACe{2-17}	Memory traces

Parameter	Range	Unit
<numeric>	-500×10^6 to 500×10^6	y-axis unit

Query response is <numeric>.

Parameter	Description
MARKer	Makes the reference value equal to the marker's absolute value (regardless of the Δ marker value). (MKR→REFERENCE under Scale Ref and Marker→)

Example

```
OUTPUT @Hp4291;"DISP:TRAC1:Y:RLEV 0"
OUTPUT @Hp4291;"DISP:TRAC1:Y:RLEV?"
ENTER @Hp4291;A
```

:Y[:SCALE]:RPOsition <numeric>

DISPlay[:WINDow]:TRACe{1-17}:Y[:SCALE]:RPOsition <numeric>

Sets the position of the reference line on the graticule of a Cartesian display.

REFERENCE POSITION under (Scale Ref)

Node	Description	
:TRACe1	Data trace	
:TRACe{2-17}	Memory traces	

Parameter	Range	Unit
<numeric>	0 to 100	%

Query response is <numeric>.

Example OUTPUT @Hp4291;"DISP:TRAC1:Y:RPOS 0"
 OUTPUT @Hp4291;"DISP:TRAC1:Y:RPOS?"
 ENTER @Hp4291;A

:Y[:SCALE]:TOP <numeric>

DISPlay[:WINDow]:TRACe{1-21}:Y[:SCALE]:TOP <numeric>

Defines the top border of the display and adjusts the scale value. (TOP VALUE under

(Scale Ref))

Node	Description	
:TRACe1	Data trace	
:TRACe{2-17}	Memory traces	
:TRACe{18-21}	User traces	

Parameter	Range	Unit
<numeric>	-1×10^9 to 1×10^9	y-axis unit

Query response is <numeric>.

:Y:SPACing {LOGarithmic|LINear}

DISPlay[:WINDow]:TRACe{1-21}:Y:SPACing {LOGarithmic|LINear}

Sets the Y-axis to LINear or LOGarithmic. (LIN Y-AXIS, LOG Y-AXIS,

Y-AXIS: LIN, LOG under (Format))

Node	Description
:TRACe{1-17}	Data & memory trace. (This command works the same whatever TRACe{1-17} is selected.)
:TRACe{18-21}	User traces.

Parameter	Description
LOGarithmic	Logarithmic scale (LOG Y-AXIS ;for data and memory trace, Y SPACING [LOG] ;for user trace)
LINEar	Linear scale (LIN Y-AXIS ;for data and memory trace, Y SPACING [LIN] ;for user trace)

Query response is {LOG|LIN}.

Related Command DISPlay[:WINDow]:TRACe:GRATICule:FORMat

Example To select the logarithmic scale,

```
OUTPUT @Hp4291;"DISP:TRAC:GRAT:FORM RECT"
OUTPUT @Hp4291;"DISP:TRAC1:Y:SPAC LOG"
```

:Y:UNIT <string>

DISPlay[:WINDow]:TRACe{18-21}:Y:UNIT <string>

Defines the Y-axis unit of the graticule for the user trace. (Y UNIT LABEL under

Display)

Node	Description
:TRACe{18-21}	User traces

Parameter	Description
<string>	Unit to display, up to 4 characters

Query response is <string>.

DISSMEMO {OFF|ON|0|1}, DUAC {OFF|ON|0|1}

(*Simple Command*) See “Simple Commands” later in this chapter.

DPI <numeric>

(*Simple Command*) See “Simple Commands” later in this chapter.

DUAM {IMPH|IRIM|APPH|ARIM|LSQ|LPQ|CSD|CPD|DMPH|DRIM|PMPH|PRIM}

(*Simple Command*) See “Simple Commands” later in this chapter.

FMT {LINY|LOGY|POL|SMIT|ADM|COMP}

(*Simple Command*) See “Simple Commands” later in this chapter.

FORMat Subsystem

FORMat[:DATA] {ASCIi|REAL,32|REAL,64|PACKed,32}

FORMat[:DATA] {ASCIi|REAL,32|REAL,64|PACKed,32}

Sets the format to transfer data via GPIB. See the *Programming Guide* for more information.

Parameter	Description
ASCIi	ASCII transfer format
REAL,32	IEEE 32-bit floating point format
REAL,64	IEEE 64-bit floating point format
PACKed,32	DOS format

Query Response is {ASC|REAL,32|REAL,64|PACK,32}.

At *RST, ASCIi is selected.

Related Command The FORMat[:DATA] command selects the format of the data sent by the following commands:

```
DATA[:DATA]
DATA[:DATA]?
DATA[:DATA]:VALue?
TRACe[:DATA]
TRACe[:DATA]:VALue?
```

Example

```
OUTPUT @Hp4291;"FORM REAL,32"  
OUTPUT @Hp4291;"FORM?"  
ENTER @Hp4291;A$
```

FREO

(*Simple Command*) See “Simple Commands” later in this chapter.

FORMFEED {OFF|ON|0|1}

(*Simple Command*) See “Simple Commands” later in this chapter.

HCOPY Subsystem

The HCOpy subsystem controls the hard copy functions (printing).

HCOpy

HCOpy

See “HCOpy[:IMMediate]”.

HCOpy:ABORt

HCOpy:ABORt

Aborts print in progress. (COPY ABORT under Copy); No query)

Example OUTPUT @Hp4291;"HCOP:ABOR"

HCOpy:DEFault

HCOpy:DEFault

Sets the printing parameters to their default values. (DEFAULT SETUP under Copy); No query)

The table below lists the default values.

Command	Default value	Unit
DPI	75	dpi
FORMFEED	ON	
LANDSCAPE	OFF	
LMARG	1.0	inch
TMARG	1.0	inch
SKEY	OFF	

Example OUTPUT @Hp4291;"HCOP:DEF"

HCOPY:DEVIce:CMAP:COLor {FIXed|VARIable}

HCOPY:DEVIce:CMAP:COLor {FIXed|VARIable}

- Sets the default colors for printing a hard copy. (PRINT COLOR [FIXED] under (Copy))
- Sets the colors used for printing a hard copy as close as possible to the display colors. (PRINT COLOR [VARIABLE] under (Copy))

Parameter	Description
FIXed	Fixed colors (default colors)
VARIable	Variable colors (colors similar to the display)

Query Response is {FIX|VAR}.

Example OUTPUT @Hp4291;"HCOP:DEV:CMAP:COL FIX"
 OUTPUT @Hp4291;"HCOP:DEV:CMAP:COL?"
 ENTER @Hp4291;A\$

HCOPY:DEVIce:COLor {OFF|ON|0|1}

HCOPY:DEVIce:COLor {OFF|ON|0|1}

Sets the print command to color printing. (COLOR under (Copy))

Parameter	Description
OFF or 0	Single-color printing
ON or 1	Color printing

Query Response is {0|1}.

Example OUTPUT @Hp4291;"HCOP:DEV:COL ON"

HCOPY:DEVIce:DPI <numeric>

HCOPY:DEVIce:DPI <numeric>

Specifies the resolution of a printer used for printing by dpi. (DPI under (Copy))

Parameter	Range	Unit
<numeric>	75 to 600	DPI(Dot Per Inch)

Query response is <numeric>.

Example OUTPUT @Hp4291;"HCOP:DEV:DPI 100"
 OUTPUT @Hp4291;"HCOP:DEV:DPI?"
 ENTER @Hp4291;A

HCOPY:DEVIce:FORMFeed {OFF|ON|0|1}

HCOPY:DEVIce:FORMFeed {OFF|ON|0|1}

Sets the printer ON or OFF for delivering printed paper each time printing an entire screen is finished. When the paper orientation is set to Landscape, the setting by this FORMFEED command will not take effect and the printer delivers printed paper screen by screen.

(FORMFEED ON off under **Copy**)

Parameter	Description
OFF or 0	Does not deliver printed paper
ON or 1	Delivers printed paper

Query response is {0|1}.

Example OUTPUT @Hp4291;"HCOP:DEV:FORMF ON"

OUTPUT @Hp4291;"HCOP:DEV:FORMF?"
ENTER @Hp4291;A

HCOPY:DEVIce:LANDScape {OFF|ON|0|1}

HCOPY:DEVIce:LANDScape {OFF|ON|0|1}

Sets the orientation of paper landscape or not, using ON or OFF. This setting takes effect for printers which support printing paper placed in the landscape orientation. Setting the paper orientation mode will invalidate the setting by the FORMFEED command.

(ORIENT [PORTRAIT] under **Copy**)

Parameter	Description
OFF or 0	The orientation of paper is not set to Landscape. (Thus, Portrait)
ON or 1	The orientation of paper is set to Landscape.

Query response is {0|1}.

Example OUTPUT @Hp4291;"HCOP:DEV:LANDS ON"

OUTPUT @Hp4291;"HCOP:DEV:LANDS?"
ENTER @Hp4291;A

HCOPY:DEVIce:LEFTMarg <numeric>

HCOPY:DEVIce:LEFTMarg <numeric>

Specifies the left margin of printing by inch. (LFT MARGIN under **Copy**)

Parameter	Range	Unit
<numeric>	0 to 5	inch

Query response is <numeric>.

Example OUTPUT @Hp4291;"HCOP:DEV:LEFTM 2"

OUTPUT @Hp4291;"HCOP:DEV:LEFTM?"
ENTER @Hp4291;A

HCOPY:DEVIce:SKEY {OFF|ON|0|1}

HCOPY:DEVIce:SKEY {OFF|ON|0|1}

Sets printing the softkeys displayed in the screen ON or OFF. (COPY SKEY on OFF under **Copy**)

Parameter	Description
OFF or 0	Does not print the soft keys
ON or 1	Print the soft keys

Query response is {0|1}.

Example OUTPUT @Hp4291;"HCOP:DEV:SKEY ON"

 OUTPUT @Hp4291;"HCOP:DEV:SKEY?"
 ENTER @Hp4291;A

HCOPY:DEVIce:TOPMarg <numeric>

HCOPY:DEVIce:TOPMarg <numeric>

Specifies the top margin of printing by inch. (TOP MARGIN under **Copy**)

Parameter	Range	Unit
<numeric>	0 to 5	inch

Query response is <numeric>.

Example OUTPUT @Hp4291;"HCOP:DEV:TOPM 3"

 OUTPUT @Hp4291;"HCOP:DEV:TOPM?"
 ENTER @Hp4291;A

HCOPY[:IMMediate]

HCOPY[:IMMediate]

Executes printing. (PRINT [] under **Copy**);No query)

Example OUTPUT @Hp4291;"HCOP"

HCOPY:ITEM Subsystem

This subsystem is used to select the display hard copy object.

:TDSTamp:STATe {OFF|ON|0|1}

HCOPY:ITEM:TDSTamp:STATe {OFF|ON|0|1}

Turns printing time and date (the time stamp function) ON or OFF.

(COPY TIME ON off under **Copy**)

Parameter	Description
OFF or 0	Time stamp function OFF.
ON or 1	Time stamp function ON.

Query Response is {0|1}.

Example

```
OUTPUT @Hp4291;"HCOP:ITEM:TDST:STAT ON"
```

```
OUTPUT @Hp4291;"HCOP:ITEM:TDST:STAT?"
```

```
ENTER @Hp4291;A
```

HOLD

(*Simple Command*) See “Simple Commands” later in this chapter.

INITiate Subsystem

The INITiate subsystem is used to control the initiation of the trigger system. To trigger a measurement, the trigger sequence should be in the waiting-for-trigger state (that is, the trigger system should be initiated). For more information about triggering, see the *Programming Manual*.

Related Command `ABORt`
 `TRIGger:EVENT TYPE {POINT|SWEep}`
 `TRIGger:SLOPe {POSitive|NEGative}`
 `TRIGger:SOURce {BUS|EXTernal|INTernal|MANual}`

INITiate:CONTinuous {OFF|ON|0|1}

INITiate:CONTinuous {OFF|ON|0|1}

Initiate the trigger system continuously.

Parameter	Description
OFF or 0	After measurement, the trigger sequence moves to the idle state.
ON or 1	After measurement, the trigger sequence moves to the wait-for-trigger state (continuously initiated).

Query Response is {0|1}.

Example To abort the measurement and hold the sweep immediately,

```
OUTPUT @Hp4291;"INIT:CONT OFF"
OUTPUT @Hp4291;"ABOR"
```

Query statement,

```
OUTPUT @Hp4291;"INIT:CONT?"
ENTER @Hp4291;A
```

INITiate[:IMMediate]

INITiate[:IMMediate]

Initiate trigger system. (`SING`, `NUMBER OF GROUP` under `(Trigger)`;No query)

This command execution can cause an error if the trigger sequence is not in the idle state. To place it in the idle state, execute the `ABORt` command.

Example To trigger a measurement,

```
OUTPUT @Hp4291;"TRIG:SOUR INT"
OUTPUT @Hp4291;"INIT:CONT OFF"
OUTPUT @Hp4291;"ABOR"
OUTPUT @Hp4291;"SENS:SWE:COUN 1"
OUTPUT @Hp4291;"INIT"
```

After measurement, the analyzer goes to trigger hold.

INITiate[:IMMEDIATE]

INITiate[:IMMEDIATE]:AGain:ALL

INITiate[:IMMEDIATE]:AGain:ALL

Aborts the sweep in progress to exit to the idle state and initiates the trigger system again.

(MEASURE RESTART under Trigger); No query)

INSTrument Subsystem

The INSTrument subsystem controls the channel 1 and 2 displays and measurement.

INSTrument {CH1|CH2}

INSTrument {CH1|CH2}

See “INSTrument[:SElect] {CH1|CH2}”

INSTrument:COUple {ALL|NONE}

INSTrument:COUple {ALL|NONE}

Sets the channel coupling of stimulus values. (COUPLED CH ON off under Sweep)

Parameter	Description
ALL	Channel coupling ON.
NONE	Channel coupling OFF.

Query response is {ALL|NONE}. (<NR1> format)

```
Example          OUTPUT @Hp4291;"INST:COUP ALL"
                  OUTPUT @Hp4291;"INST:COUP?"
                  ENTER @Hp4291;A$
```

INSTrument:NSElect {1|2}

INSTrument:NSElect {1|2}

Selects the active channel. (Chan 1) or (Chan 2)

This command serves the same purpose as the INSTrument[:SElect] {CH1|CH2} command.

Parameter	Description
1	Selects channel 1 as the active channel. (Chan 1)
2	Select s channel 2 as the active channel. (Chan 2)

Query Response is {1|2}.

Related Command INSTrument:STATe {OFF|ON|0|1}

```
Example          OUTPUT @Hp4291;"INST:NSEL 1"
                  OUTPUT @Hp4291;"INST:STAT ON"
                  OUTPUT @Hp4291;"INST:NSEL?"
                  ENTER @Hp4291;A
```

INSTRument[:SElect] {CH1|CH2}

INSTRument[:SElect] {CH1|CH2}

Selects the active channel. (**Chan 1**) or (**Chan 2**)

This command serves the same purpose as the INSTRument:NSElect {1|2} command.

Parameter	Description
CH1	Selects channel 1 as the active channel. (Chan 1)
CH2	Selects channel 2 as the active channel. (Chan 2)

Query Response is {CH1|CH2}.

Related Command INSTRument:STATe {OFF|ON|0|1}

Example

```
OUTPUT @Hp4291;"INST CH1"
OUTPUT @Hp4291;"INST:STAT ON"

OUTPUT @Hp4291;"INST?"
ENTER @Hp4291;A$
```

INSTRument:STATe {OFF|ON|0|1}

INSTRument:STATe {OFF|ON|0|1}

Displays the selected channel.

Parameter	Description
OFF or 0	Does not display the selected channel.
ON or 1	Displays the selected channel.

Query response is {0|1}.

Related Command INSTRument[:SElect] {CH1|CH2}
INSTRument:NSElect {1|2}

Example To display channel 1 and channel 2 at same time (dual channel ON),

```
OUTPUT @Hp4291;"INST CH1"
OUTPUT @Hp4291;"INST:STAT ON"
OUTPUT @Hp4291;"INST CH2"
OUTPUT @Hp4291;"INST:STAT ON"
```

To display channel 1 only (dual channel off),

```
OUTPUT @Hp4291;"INST CH2"
OUTPUT @Hp4291;"INST:STAT OFF"
OUTPUT @Hp4291;"INST CH1"
OUTPUT @Hp4291;"INST:STAT ON"
```

LANDSCAPE {OFF|ON|0|1}, **LISD** {FBAS|OBAS}, **LMARG**
<numeric>

(*Simple Command*) See “Simple Commands” later in this chapter.

MARDCENT, **MARDSPAN**, **MARK** {OFF|ON|0|1}, **MARKCENT**,
MARKREF **MARKSTAR**, **MARKSTOP**, **MARZ**

(*Simple Command*) See “Simple Commands” later in this chapter.

MATH {DATA|DMNM|DPLM|DDVM|DMLM}

(*Simple Command*) See “Simple Commands” later in this chapter.

MEAS {IMAG|IPH|IRE|IIM|AMAG|APH|ARE|AIM|RCM|RCPH|
RCR|RCIM|DCM|DCPH|DCR|DCIM|PHMA|PPH|PRE|PIM|CP|CS|LP|
LS|D|Q|RP|RS}

(*Simple Command*) See “Simple Commands” later in this chapter.

MMEMory Subsystem

The MMEMory subsystem controls the save/recall function.

MMEMory:CDIRectory [<string>]

MMEMory:CDIRectory [<string>]

Changes the current directory of a DOS format disk. (CHANGE DIRECTORY under **Save**); No query)

Parameter	Description
<string>	Directory path.

Example OUTPUT @Hp4291;"MMEM:CDIR ""..""

MMEMory:COpy {<string(s)>,<string(m_s)>,<string(d)>,<string(m_d)> }

MMEMory:COpy {<string(s)>,<string(m_s)>,<string(d)>,<string(m_d)> }

Copies files. (**COPY FILE** under **Save**); No query

Parameter	Description
<string(s)>	Source file name.
<string(m_s)>	Source mass storage name ("DISK" or "MEMORY"). ¹
<string(d)>	Destination file name, up to 8 characters. In addition to the file name, the extension may be sent. Add appropriate extension if the destination mass storage is "MEMORY".
<string(m_d)>	Destination mass storage name. ("DISK" or "MEMORY")

¹ "DISK" for the built-in floppy disk drive; "MEMORY" for the memory disk.

Example

```
OUTPUT @Hp4291;"MMEM:COpy ""DAT1.TXT"",""MEMORY"",""DAT1.TXT"",""DISK"""
```

Note

You cannot copy the file if the source mass storage format (DOS or LIF) is different from that of mass storage.

MMEMory:CREate:DIRectory <string>

MMEMory:CREate:DIRectory <string>

Creates a new directory in a DOS format disk. (**CREATE DIRECTORY** under **Save**); No query

Parameter	Description
<string>	Directory name, up to 8 characters. In addition to the directory name, the extension (up to 3 character) may be sent.

Example

```
OUTPUT @Hp4291;"MMEM:CRE:DIR ""DATA"""
```

```
OUTPUT @Hp4291;"MMEM:CRE:DIR ""INDUCTOR.DIR"""
```

MMEMory:DELeTe <string(file_name)>[,<string(m)>]

MMEMory:DELeTe <string(file_name)>[,<string(m)>]

Removes the file. (**PURGE FILE** under **SAVE**); No query

Parameter	Description
<string(file_name)>	File name with extension.
<string(m)>	"DISK" for the floppy disk drive. "MEMORY" for the memory disk.

Example

```
OUTPUT @Hp4291;"MMEM:DEL ""TEST_S"""
```

MMEMory:INITialize <string>,{LIF|DOS}

MMEMory:INITialize <string>,{LIF|DOS}

Initializes the disk in the floppy disk drive or the memory disk. (INITIALIZE under **Save**); No query)

Parameter	Description
<string>	"DISK" for the floppy disk drive. "MEMORY" for the memory disk.
LIF	LIF format.
DOS	DOS format.

Example OUTPUT @Hp4291;"MMEM:INIT ""DISK"",DOS"

MMEMory:LOAD Subsystem

This subsystem is used to load data from the floppy disk drive or the memory disk.

:STATE <string(file_name)>[,<string(m)>]

MMEMory:LOAD:STATe <string(file_name)>[,<string(m)>]

Loads the instrument states. (file name under **Recall**); No query)

Parameter	Description
<string(file_name)>	File name with extension
<string(m)>	"DISK" for the floppy disk drive. "MEMORY" for the memory disk.

Example OUTPUT @Hp4291;"MMEM:LOAD:STAT ""TEST_S"""

:TRACe SEL,<string(file_name)>[,<string(m)>]

MMEMory:LOAD:TRACe SEL,<string(file_name)>[,<string(m)>]

Loads data. (file name under **Recall**); No query)

Parameter	Description
<string(file_name)>	File name with extension.
<string(m)>	"DISK" for the floppy disk drive. "MEMORY" for the memory disk.

Example OUTPUT @Hp4291;"MMEM:LOAD:TRAC SEL, ""TEST_D"""

:ITEM:TRACe:DELeTe {CCO|DATA|DTR|UTR|MEM|MTR|RAW}

MMEMory:STORe:ITEM:TRACe:DELeTe {CCO|DATA|DTR|GTR|MEM|MTR|RAW}

Selects the item that is not saved. (DEFINE SAVE DATA under **Save**); No query)

This command deletes the item which was set to be saved by the MMEMory:STORe:ITEM:TRACe:SELeCt command. The deleted items will not be saved.

Parameter	Description
CCO	calibration/compensation coefficient (CAL on OFF)
DATA	data (DATA on OFF)
DTR	data trace (DATA TRACE on OFF)
UTR	user trace (USER TRACE on OFF)
MEM	memory (MEM on OFF)
MTR	memory trace (MEM TRACE on OFF)
RAW	raw data (RAW on OFF)

Related Command MMEMory:STORe:ITEM:TRACe:SELeCt
 MMEMory:STORe:TRACe
 MMEMory:STORe:DINTerchange:TRACe

Example OUTPUT @Hp4291;"MMEM:STOR:ITEM:TRAC:DEL CCO"

:ITEM:TRACe:SELeCt {CCO|DATA|DTR|UTR|MEM|MTR|RAW}

MMEMory:STORe:ITEM:TRACe:SELeCt {CCO|DATA|DTR|GTR|MEM|MTR|RAW}

Selects the items to be saved. (DEFINE SAVE DATA under **Save**); No query)

Parameter	Description
CCO	calibration/compensation coefficient (CAL on OFF)
DATA	data (DATA ON off)
DTR	data trace (DATA TRACE ON off)
UTR	user trace (USER TRACE ON off)
MEM	memory (MEM ON off)
MTR	memory trace (MEM TRACE ON off)
RAW	raw data (RAW ON off)

Related Command MMEMory:STORe:ITEM:TRACe:DELeTe
 MMEMory:STORe:TRACe
 MMEMory:STORe:DINTerchange:TRACe

Example OUTPUT @Hp4291;"MMEM:STOR:ITEM:TRAC:SEL CCO"

NUMG <numeric>

(*Simple Command*) See “Simple Commands” later in this chapter.

PROGram Subsystem

The PROGram subsystem controls the HP instrument BASIC furnished in the analyzer from an external controller. The PROGram subsystem is used from an external controller.

The PROGram subsystem is organized by,

- PROGram:CATalog? command
- PROGram[:SElected] subsystem
- PROGram:EXPLicit subsystem

The PROGram[:SElected] subsystem and PROGram:EXPLicit subsystem are structured the same and have same function. This manual provides complete information on the PROGram[:SElected]. For PROGram:EXPLicit subsystem, it refers to the corresponding description in the PROGram[:SElected] subsystem.

PROGram:CATalog?

PROGram:CATalog?

Returns the defined program name (defined by PROGram[:SElected]:NAME command). This command performs no practical function for the analyzer. (Query only)

Query response is <string>.

PROGram[:SElected] Subsystem

This subsystem is used to control the HP instrument BASIC from an external controller.

:DEFine <block>

PROGram[:SElected]:DEFine <block>

Creates and downloads programs from an external controller to HP instrument BASIC. The PROGram[:SElected]:DEFine? query uploads HP instrument BASIC programs to an external controller. This command can be used from an external controller only.

Parameter	Description
<block>	program

PROGram[:SElected]

The <block> must be arbitrary block program data containing the lines of program code. The first line of <block> must be a header that shows the program size. There are two formats for the header as follows:

```
#0           Allows the OUTPUT statement to send program lines until END is specified in the OUTPUT
            statement.
#NMM... M   Specifies the program size.
            N specifies the number of digits that define the program size.
            M... M is program size in bytes (N digits).
```

Each line of the program must be separated by <New Line>. When the size of the <block> exceeds the amount of available memory in the analyzer, the program lines are saved up to the point of memory overflow.

In response to the DEFine query, the <block> is uploaded as the definite length arbitrary block response data. The program size is returned in the first line as the header, then the program lines are returned. The program must be either in the paused or stopped state for the program to be uploaded.

Example

To create a program in the HP instrument BASIC editor,

```
OUTPUT @Hp4291;"PROG:DEF #0"
OUTPUT @Hp4291;"10 PRINT ""HELLO!""
OUTPUT @Hp4291;"20 END"
OUTPUT @Hp4291;" ",END
```

To upload the program to the external controller,

```
DIM A$ [100000]
OUTPUT @Hp4291;"PROG:DEF?"
ENTER @Hp4291 USING "%,2A";HEAD$
B=VAL(HEAD$[2])
FOR I=1 TO B
  ENTER @Hp4291 USING "%,A";HEAD$
NEXT I
ENTER @Hp4291 USING "-K";A$
```

:DELete[:SElected]

PROGram[:SElected]:DELete[:SElected]

Deletes the HP instrument BASIC program in the analyzer. This command can be used from an external controller only. (No query)

:DELete:ALL

PROGram[:SElected]:DELete:ALL

Deletes the HP instrument BASIC program in the analyzer. This command can be used from an external controller only. (No query)

This command works the same as the PROGram[:SElected]:DELete[:SElected] command.

:EXECute <string>

PROGram[:SElected]:EXECute <string>

Executes the program command. The program must be either paused or stopped before the EXECute command is executed. This command can be used from an external controller only. (No query)

Parameter	Description
<string>	Legal program command

Example¹

To execute the program command, STEP

```
OUTPUT @Hp4291;"PROG:EXEC 'STEP'"
OUTPUT @Hp4291;"PROG:EXEC ""STEP"""
```

To execute the program command, MSI ":MEMORY"

```
OUTPUT @Hp4291;"PROG:EXEC 'MSI ':MEMORY'"
OUTPUT @Hp4291;"PROG:EXEC ""MSI """:MEMORY"""""
```

To execute the program command, GET "";File\$;""

```
OUTPUT @Hp4291;"PROG:EXEC 'GET ''";File$;""'"
OUTPUT @Hp4291;"PROG:EXEC ""GET """";";File$;"""""""
```

¹ Each example contains two cases. One uses the single quote, the other uses the double quote.

:MALLocate {<numeric>|DEfault}

PROG:SElected].MALLocate {<numeric>|DEfault}

Performs no practical function for the analyzer.

:NAME <string>

PROG:SElected].NAME <string>

Defines the program name. This command performs no practical function for the analyzer.

Parameter	Description
<string>	Program name (Default is "PROG".)

Query response is <string>.

:NUMBER <var>,<numeric1>,<numeric2>,...,<numeric n>

PROG:SElected].NUMBER <var>,<numeric1>,<numeric2>,...,<numeric n>

Sets or queries the contents of numeric program variables and arrays in the HP instrument BASIC of the analyzer. This command performs a practical function from the external controller only.

Parameter	Description
<var>	Name of an existing variable in the program (either character data or string data)
<numeric1>,...,<numeric n>	Variable value (n is the number of the array.)

Query response is <numeric1>,<numeric2>,...,<numeric n>.

Example

```
OUTPUT @Hp4291;"PROG:NUMB A,1"
OUTPUT @Hp4291;"PROG:NUMB? A"
ENTER @Hp4291;B
```

PROGram[:SElected]**:STATe** {**RUN**|**PAUSE**|**STOP**|**CONTInue**}PROGram[:SElected].STATe {**RUN**|**PAUSE**|**STOP**|**CONTInue**}

Sets or queries the state of the program in the analyzer. The table below defines the affect of setting the state to the specified state from each of the possible current states. This command performs a practical function from the external controller only.

Parameter	Current State		
	RUN	PAUSE	STOP
RUN	error (-221)	RUN	RUN
CONT	error (-221)	RUN	error (-221)
PAUSE	PAUSE	PAUSE	STOP
STOP	STOP	STOP	STOP

Query response is {**RUN**|**PAUS**|**STOP**|**CONT**}.

Example **OUTPUT @Hp4291;"PROG:STAT STOP"**
 OUTPUT @Hp4291;"PROG:STAT?"
 ENTER @Hp4291;A\$

:STRing <var>,<string1>,<string2>,...,<string n>

PROGram[:SElected].STRing <var>,<string1>,<string2>,...,<string n>

Sets or queries the contents of the string program variables and arrays in the HP instrument BASIC of the analyzer. If a string value is too long, it is truncated when stored in the program's variable. This command performs a practical function from the external controller only.

Parameter	Description
<var>	Name of an existing variable in the program (either character data or string data).
<string1>, ... ,<string n>	Variable value (n is number of the element in the array).

Query response is <string1>,<string2>, ... ,<string n>.

Example **OUTPUT @Hp4291;"PROG:STR A,""HELLO"""**
 OUTPUT @Hp4291;"PROG:STR? A"
 ENTER @Hp4291;B\$

:WAIT

PROGram[:SElected].WAIT

Causes no further commands or queries to be executed until the program in the analyzer exits from the RUN state (either stopped or paused). This command performs a practical function from the external controller only.

Query Response is 1.
 (1 is returned when the program is either stopped or paused.)

Example To wait until the program in the analyzer completes,

```
OUTPUT @Hp4291;"PROG:WAIT"
PRINT "The program complete"

OUTPUT @Hp4291;"PROG:WAIT?"
ENTER @Hp4291;A
```

PROG:EXPLICIT Subsystem

The following commands under the :EXPLICIT node perform the specified functions in the same manner as the corresponding commands under the :SELETED node. The :EXPLICIT commands are included in the analyzer's GPIB commands to maintain compatibility with other SCPI instruments. Therefore, you can use either the :EXPLICIT or the :SELETED commands for the analyzer. However, you should select one set and use it consistently to avoid confusion.

:DEFine "PROG", <block>

PROG:EXPLICIT:DEFine "PROG", <block>

See "PROG:SELETED:DEFine <block>"

:DELeTe "PROG"

PROG:EXPLICIT:DELeTe "PROG"

See "PROG:SELETED:DELeTe[:SELETED]"

:EXECute "PROG", <string>

PROG:EXPLICIT:EXECute "PROG", <string>

See "PROG:SELETED:EXECute <string>"

:MALLocate "PROG", {<numeric>|DEFault}

PROG:EXPLICIT:MALLocate "PROG", {<numeric>|DEFault}

See "PROG:SELETED:MALLocate {<numeric>|DEFault}"

:NAME "PROG", <string>

PROG:EXPLICIT:NAME "PROG", <string>

See "PROG:SELETED:NAME <string>"

:NUMBER “PROG”, <var>, <numeric1>, <numeric2>, ... , <numeric n>

PROG:EXPLICIT:NUMBER "PROG", <var>, <numeric1>, <numeric2>, ..., <numeric n>

See “PROGRAM[:SELECTED]:NUMBER <var>, <numeric1>, <numeric2>, ... , <numeric n>”

:STATE “PROG”, {RUN|PAUSE|STOP|CONTINUE}

PROG:EXPLICIT:STATE "PROG", {RUN|PAUSE|STOP|CONTINUE}

See “PROGRAM[:SELECTED]:STATE {RUN|PAUSE|STOP|CONTINUE}”

:STRING “PROG”, <var>, <string1>, <string2>, ... , <string n>

PROG:EXPLICIT:STRING "PROG", <var>, <string1>, <string 2>, ..., <string n>

See “PROGRAM[:SELECTED]:STRING <var>, <string1>, <string2>, ... , <string n>”

:WAIT “PROG”

PROG:EXPLICIT:WAIT "PROG"

See “PROGRAM[:SELECTED]:WAIT”

PEAKCENT, POIDTIME <numeric>, **PRSOFT** {OFF|ON|0|1},
SELM <numeric>

(*Simple Command*) See "Simple Commands" later in this chapter.

SAVDSTAC <string>, **SAVDTIF** <string>, **STOD**{DISK|MEMO},
STORMDISK

(*Simple Command*) See "Simple Commands" later in this chapter.

SENSe Subsystem

SENSe:AVERage1:COUNT <numeric>

SENSe:AVERage1:COUNt <numeric>

Sets the point averaging factor for the active channel. (POINT AVG FACTOR under **Bw/Avg**)

Parameter	Range	Unit
<numeric>	1 to 999	—

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:AVER1:COUN 2"
 OUTPUT @Hp4291;"SENS:AVER1:COUN?"
 ENTER @Hp4291;A

SENSe:AVERage1[:STATe] {OFF|ON|0|1}

SENSe:AVERage1[:STATe] {OFF|ON|0|1}

Sets the point averaging of the active channel ON or OFF. (POINT AVG ON off under **Bw/Avg**)

Parameter	Description
OFF or 0	Point averaging function OFF.
ON or 1	Point averaging function ON.

Query response is {0|1}.

Example OUTPUT @Hp4291;"SENS:AVER ON"
 OUTPUT @Hp4291;"SENS:AVER?"
 ENTER @Hp4291;A

SENSE:AVERage2:CLEar

SENSE:AVERage2:CLEar

Resets the averaging and restarts the sweep count at 1 at the beginning of the next sweep on the active channel. (AVERAGING RESTART under (Bw/Avg); No query)

Example OUTPUT @Hp4291;"SENS:AVER2:CLE"

SENSE:AVERage2:COUNT <numeric>

SENSE:AVERage2:COUNT <numeric>

Sets the sweep averaging factor for the active channel. (AVERAGING FACTOR under (Bw/Avg))

Parameter	Range	Unit
<numeric>	1 to 999	—

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:AVER2:COUNT 16"
 OUTPUT @Hp4291;"SENS:AVER2:COUNT?"
 ENTER @Hp4291;A

SENSE:AVERage2[:STATe] {OFF|ON|0|1}

SENSE:AVERage2[:STATe] {OFF|ON|0|1}

Turns the sweep averaging function ON or OFF for the active channel. (AVERAGING ON off under (Bw/Avg))

Parameter	Description
OFF or 0	Averaging function OFF.
ON or 1	Averaging function ON.

Query response is {0|1}.

Example OUTPUT @Hp4291;"SENS:AVER2 ON"
 OUTPUT @Hp4291;"SENS:AVER2?"
 ENTER @Hp4291;A

SENSE:CORRection1 Subsystem

This subsystem is used to control the calibration function.

:CKIT {APC7|UDEFined}

SENSe:CORRection1:CKIT {APC7|UDEFined}

Selects the calibration kit. (CAL KIT:7mm, or USER KIT under **Cal**)

Parameter	Description
APC7	Select furnished calibration kit.
UDEFined	Select user-modified calibration kit.

Query response is {APC7|UDEF}.

Example OUTPUT @Hp4291;"SENS:CORR:CKIT APC7"
 OUTPUT @Hp4291;"SENS:CORR:CKIT?"
 ENTER @Hp4291;A\$

:CKIT:LABel <string>

SENSe:CORRection1:CKIT:LABel <string>

Defines a label for the user-modified calibration kit. (LABEL KIT under **Cal**)

Parameter	Description
<string>	Up to eight characters.

Example OUTPUT @Hp4291;"SENS:CORR:CKIT:LAB ""MYKIT"""

:CKIT:SAVE

SENSe:CORRection1:CKIT:SAVE

Stores the user-modified calibration kit into the memory. (SAVE USER KIT under **Cal**);
 No query)

:CKIT:STANdard1:C <numeric>

SENSe:CORRection1:CKIT:STANdard1:C <numeric>

Enters the capacitance value of the OPEN standard that is used in the user-modified calibration kit. (CAP.(C) under **Cal** CAL KIT [])

Parameter	Range	Unit
<numeric>	-1×10^{-9} to 1×10^{-9}	F

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:CORR1:CKIT:STAN1:C 53"

:CKIT:STANdard3:X <numeric>

SENSe:CORRection1:CKIT:STANdard3:X <numeric>

Enters the reactance value of the LOAD standard that is used in the user-modified calibration kit. (REACT.(X) under (Cal) CAL KIT [])

Parameter	Range	Unit
<numeric>	-1×10^6 to 1×10^6	Ω

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:CORR1:CKIT:STAN3:X 0"

:COLlect[:ACquire] {STANdard1|STANdard2|STANdard3|STANdard4}

SENSe:CORRection1:COLlect[:ACquire] {STANdard1|STANdard2|STANdard3|STANdard4}

Selects and acquires the calibration. (OPEN, SHORT, LOAD, and LOW-LOSS CAPACITOR under (Cal); No query)

The order in which you acquire the STANdard1, STANdard2, STANdard3, and STANdard4 is changeable. You can suspend a calibration sequence and do a different operation, and then resume the calibration sequence.

You must program waiting time for the calibration standard measurement to complete. An example of acquiring the OPEN standard measurement is shown below.

Parameter	Description
STANdard1	OPEN
STANdard2	SHORT
STANdard3	LOAD
STANdard4	Low-loss capacitor

```
Example
ON INTR 8 GOTO Meas_Complete
OUTPUT @Hp4291;"STAT:INST:ENAB 256;*SRE 4"
OUTPUT @Hp4291;"*CLS;*OPC?"
ENTER @Hp4291;0pc
INPUT "Connect OPEN, then press [Return]",Dum$
ENABLE INTR 8;2
    OUTPUT @Hp4291;"SENS:CORR1:COLL STAN1"
    Waiting: GOTO Waiting
Meas_complete: !
```

:COLlect:FPOints {FIXed|USER}

SENSe:CORRection1:COLlect:FPOints {FIXed|USER}

Selects the frequency points where the calibration data is collected. (CAL POINTS [] under (Cal))

SENSe:CORRection1

Parameter	Description
FIXed	172 points (analyzer specific). ¹
USER	All measurement points of current setting.

Query response is {FIX|USER}.

¹ For the frequency values of the points, see the *Function Reference*.

:COLLect:SAVE

SENSe:CORRection1:COLLect:SAVE

Calculates the error-correction coefficients from the calibration data and stores the coefficients. (DONE: CAL under **Cal**); No query)

:EDELay:STATe {OFF|ON|0|1}

SENSe:CORRection1:EDELay:STATe {OFF|ON|0|1}

Sets the port extension ON or OFF. (EXTENSION ON off under **Cal**)

Parameter	Description
OFF or 0	Port extension OFF.
ON or 1	Port extension ON.

Query response is {0|1}.

Example OUTPUT @Hp4291;"SENS:CORR:EDEL:STAT ON"
 OUTPUT @Hp4291;"SENS:CORR:EDEL:STAT?"
 ENTER @Hp4291;A

:EDELay[:TIME] <numeric>

SENSe:CORRection1:EDELay[:TIME] <numeric>

Sets the port extension value. (EXTENSION VALUE under **Cal**)

Parameter	Range	Unit
<numeric>	-10 to 10	s

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:CORR1:EDEL 0"
 OUTPUT @Hp4291;"SENS:CORR1:EDEL?"
 ENTER @Hp4291;A

[:STATe]?

SENSe:CORRection1[:STATe]?

Queries to determine if the correction state is ON or OFF (Query only)

Query response is {0|1}.

Where,

- 0 : Correction OFF (Calibration was not performed).
- 1 : Correction ON (Calibration was performed).

SENSe:CORRection1**SENSe:CORRection2 Subsystem**

This subsystem controls test fixture compensation.

:CKIT[1]:LABel <string>

SENSe:CORRection2:CKIT[1]:LABel <string>

Defines the label for the user-modified compensation kit. (LABEL KIT under COMPEN KIT, (Cal))

Parameter	Description
<string>	Up to eight characters.

Example OUTPUT @Hp4291;"SENS:CORR:CKIT:LAB ""MYKIT""

:CKIT[1]:SAVE

SENSe:CORRection2:CKIT[1]:SAVE

Stores the user-modified compensation kit into memory. (SAVE COMPEN KIT under (Cal); No query)

:CKIT[1]:STANdard1:C <numeric>

SENSe:CORRection2:CKIT[1]:STANdard1:C <numeric>

Enters the capacitance value of the OPEN standard that is used in OPEN compensation. (CAP.(C) under (Cal))

Parameter	Range	Unit
<numeric>	-1×10^{-9} to 1×10^{-9}	F

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:CORR2:CKIT:STAN1:C 53"

:CKIT[1]:STANdard1:G <numeric>

SENSe:CORRection2:CKIT[1]:STANdard1:G <numeric>

Enters the conductance value of the OPEN standard that is used in OPEN compensation. (OPEN: CONDUCT(G) under (Cal))

Parameter	Range	Unit
<numeric>	-1×10^6 to 1×10^6	S

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:CORR2:CKIT:STAN1:G 0"

:CKIT[1]:STANdard1[:SElect] {LIST|LPARAmeter}

SENSe:CORRection2:CKIT[1]:STANdard1[:SElect] {LIST|LPARAmeter}

Selects the fixture compensation open standard.

Parameter	Description
LIST	Fixture compensation open standard array
LPARAmeter	The value set by SENSe:CORRection2:CKIT[1]:STANdard1:C, and SENSe:CORRection2:CKIT[1]:STANdard1:G

Query response is {LIST|LPAR}.

:CKIT[1]:STANdard2:L <numeric>

SENSe:CORRection2:CKIT[1]:STANdard2:L <numeric>

Enters the inductance value of the SHORT standard that is used in SHORT compensation. (INDUCT. (L) under **Cal**)

Parameter	Range	Unit
<numeric>	-1×10^6 to 1×10^6	H

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:CORR2:CKIT:STAN2:L 0"

:CKIT[1]:STANdard2:R <numeric>

SENSe:CORRection2:CKIT[1]:STANdard2:R <numeric>

Enters the resistance value of the SHORT standard that is used in SHORT compensation. (SHORT: RESIST. (R) under **Cal**)

Parameter	Range	Unit
<numeric>	-1×10^6 to 1×10^6	Ω

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:CORR2:CKIT:STAN2:R 0"

:CKIT[1]:STANdard2[:SElect] {LIST|LPARAmeter}

SENSe:CORRection2:CKIT[1]:STANdard2[:SElect] {LIST|LPARAmeter}

Selects the fixture compensation short standard.

Parameter	Description
LIST	Fixture compensation short standard array
LPARAmeter	The value set by SENSe:CORRection2:CKIT[1]:STANdard2:L, and SENSe:CORRection2:CKIT[1]:STANdard2:R

Query response is {LIST|LPAR}.

:CKIT[1]:STANdard3:L <numeric>

SENSe:CORRection2:CKIT[1]:STANdard3:L <numeric>

Enters the inductance value of the LOAD standard that is used in LOAD compensation.

(INDUCT.(L) under (Cal))

Parameter	Range	Unit
<numeric>	-1×10^6 to 1×10^6	H

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:CORR2:CKIT:STAN3:L 0"

:CKIT[1]:STANdard3:R <numeric>

SENSe:CORRection2:CKIT[1]:STANdard3:R <numeric>

Enters the resistance value of the LOAD standard that is used in LOAD compensation.

(LOAD: RESIST.(R) under (Cal))

Parameter	Range	Unit
<numeric>	-1×10^6 to 1×10^6	Ω

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:CORR2:CKIT:STAN3:R 50"

:CKIT[1]:STANdard3[:SElect] {LIST|LPARameter}

SENSe:CORRection2:CKIT[1]:STANdard3[:SElect] {LIST|LPARameter}

Selects the fixture compensation load standard.

Parameter	Description
LIST	Fixture compensation load standard array
LPARameter	The value set by SENSe:CORRection2:CKIT[1]:STANdard3:L, and SENSe:CORRection2:CKIT[1]:STANdard3:R

Query response is {LIST|LPAR}.

:CKIT2 {TEFLon|UDEfined}

SENSe:CORRection2:CKIT2 {TEFLon|UDEfined}

Selects the load standard used for the dielectric material test fixture compensation.

(Option 002 only)

Parameter	Description
TEFLon	Load standard furnished with the 16453A.
UDEfined	User-defined load standard.

Example OUTPUT @Hp4291;"SENS:CORR2:CKIT2 TEFL"

:CKIT2:LABel <string>

SENSe:CORRection2:CKIT2:LABel <string>

Defines the label of the user-modified compensation kit for the permittivity measurement. (LABEL KIT under COMPEN KIT, **[Cal]**); Option 002 only)

Parameter	Description
<string>	Up to eight characters.

Example OUTPUT @Hp4291;"SENS:CORR2:CKIT2:LAB "MYKIT""

:CKIT2:SAVE

SENSe:CORRection2:CKIT2:SAVE

Stores the user-modified compensation kit for the permittivity measurement into memory. (SAVE COMPEN KIT under **[Cal]**); No query, Option 002 only)

:CKIT2:STANdard6:PREal <numeric>

SENSe:CORRection2:CKIT2:STANdard6:PREal <numeric>

Enters the relative permittivity (ϵ_r') value of the LOAD standard that is used in LOAD compensation for the permittivity measurement. (LOAD: ϵ_r REAL under **[Cal]**); Option 002 only)

Parameter	Range	Unit
<numeric>	-1×10^6 to 1×10^6	—

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:CORR2:CKIT2:STAN6:PRE 10"

:CKIT2:STANdard6:PLFactor <numeric>

SENSe:CORRection2:CKIT2:STANdard6:PLFactor <numeric>

Enters the relative dielectric loss index (ϵ_r'') value of the user-modified LOAD standard that is used in LOAD compensation for the permittivity measurement. (ϵ_r LOSS under **[Cal]**); Option 002 only)

Parameter	Range	Unit
<numeric>	-1×10^6 to 1×10^6	—

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:CORR2:CKIT2:STAN6:PIM 10"

:COLLect:SAVE

SENSe:CORRection2:COLLect:SAVE

Calculates the fixture compensation coefficients and stores the coefficients.

(DONE: COMPEN under **Cal**); No query)

:OPEN[:STATe] {OFF|ON|0|1}

SENSe:CORRection2:OPEN[:STATe] {OFF|ON|0|1}

Sets the OPEN fixture compensation ON or OFF. (OPEN ON off under **Cal**)

Parameter	Description
OFF or 0	OPEN fixture compensation OFF.
ON or 1	OPEN fixture compensation ON.

Query response is {0|1}.

Example OUTPUT @Hp4291;"SENS:CORR2:OPEN ON"
 OUTPUT @Hp4291;"SENS:CORR2:OPEN?"
 ENTER @Hp4291;A

:SHORT[:STATe] {OFF|ON|0|1}

SENSe:CORRection2:SHORT[:STATe] {OFF|ON|0|1}

Sets the SHORT fixture compensation ON or OFF. (SHORT ON off under **Cal**)

Parameter	Description
OFF or 0	SHORT fixture compensation OFF.
ON or 1	SHORT fixture compensation ON.

Query response is {0|1}.

Example OUTPUT @Hp4291;"SENS:CORR2:SHOR ON"
 OUTPUT @Hp4291;"SENS:CORR2:SHOR?"
 ENTER @Hp4291;A

:LOAD[:STATe] {OFF|ON|0|1}

SENSe:CORRection2:LOAD[:STATe] {OFF|ON|0|1}

Sets the LOAD fixture compensation ON or OFF. (LOAD ON off under **Cal**)

Parameter	Description
OFF or 0	LOAD fixture compensation OFF.
ON or 1	LOAD fixture compensation ON.

Query response is {0|1}.

Example OUTPUT @Hp4291;"SENS:CORR2:LOAD ON"
 OUTPUT @Hp4291;"SENS:CORR2:LOAD?"
 ENTER @Hp4291;A

:SPAN {<numeric>|DMARKer|MZAPerture}

SENSe:FREQuency:SPAN {<numeric>|DMARKer|MZAPerture}

Sets the frequency span. ((SPAN))

Parameter	Range or Description	Unit
<numeric>	0 to 1.799×10^9 (= 1.799 G)	Hz (Frequency)
DMARKer	The difference between the marker and the Δ marker values. (MKRA→CENTER under (Span) or (Marker→))	
MZAPerture	The "frequency span x zooming aperture." (MKR ZOOM under (Marker→))	

Query response is <numeric>.

```
Example          OUTPUT @Hp4291;"SENS:FREQ:SPAN 1GHZ"
                  OUTPUT @Hp4291;"SENS:FREQ:SPAN?"
                  ENTER @Hp4291;A
```

:START {<numeric>|MARKer}

SENSe:FREQuency:START {<numeric>|MARKer}

Sets the start frequency value. ((Start))

Parameter	Range or Description	Unit
<numeric>	1×10^6 (= 1 M) to 1.8×10^9 (= 1.8 G)	Hz (Frequency)
MARKer	The sweep parameter value of the marker. (MKR→START under (Marker→))	

Query response is <numeric>.

```
Example          OUTPUT @Hp4291;"SENS:FREQ:STAR 100MAHZ"
                  OUTPUT @Hp4291;"SENS:FREQ:STAR?"
                  ENTER @Hp4291;A
```

:STOP {<numeric>|MARKer}

SENSe:FREQuency:STOP {<numeric>|MARKer}

Sets the stop frequency value. ((STOP))

Parameter	Range or Description	Unit
<numeric>	1×10^6 (= 1 M) to 1.8×10^9 (= 1.8 G)	Hz (Frequency)
MARKer	The frequency value of the marker. (MKR→STOP under (Marker→))	

Query response is <numeric>.

```
Example          OUTPUT @Hp4291;"SENS:FREQ:STOP 1.8GHZ"
                  OUTPUT @Hp4291;"SENS:FREQ:STOP?"
                  ENTER @Hp4291;A
```

SENSe:FREQuency**SENSe:LIST Subsystem**

This subsystem is used to control the list sweep function.

:CLEAr

SENSe:LIST:CLEAr

Clears the entire list. (CLEAR LIST under **Sweep**); No query)

:SAVE

SENSe:LIST:SAVE

Completes editing the frequency sweep list. (LIST DONE under **Sweep**); No query)

:SEGMENT <numeric>

SENSe:LIST:SEGMENT <numeric>

Selects the segment to edit.

Parameter	Description
<numeric>	Segment number, 1 to 15

Query response is <numeric>.

:SEGMENT:ADD

SENSe:LIST:SEGMENT:ADD

Adds a new segment to a list sweep table. (ADD under **Sweep**); No query)

:SEGMENT:AVERAGE:COUNT <numeric>

SENSe:LIST:SEGMENT:AVERAGE:COUNT <numeric>

Sets the point averaging factor for the list sweep table. (AVERAGING ON POINT under **Sweep**)

Parameter	Range	Unit
<numeric>	0 to 999	—

Query response is <numeric>.

Related Command SENSe:LIST:SEGMENT:ADD

Example OUTPUT @Hp4291;"SENS:LIST:SEGM:ADD"
 OUTPUT @Hp4291;"SENS:LIST:SEGM:AVER:COUN 4"
 OUTPUT @Hp4291;"SENS:LIST:SEGM:AVER:COUN?"
 ENTER @Hp4291;A

:SEGMENT:CURRENT <numeric>

SENSe:LIST:SEGMENT:CURRENT <numeric>

Sets the OSC level for the list sweep table to A (Ampere). (OSC LEVEL under **Sweep** LIST MENU EDIT)

Parameter	Range	Unit
<numeric>	-4×10^{-6} to 20×10^{-3}	A

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:LIST:SEGM:ADD"
 OUTPUT @Hp4291;"SENS:LIST:SEGM:CURR 1"
 OUTPUT @Hp4291;"SENS:LIST:SEGM:CURR?"
 ENTER @Hp4291;A

:SEGMENT:DELETE

SENSe:LIST:SEGMENT:DELETE

Deletes a segment from a list sweep table. (DELETE under **Sweep**); No query)

:SEGMENT:EDIT

SENSe:LIST:SEGMENT:EDIT

Edits the segment. (EDIT under **Sweep**); No query)

:SEGMENT:FREQUENCY:CENTER <numeric>

SENSe:LIST:SEGMENT:FREQUENCY:CENTER <numeric>

Sets center the frequency value of the segment in the list sweep table. (Center), or CENTER under **Sweep**)

Parameter	Range	Unit
<numeric>	1000000 (= 1 M) to 1.8×10^9 (= 1.8 G)	Hz

Query response is <numeric>.

:SEGment:POINTs <numeric>

SENSe:LIST:SEGment:POINTs <numeric>

Sets the number of points for the segment for the list sweep table.

(NUMBER OF POINTS under **Sweep**)

Parameter	Description
<numeric>	2 to 801.

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:LIST:SEGM:POIN 201"
 OUTPUT @Hp4291;"SENS:LIST:SEGM:POIN?"
 ENTER @Hp4291;A

:SEGment:POWER <numeric>

SENSe:LIST:SEGment:POWER <numeric>

Sets the OSC level for the list sweep table to dBm. (OSC LEVEL under **Sweep**)

(LIST MENU EDIT)

Parameter	Range	Unit
<numeric>	-67 to +7	dBm

Query response is <numeric>.

Example OUTPUT @Hp4291;"SENS:LIST:SEGM:POW 0"
 OUTPUT @Hp4291;"SENS:LIST:SEGM:POW?"
 ENTER @Hp4291;A

:SEGment:QUIT

SENSe:LIST:SEGment:QUIT

Quits editing a segment of the list sweep table. (SEGMENT QUIT under **Sweep**); No query)**:SEGment:SAVE**

SENSe:LIST:SEGment:SAVE

Completes modifying a segment in a list sweep table. (SEGMENT DONE under **Sweep**); No query)

:SEGMent:VOLTage <numeric>

SENSe:LIST:SEGMent:VOLTage <numeric>

Sets the OSC level of the segment in the list sweep table with unit of V (Volt).

(OSC LEVEL under **Sweep** LIST MENU EDIT)

Parameter	Range	Unit
<numeric>	0.2x10 ⁻³ to 1	V

Query response is <numeric>.

Example

```
OUTPUT @Hp4291;"SENS:LIST:SEGM:VOLT 0"
OUTPUT @Hp4291;"SENS:LIST:SEGM:VOLT?"
ENTER @Hp4291;A
```

SENSe:SWEep:COUNT <numeric>

SENSe:SWEep:COUNT <numeric>

Defines number of sweeps. (NUMBER OF GROUPS under **Trigger**)

Parameter	Description
<numeric>	Greater than 0 (if <numeric> is 0 or less than 0, it is set to 1.)

Query response is <numeric>.

Example

```
OUTPUT @Hp4291;"INIT:CONT OFF"
OUTPUT @Hp4291;"SENS:SWE:COUNT 10"
OUTPUT @Hp4291;"INIT"
```

SENSe:SWEep:DWELl{1|2} <numeric>

SENSe:SWEep:DWELl{1|2} <numeric>

Defines the delay times set by SENSe:SWEep:DWELl{1|2}:AUTO. (POINT DELAY TIME, SWEEP DELAY TIME under **Sweep**)

Node	Description
:DWEL11	Point delay time (POINT DELAY TIME)
:DWEL12	Sweep delay time (SWEEP DELAY TIME)

Parameter	Range	Unit
<numeric>	0 to 3600 (resolution : 12.5 μ)	s

Query response is <numeric>.

Related Command

```
SENSe:SWEep:DWELl{1|2}:AUTO
```

```

Example      OUTPUT @Hp4291;"SENS:SWE:DWEL 10"
             OUTPUT @Hp4291;"SENS:SWE:DWEL:AUTO ON"

             OUTPUT @Hp4291;"SENS:SWE:DWEL?"
             ENTER @Hp4291;A

```

SENSe:SWEep:DWEL1{1|2}:AUTO {OFF|ON|0|1}

SENSe:SWEep:DWEL1{1|2}:AUTO {OFF|ON|0|1}

Sets the delay time to the value defined by SENSe:SWEep:DWEL1{1|2} <numeric>.

Node	Description
:DWEL11	Point delay time (POINT DELAY TIME)
:DWEL12	Sweep delay time (SWEEP DELAY TIME)

Parameter	Description
OFF or 0	Sets the delay time 0.
ON or 1	Sets the delay time to value set by SENSe:SWEep:DWEL1{1 2} <numeric>.

Query response is {0|1}.

Related Command SENSe:SWEep:DWEL1{1|2}

SENSe:SWEep:POINTs <numeric>

SENSe:SWEep:POINTs <numeric>

Sets the number of points. (NUMBER OF POINTS under **Sweep**)

Parameter	Description
<numeric>	2 to 801.

Query response is <numeric>.

```

Example      OUTPUT @Hp4291;"SENS:SWE:POIN 201"
             OUTPUT @Hp4291;"SENS:SWE:POIN?"
             ENTER @Hp4291;A

```

SENSe:SWEep:SPACing {LINear|LOGarithmic}

SENSe:SWEep:SPACing {LINear|LOGarithmic}

Selects the sweep type. (SWEEP TYPE:LIN, LOG under **Sweep**)

Parameter	Description
LINear	Linear frequency
LOGarithmic	Log frequency

Query response is {LIN|LOG}.

SENSe:SWEep:SPACing

Related Command `DISPlay[:WINDow]:TRACe{1-17}:X:SPACing`
 `SENSe:FREQuency:MODE SWEep`
 `SOURce1:SWEep:SPACing {LINear|LOGarithmic}`
 `SOURce2:SWEep:SPACing {LINear|LOGarithmic}`

Example To select linear frequency sweep

```
OUTPUT @Hp4291;"SENS:SWE:SPAC LIN"
OUTPUT @Hp4291;"DISP:TRAC:X:SPAC LIN"
```

Query command is,

```
OUTPUT @Hp4291;"SENS:SWE:SPAC?"
ENTER @Hp4291;A$
```

SENSe:SWEep:TIME <numeric>

`SENSe:SWEep:TIME <numeric>`

Sets the sweep time. Setting this value automatically turns `SENSe:SWEep:TIME:AUTO` OFF.
 (`SWEEP TIME []` under `Sweep`)

Parameter	Range	Unit
<numeric>	(depends on the analyzer's setting)	s

Query response is <numeric>.

Example `OUTPUT @Hp4291;"SENS:SWE:TIME 10"`
 `OUTPUT @Hp4291;"SENS:SWE:TIME?"`
 `ENTER @Hp4291;A`

SENSe:SWEep:TIME:AUTO {OFF|ON|0|1}

`SENSe:SWEep:TIME:AUTO {OFF|ON|0|1}`

Sets the automatic or manual sweep time. The automatic mode gives the fastest sweep time at the analyzer's current settings for the channel. (`SWEEP TIME AUTO` under `Sweep`)

Parameter	Description
OFF or 0	Manual sweep time
ON or 1	Automatic sweep time

Query response is {0|1}.

Example `OUTPUT @Hp4291;"SENS:SWE:TIME:AUTO ON"`
 `OUTPUT @Hp4291;"SENS:SWE:TIME:AUTO?"`
 `ENTER @Hp4291;A`

SOURce1:{CURRent|POWer}[:LEVel][:IMMediate][:AMPLitude]

SING

(*Simple Command*) See “Simple Commands” later in this chapter.

SOURce Subsystem

The SOURce subsystem controls the signal source of the analyzer.

- Controls the ac source (SOURce1).
- Controls the dc source (SOURce2, option 001 only).

SOURce1:FREQuency[:CW|:FIXed] <numeric>

SOURce1:FREQuency[:CW|:FIXed] <numeric>

Sets the frequency for the OSC level, dc voltage bias, or dc current bias sweep. (CW FREQ under Source)

Parameter	Range	Unit
<numeric>	1000000 (= 1 M) to 1.8×10^9 (= 1.8 G)	Hz

Query response is <numeric>.

```
Example          OUTPUT @Hp4291;"SOUR:FREQ 500MAHZ"
                  OUTPUT @Hp4291;"SOUR:FREQ?"
                  ENTER @Hp4291;A
```

SOURce1:{CURRent|POWer}[:LEVel][:IMMediate][:AMPLitude] <numeric>

SOURce1:{CURRent|POWer}[:LEVel][:IMMediate][:AMPLitude] <numeric>

Sets the OSC level for a frequency sweep. (OSC LEVEL under Source)

Node	Description
:CURRent	Current
:POWer	Power

Parameter	Range	Unit
<numeric>	$+4 \times 10^{-6}$ to 20×10^{-3}	A (:CURRent)
	-67 to 7	dBm (:POWer)

Query response is <numeric>.

Related Command SOURce1:VOLTage[:LEVel][:IMMediate][:AMPLitude]

SOURce1:SWEEp:DIRection {UP|DOWN}

SOURce1:SWEEp:DIRection {UP|DOWN}

Selects a sweep direction of UP or DOWN when in the OSC level sweep mode.

(SWEEP DIR [] under **Sweep**)

Parameter	Description
UP	Sweep direction UP.
DOWN	Sweep direction DOWN.

Query response is {UP|DOWN}.

SOURce1:SWEEp:SPACing {LINear|LOGarithmic}

SOURce1:SWEEp:SPACing {LINear|LOGarithmic}

Selects the sweep type in the OSC level sweep. (SWEEP TYPE:LIN, LOG under **Sweep**)

Parameter	Description
LINear	Sweep type linear.
LOGarithmic	Sweep type logarithmic.

Query response is {LIN|LOG}.

Related Command DISPlay[:WINDow]:TRACe{1-17}:X:SPACing

Example To select linear OSC level sweep

```
OUTPUT @Hp4291;"SOUR:SWE:SPAC LIN"
OUTPUT @Hp4291;"DISP:TRAC:X:SPAC LIN"
```

SOURce1:VOLTage Subsystem

This subsystem is used to control the ac signal source.

:CENTer {<numeric>|DMARker|MARKer|TPEak}

SOURce1:VOLTage:CENTer {<numeric>|DMARker|MARKer|TPEak}

Defines the OSC level sweep center value. (**Center**)

Parameter	Range or Description	Unit
<numeric>	0.2x10 ⁻³ to 1	V
DMARker	The difference value between the marker and the Δmarker values. (MKRA→CENTER under Marker→)	
MARKer	The sweep parameter value of the marker and centers the new span about that value. (MKR→CENTER under Marker→)	
TPEak	The sweep parameter value of that peak. (PEAK→CENTER under Marker→)	

Query response is <numeric>.

[[:LEVel]][:IMMediate]][:AMPLitude] <numeric>

SOURce1:VOLTage[:LEVel]][:IMMediate]][:AMPLitude] <numeric>

Sets the OSC level for frequency/DC-I/DC-V sweep. (OSC LEVEL under (Source))

Parameter	Range	Unit
<numeric>	-200×10^{-6} to 1	V

Query response is <numeric>.

:MODE {FIXed|LIST|SWEep}

SOURce1:VOLTage:MODE {FIXed|LIST|SWEep}

- Selects the sweep source. (SWP SRC:FREQ, OSC LEVEL, DC BIAS V, DC BIAS I under (Sweep))
- Selects the list sweep mode. (SWP TYPE: LIST under (Sweep))

Parameter	Description
FIXed	Frequency/DC-I/DC-V sweep
LIST	Frequency list sweep
SWEep	OSC level sweep

Query response is {FIX|SWE}.

Related Command SENSE:FREQUENCY:MODE
 SOURCE2:CURRENT:MODE
 SOURCE2:VOLTage:MODE

Example To select the frequency sweep,

```
OUTPUT @Hp4291;"SENS:FREQ:MODE SWE"
OUTPUT @Hp4291;"SOUR1:VOLT:MODE FIX"
OUTPUT @Hp4291;"SOUR2:CURR:MODE FIX"
OUTPUT @Hp4291;"SOUR2:VOLT:MODE FIX"
```

To select the OSC level sweep,

```
OUTPUT @Hp4291;"SENS:FREQ:MODE FIX"
OUTPUT @Hp4291;"SOUR1:VOLT:MODE SWE"
OUTPUT @Hp4291;"SOUR2:CURR:MODE FIX"
OUTPUT @Hp4291;"SOUR2:VOLT:MODE FIX"
```

To set list sweep,

```
OUTPUT @Hp4291;"SENS:FREQ:MODE LIST"
OUTPUT @Hp4291;"SOUR1:VOLT:MODE LIST"
OUTPUT @Hp4291;"SOUR2:CURR:MODE FIX"
OUTPUT @Hp4291;"SOUR2:VOLT:MODE FIX"
```

:SPAN {<numeric>|DMARKer|MZAPerture}

SO URce1:VOLTage:SPAN {<numeric>|DMARKer|MZAPerture}

Sets the OSC level sweep range. ((SPAN))

Parameter	Range	Unit
<numeric>	0 to 0.9998	V (volt) ¹

Query response is <numeric>.

1 The maximum range depends on the center value.

```

Example      OUTPUT @Hp4291;"SOUR1:VOLT:CENT 0.5V"
              OUTPUT @Hp4291;"SOUR1:VOLT:SPAN 0.1V"

              OUTPUT @Hp4291;"SOUR1:VOLT:SPAN?"
              ENTER @Hp4291;A

```

:STARt {<numeric>|MARKer}

SO URce1:VOLTage:STARt {<numeric>|MARKer}

Sets the OSC level sweep start value. ((Start))

Parameter	Range or Description	Unit
<numeric>	0.2×10^{-3} to 1	V
MARKer	The sweep parameter value of the marker. (MKR→STARt under (Marker→))	

Query response is <numeric>.

```

Example      OUTPUT @Hp4291;"SOUR1:VOLT:STAR 0.2V"
              OUTPUT @Hp4291;"SOUR1:VOLT:STOP 0.6V"

              OUTPUT @Hp4291;"SOUR1:VOLT:STAR?"
              ENTER @Hp4291;A

```

:STOP {<numeric>|MARKer}

SO URce1:VOLTage:STOP {<numeric>|MARKer}

Sets the OSC level sweep stop value. ((Stop))

Parameter	Range or Description	Unit
<numeric>	0.2×10^{-3} to 1	V
MARKer	The sweep parameter value of the marker. (MKR→STOP under (Marker→))	

Query response is <numeric>.

SOURCE2:{CURRENT|VOLTAGE} Subsystem

This subsystem is used to control the dc source.

:ALC[:STATe] {OFF|ON|0|1}

SOURCE2:{CURRENT|VOLTAGE}:ALC[:STATe] {OFF|ON|0|1}

Sets dc bias mode to either the voltage setting mode or the current setting mode.

(BIAS SRC [] under (Source))

Setting the SOURCE2:CURRENT:ALC ON sets the SOURCE2:VOLTAGE:ALC OFF, and selects the dc bias current setting mode. (BIAS SRC [CURRENT])

Setting the SOURCE2:VOLTAGE:ALC ON sets the SOURCE2:CURRENT:ALC OFF, and selects the dc bias voltage setting mode. (BIAS SRC [VOLTAGE])

:CENTer {<numeric>|DMARker|MARKer|TPEak}

SOURCE2:{CURRENT|VOLTAGE}:CENTer {<numeric>|DMARker|MARKer|TPEak}

Sets the dc bias sweep center value. ((Center))

Parameter	Range to Description	Unit
<numeric>	-100×10 ⁻³ to 100×10 ⁻³ -40 to 40	A (for :CURRENT node) V (for :VOLTAGE node)
DMARker	The difference value between the marker and the Δmarker values. (MKR→CENTER under (Center) and (Marker→))	
MARKer	The sweep parameter value of the marker and centers the new span about that value. (MKR→CENTER under (Marker→))	
TPEak	Searches for a peak using the marker and then changes the CENTER of the destination channel to the sweep parameter value of that peak. (PEAK→CENTER under (Center) or (Marker→))	

Query response is <numeric>.

[:LEVel][:IMMediate][:AMPLitude] <numeric>

SOURCE2:{CURRENT|VOLTAGE}[:LEVel][:IMMediate][:AMPLitude] <numeric>

Set dc bias value. (BIAS VOLTAGE , BIAS CURRENT under (Source))

Node	Description
:CURRENT	Sets the DC-I level (BIAS CURRENT)
:VOLTAGE	Sets the DC-V level (BIAS VOLTAGE)

Parameter	Range	Unit
<numeric>	-100×10 ⁻³ to 100×10 ⁻³ -40 to +40	A (for :CURRENT node) V (for :VOLTAGE node)

Query response is <numeric>.

:LIMit[:AMPlitude] <numeric>

SOURCE2:{CURRENT|VOLTage}:LIMit[:AMPlitude] <numeric>

Sets the voltage/current limit in the dc bias sweep mode. (BIAS CUR LIMIT, BIAS VOLT LIMIT under (Source))

Node	Description
:CURRENT	Sets the current limit (BIAS CUR LIMIT)
:VOLTage	Sets the voltage limit (BIAS VOLT LIMIT)

Parameter	Range	Unit
<numeric>	1 to +40	V (for :VOLTage node)
<numeric>	2×10^{-3} to 100×10^{-3}	A (for :CURRENT node)

Query response is <numeric>.

:MODE {FIXed|SWEep}

SOURCE2:{CURRENT|VOLTage}:MODE {FIXed|SWEep}

Selects the sweep source. (SWP SRC:FREQ, OSC LEVEL, DC BIAS V, DC BIAS I under (Sweep))

Node	Description
:CURRENT	Current
:VOLTage	Voltage

Parameter	Description
FIXed	Fixed dc bias (frequency or OSC level sweep)
SWEep	dc bias sweep mode

Query response is {FIX|SWE}.

Related Command SENSE:FREQUENCY:MODE
 SOURce1:VOLTage:MODE
 SOURce2:{CURRENT|VOLTage}:STATE

Example To set the sweep source to dc bias voltage

```
OUTPUT @Hp4291;"SENS:FREQ:MODE FIX"
OUTPUT @Hp4291;"SOUR1:VOLT:MODE FIX"
OUTPUT @Hp4291;"SOUR2:CURR:MODE FIX"
OUTPUT @Hp4291;"SOUR2:VOLT:MODE SWE"
OUTPUT @Hp4291;"SOUR2:VOLT:STAT ON"
```

:SPAN {<numeric>|DMARker|MZAPerture}

SOURCE2:{CURRENT|VOLTage}.SPAN {<numeric>|DMARker|MZAPerture}

Sets the dc bias sweep span. ((Span))

Node	Description	
:CURRENT	Current	
:VOLTage	Voltage	

Parameter	Range or Description	Unit
<numeric>	0 to 200×10^{-3}	A (for :CURRENT node)
<numeric>	0 to 80	V (for :VOLTage node)
DMARker	Sets the dc bias sweep range of the marker to the Δ marker. (MKR Δ —SPAN under (Marker→))	
MZAPerture	Sets the dc bias sweep range as follows: <ul style="list-style-type: none"> • Center to the marker value. • Span to (Current span) \times (Zooming aperture) (MKR ZOOM under (Marker→))	

Query response is <numeric>.

:START {<numeric>|MARKer}

SOURCE2:{CURRENT|VOLTage}.START {<numeric>|MARKer}

Sets the dc bias sweep start value. ((Start))

Node	Description	
:CURRENT	Current	
:VOLTage	Voltage	

Parameter	Range or Description	Unit
<numeric>	-100×10^{-3} to 100×10^{-3}	A (for :CURRENT node)
	-40 to 40	V (for :VOLTage node)
MARKer	Sets the dc bias sweep start to the marker. (MKR→START under (Marker→))	

Query response is <numeric>.

:STATE {OFF|ON|0|1}

SOURCE2:{CURRENT|VOLTage}.STATE {OFF|ON|0|1}

Sets the dc bias output ON or OFF. (DC BIAS ON off under (Source))

The dc bias ON and OFF states are common to both channels.

Node	Description
:CURRENT	Current
:VOLTage	Voltage

Parameter	Description
OFF or 0	dc bias OFF
ON or 1	dc bias ON

Query response is {0|1}.

SOURce2:{CURRent|VOLTage}

Example OUTPUT @Hp4291;"SOUR2:VOLT:STAT ON"
 OUTPUT @Hp4291;"SOUR2:VOLT:STAT?"
 ENTER @Hp4291;A

:STOP {<numeric>|MARKer}

SOURce2:{CURRent|VOLTage}:STOP {<numeric>|MARKer}

Sets the dc bias sweep stop value. (**Stop**)

Node	Description	
:CURRent	Current	
:VOLTage	Voltage	
Parameter	Range or Description	Unit
<numeric>	-100×10 ⁻³ to 100×10 ⁻³	A (for :CURRent node)
	-40 to 40	V (for :VOLTage node)
MARKer	Sets the dc bias sweep stop to the marker.. (MKR→STOP under Marker→)	

Query response is <numeric>.

SOURce2:SWEEP:DIRection {UP|DOWN}

SOURce2:SWEEP:DIRection {UP|DOWN}

Selects the sweep direction (UP or DOWN) when in the dc bias sweep mode. (**SWEEP DIR []** under **Sweep**)

Parameter	Description
UP	Sweep direction UP.
DOWN	Sweep direction DOWN.

Query response is {UP|DOWN}.

SOURce2:SWEEP:SPACing {LINear|LOGarithmic}

SOURce2:SWEEP:SPACing {LINear|LOGarithmic}

Selects the sweep type in dc bias sweep. (**SWEEP TYPE: LIN**, **LOG** under **Sweep**)

Parameter	Description
LINear	Sweep type linear.
LOGarithmic	Sweep type logarithmic.

Query response is {LIN|LOG}.

Related Command DISPlay[:WINDow]:TRACe{1-17}:X:SPACing

Example To select linear sweep in OSC level sweep,
 OUTPUT @Hp4291;"SOUR2:SWE:SPAC LIN"
 OUTPUT @Hp4291;"DISP:TRAC:X:SPAC LIN"

SPAN <numeric>, STAR <numeric>

(*Simple Command*) See “Simple Commands” later in this chapter.

STATus Subsystem

The analyzer has a status-reporting structure that is defined by IEEE 488.2 and SCPI. The STATus subsystem controls the SCPI-defined status-reporting structures, which are STATus:INSTRument, STATus:OPERation, and STATus:QUESTionable registers. The common commands (listed under “Related Command”) control the IEEE 488.2-defined status reporting structure.

For the status register bits assignment and usage, see the *Programming Manual*.

Related Command	*CLS
	*ESE
	*ESR
	*OPC
	*SRE
	*STB?

STATus:INSTRument Subsystem

This subsystem is used to control the Instrument Status Registers (Instrument Event Status Register and Instrument Event Status Enable Register).

:ENABLE <numeric>

STATus:INSTRument:ENABle <numeric>

Sets the contents of the Instrument Event Status Enable Register.

Parameter	Range
<numeric>	Decimal expression of the contents of the register; 0 to 32767 ($=2^{15}-1$)

Query response is <numeric>.

```
Example      OUTPUT @Hp4291;"STAT:INST:ENAB 8"
              OUTPUT @Hp4291;"STAT:INST:ENAB?"
              ENTER @Hp4291;A
```

[:EVENT]?

STATus:INSTRument[:EVENT]?

Returns the contents of the Instrument Event Status Register. (Query Only)

Reading the event register clears it.

Query response is <numeric>.

```
Example      OUTPUT @Hp4291;"STAT:INST?"
              ENTER @Hp4291;A
```

STATus:INSTRument**STATus:OPERation Subsystem**

This subsystem is used to control the Operation Status Registers (Operation Event Status Register, Operation Event Status Enable Register, and Operation Status Condition Register) and the transition filters.

:CONDition?

STATus:OPERation:CONDition?

Queries the contents of Operation Status Condition Register. (Query only)

Query response is *<numeric>*.

Example OUTPUT @Hp4291;"STAT:OPER:COND?"
 ENTER @Hp4291;A

:ENABle <numeric>

STATus:OPERation:ENABle <numeric>

Sets the contents of the Operation Status Enable Register.

Parameter	Description
<i><numeric></i>	Decimal expression of the contents of the register; 0 to 65535 ($=2^{16}-1$)

Query response is *<numeric>*.

Example OUTPUT @Hp4291;"STAT:OPER:ENAB 1"
 OUTPUT @Hp4291;"STAT:OPER:ENAB?"

[:EVENT]?

STATus:OPERation[:EVENT]?

Queries the contents of the Operation Status Event Register. (Query only)

Reading the event register clears it.

Query response is *<numeric>*.

Example OUTPUT @Hp4291;"STAT:OPER?"
 ENTER @Hp4291;A

:NTRansition <numeric>

STATus:OPERation:NTRansition <numeric>

Sets the negative transition filter of the Operation Status Register.

Parameter	Description
<numeric>	Decimal expression of the contents of the register; 0 to 65535 ($=2^{16}-1$)

Query response is <numeric>.

Example

```
OUTPUT @Hp4291;"STAT:OPER:NTR 1"
OUTPUT @Hp4291;"STAT:OPER:NTR?"
ENTER @Hp4291;A
```

:PTRansition <numeric>

STATus:OPERation:PTRansition <numeric>

Sets the positive transition filter of the Operational Status Register.

Parameter	Description
<numeric>	Decimal expression of the contents of the register; 0 to 65535 ($=2^{16}-1$)

Query response is <numeric>.

Example

```
OUTPUT @Hp4291;"STAT:OPER:PTR 1"
OUTPUT @Hp4291;"STAT:OPER:PTR?"
ENTER @Hp4291;A
```

STATus:PRESet

STATus:PRESet

Presets the Operation and Questionable Status Enable Registers and transition filters. This command does not affect the Event Registers. (No query)

This command does not affect the Instrument Event Status Register and the Enable Register.

Register/Filter	Preset value
Enable register	0
Positive transition filter	65535 ($2^{16}-1$: all bits are 1)
Negative transition filter	0

STATUS:PRESet

STATUS:QUESTIONable Subsystem

This subsystem is used to control the Questionable Status Registers (Questionable Event Status Register, Questionable Event Status Enable Register, and Questionable Status Condition Register). The analyzer has no operation that reports an event to the Questionable Status Register.

:CONDition?

STATUS:QUESTIONable:CONDition?

Queries the contents of the Questionable Status Condition Register. (Query only)

The analyzer has no operation that reports an event to the questionable register.

:ENABle <numeric>

STATUS:QUESTIONable:ENABle <numeric>

Sets the value of the Questionable Status Enable Register.

The analyzer has no operation that reports an event to the questionable register.

[:EVENT]?

STATUS:QUESTIONable[:EVENT]?

Queries the contents of the Questionable Event Status Register. (Query only)

The analyzer has no operation that reports an event to the questionable register.

STOP <numeric>

(*Simple Command*) See “Simple Commands” later in this chapter.

SType {LIN|LOG|LIST}, **SWED** {UP|DOWN},
SWEDTIME <numeric>, **SWES** {FREQ|OLEV|DCV|DCI}

(*Simple Command*) See “Simple Commands” later in this chapter.

SYSTem Subsystem**SYSTem:BEEPer**{1|2}:**STATe** {OFF|ON|0|1}

SYSTem:BEEPer{1|2}:STATe {OFF|ON|0|1}

Sets an annunciator that sounds to indicate completion of certain operations or as a warning.

(BEEP DONE ON off, BEEP WARN ON off under **(System)**)

Node	Description
BEEPer1	An annunciator that sounds to indicate completion of certain operations such as calibration or instrument state save. (BEEP DONE ON off under (System))
BEEPer2	The warning annunciator. When the annunciator is ON, it sounds a warning when a cautionary message is displayed. (BEEP WARN ON off under (System))

Parameter	Description
OFF or 0	Beeper OFF
ON or 1	Beeper ON

Query response is {0|1}.

```
Example          OUTPUT @Hp4291;"SYST:BEEP1:STAT ON"
                  OUTPUT @Hp4291;"SYST:BEEP1:STAT?"
                  ENTER @Hp4291;A
```

SYSTem:COMMunicate:GPIB Subsystem

This subsystem sets the GPIB address that the analyzer will use to communicate with other devices on the GPIB.

:CONTROLLER:ADDRESS <numeric>

SYSTem:COMMunicate:GPIB:CONTROLLER:ADDRess <numeric>

Sets the GPIB address that the analyzer will use to communicate with the external controller. (ADDRESS: CONTROLLER under **Local**)

Parameter	Range	Unit
<numeric>	0 to 30	–

Query response is <numeric>.

Example OUTPUT @Hp4291;"SYST:COMM:GPIB:CONT:ADDR 8"
 OUTPUT @Hp4291;"SYST:COMM:GPIB:CONT:ADDR?"
 ENTER @Hp4291;A

SYSTEM:COMMunicate:PARallel Subsystem

This subsystem controls the 8-bit parallel I/O port.

[:RECEive]:DATA?

SYSTem:COMMunicate:PAPallel[:RECEive]:DATA?

Reads data from the 4-bit parallel input of the analyzer. (Query only)

Query response is <numeric>.

Example OUTPUT @Hp4291;"SYST:COMM:PAR:DATA?"
 ENTER @Hp4291;A

:TRANsmit:DATA <numeric>

SYSTem:COMMunicate:PAPallel:TRANsmit:DATA <numeric>

Outputs the data to the 8-bit parallel output port. (No query)

Parameter	Range	Unit
<numeric>	0 to 255	–

Example OUTPUT @Hp4291;"SYST:COMM:PAR:TRAN:DATA 0"

SYSTem:DATE <numeric(year)>,<numeric(month)>,<numeric(day)>

SYSTem:DATE <numeric(year)>,<numeric(month)>,<numeric(day)>

Sets the date of the internal clock. (DATE MM/DD/YY under (System))

Parameter	Range	Unit
<numeric(year)>	1900 to 2099	—
<numeric(month)>	1 to 12	—
<numeric(day)>	1 to 31	—

Query response is <numeric(year)><numeric(month)><numeric(day)>.

Example OUTPUT @Hp4291;"SYST:DATE 1993,1,1"
 OUTPUT @Hp4291;"SYST:DATE?"
 ENTER @Hp4291;A,B,C

SYSTem:DATE:MODE {MDY|DMY}

SYSTem:DATE:MODE {MDY|DMY}

Changes the displayed date to the "month:day:year" format or the "day:month:year" format.

Parameter	Description
MDY	"month:day:year" format. (MonDayYear under (System))
DMY	"day:month:year" format. (DayMonYear under (System))

Query response is {MDY|DMY}.

Example OUTPUT @Hp4291;"SYST:DATE:MODE DMY"
 OUTPUT @Hp4291;"SYST:DATE:MODE?"
 ENTER @Hp4291;A\$

SYSTem:ERRor?

SYSTem:ERRor?

Outputs the error message in the error queue.

Query response is <numeric (Error number)> <string (Error message)>.

Example DIM A\$[50]
 OUTPUT @Hp4291;"SYST:ERR?"
 ENTER @Hp4291;A,A\$

SYSTem:FIXTure {NONE|HP16191|HP16192|HP16193|HP16194|HP16453|HP16454S|HP16454L|UDEFined}

SYSTem:FIXTure {NONE|HP16191|HP16192|HP16193|HP16194|HP16453|HP16454S|HP16454L|UDEFined}

Specifies the fixture in use in order to select which electrical length (recorded in the analyzer) is to be used. (FIXTURE [] under (Meas))

SYSTem:FIXTure

Parameter	Description
NONE	No test fixture(FIXTURE: NONE)
HP 16191	16191A (16191)
HP 16192	16192A (16192)
HP 16193	16193A (16193)
HP 16194	16194A (16194)
HP 16453	16453A (PERMITTVTY 16453)
HP 16454S	16454A(Small) (16454(S))
HP 16454L	16454A(Large) (16454(L))
UDEFined	User-defined test fixture (USER)

Query response is {NONE|HP 16191|HP 16192|HP 16193|HP 16453|HP 16454S|HP 16454L|UDEFined}.

SYSTem:FIXTure:DISTance <numeric>

SYSTem:FIXTure:DISTance <numeric>

Sets the electrical length of the user-modified fixture. (DEFINE EXTENSION under (Meas) FIXTURE [])

Parameter	Range	Unit
<numeric>	-10 to 10 (m)	-

Query response is <numeric>.

SYSTem:FIXTure:LABel <string>

SYSTem:FIXTure:LABel <string>

Modifies the label of the user-modified fixture. (LABEL FIXTURE under (Meas) FIXTURE [])

SYSTem:FIXTure:SAVE

SYSTem:FIXTure:SAVE

Saves the settings of the user-modified fixture. (SAVE USER FXTR KIT under (Meas) FIXTURE [] ;No query)

SYSTem:KEY <numeric>

SYSTem:KEY <numeric>

Sends the key code for a key or a softkey on the front panel. This is equivalent to actually pressing a key. Figure 11-1 shows the key codes of the front panel keys when using the SYSTem:KEY command.

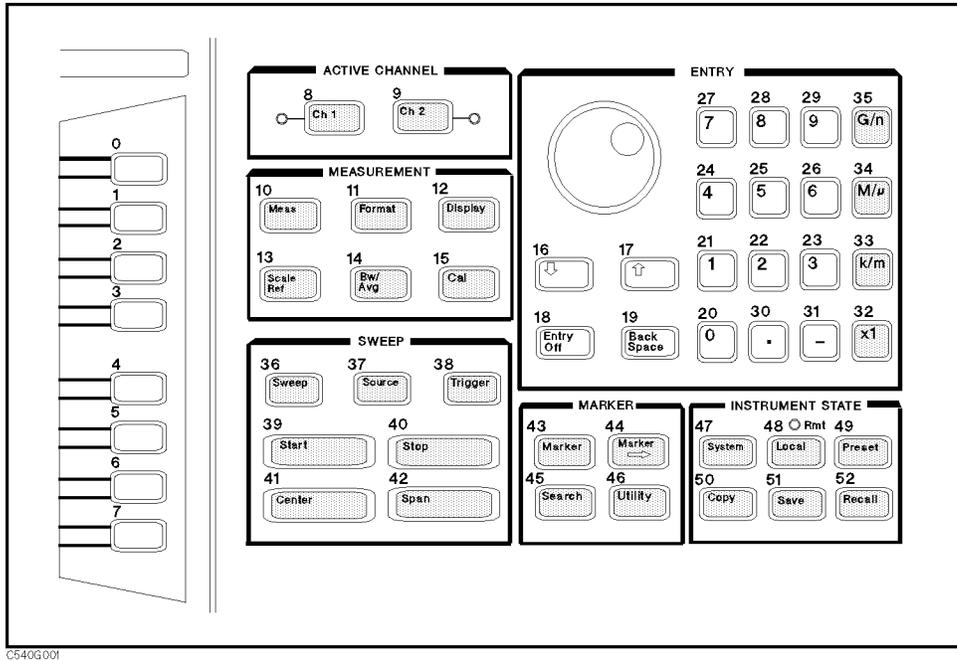


Figure 11-1. Key Codes

Do not use multiple commands in a program line (for example, OUTPUT @Hp4291;"SYST:KEY 0;KEY 1"). If you do, the analyzer's operation cannot keep up with the program.

Parameter	Description
<numeric>	Key code, 0 to 52

Query response is <numeric>. (The last key operation)

Example LOCAL @Hp4291
OUTPUT @Hp4291;"SYST:KEY 0"

SYSTem:KLOCK {OFF|ON|0|1}

SYSTem:KLOCK {OFF|ON|0|1}

Locks the front panel keys and the rotary knob.

Parameter	Description
OFF or 0	Unlock the front panel keys and rotary knob.
ON or 1	Lock the front panel keys and rotary knob.

Query response is {0|1}.

Example OUTPUT @Hp4291;"SYST:KLOCK ON"
OUTPUT @Hp4291;"SYST:KLOCK?"
ENTER @Hp4291;A

SYSTem:PRESet

SYSTem:PRESet

Presets the analyzer to the preset default values. See Appendix C of the *Function Reference* for the default values. This command does *not* preset the HP instrument BASIC. (**PRESET**; No query)

SYSTem:SECurity[:STATe] {ON|1}

SYSTem:SECurity[:STATe] {ON|1}

Blanks the displayed frequency notation for security purposes. Frequency notation cannot be restored except by sending the :SYSTem:PRESet or *RST command, or by turning the power OFF and ON. (**FREQUENCY BLANK** under **Display**)

Parameter	Description
{ON 1}	Blanks the displayed frequency notation.

Query response is {0|1}.

Related Command DISPlay:ANNotation:FREQency

Example To blank the frequency notation,

```
OUTPUT @Hp4291;"DISP:ANN:FREQ OFF"
OUTPUT @Hp4291;"SYST:SEC ON"
```

Query,

```
OUTPUT @Hp4291;"SYST:SEC?"
ENTER @Hp4291;A
```

SYSTem:TIME <numeric(hour)>,<numeric(min)>,<numeric(sec)>

SYSTem:TIME <numeric(hour)>,<numeric(min)>,<numeric(sec)>

Sets the time of the internal clock. (**TIME HH:MM:SS** under **System** **SET CLOCK**)

Parameter	Description
<numeric(hour)>	Hour, 0 to 23
<numeric(min)>	Minute, 0 to 59
<numeric(sec)>	Second, 0 to 59

Query response is <numeric(hour)>,<numeric(min)>,<numeric(sec)>.

Example OUTPUT @Hp4291;"SYST:TIME 10,30,0"

```
OUTPUT @Hp4291;"SYST:TIME?"
ENTER @Hp4291;A,B,C
```

SYSTem:VERSion?

SYSTem:VERSion?

Queries the SCPI version to which the analyzer conforms. (Query only)

Query response is *<string>*. (For example, 1993.0)

Example

```
DIM A$[10]
OUTPUT @Hp4291;"SYST:VERS?"
ENTER @Hp4291;A$
```

TMARG <numeric>

(*Simple Command*) See “Simple Commands” later in this chapter.

TRACe Subsystem**TRACe:COPY TR{2-17},TR1**

TRACe:COPY TR{2-17},TR1

Copies the data trace into the memory trace of the active channel. (DATA→MEMORY under **Display**); No query)

Parameter	Description
TR{2-17}	Memory trace; copy destination

TRACe:COPY TR{18-21},TR{1-17}

TRACe:COPY TR{18-21},TR{1-17}

Copies the data or the memory trace of the active channel into the user trace. (DATA →USR, MEM →USR under **Display**); No query)

Parameter	Description
TR{1-17}	data trace (TR1) or memory trace (TR{2-17}); copy source
TR{18-21}	user trace ; copy destination

TRACe[:DATA] <trace>,{<block>|<numeric11>,<numeric12>,...,<numeric n2>}

TRACe[:DATA] <trace>,{<block>|<numeric11>,<numeric22>,...,<numeric n1>,<numeric n2>}

Inputs data to data/memory trace arrays.

Parameter	Description
<trace>	DTR, DTRCH{1 2}, MTR, MTRCH{1 2}, TR{1-17}, TR{1-17}CH{1 2}

Where,

DTR	: Data trace of the active channel.
DTRCH{1 2}	: Data trace of the channel{1 2}.
MTR	: Selected memory trace of the active channel.
MTRCH{1 2}	: Selected memory trace of the channel{1 2}.
TR1	: Data trace of the active channel.
TR{2-17}	: Memory trace of the active channel.
TR1CH{1 2}	: Data trace of the channel{1 2}.
TR{2-17}CH{1 2}	: memory trace of the channel{1 2}.

Parameter	Description
<block>	Binary data transfer format.
<numeric1>, ... ,<numeric n2>	ASCII data transfer format.

The array dimension is defined as,
n (the number of measurement points) × 2 (value-auxiliary value).

Query response is {<block>|<numeric1>,<numeric12>, ... ,<numeric n2>}.

Example To get the data trace array,

```
DIM A(1:201,1:2)
OUTPUT @Hp4291;"TRAC? DTR,"
ENTER @HP4291;A(*)
```

TRACe[:DATA] {TRX{18-21}|TRY{18-21}},{<block>|<numeric1>,<numeric2>, ... ,<numeric n>}

TRACe[:DATA] {TRX{18-21}|TRY{18-21}},{<block>|<numeric1>,<numeric2>, ... ,<numeric n>}

Inputs data to user traces.

Parameter	Description
TRX{18-21}	X-axis data of user trace{1-4}
TRY{18-21}	Y-axis data of the user trace{1-4}

Parameter	Description
<block>	Binary data transfer format.
<numeric1>, ... ,<numeric n>	ASCII data transfer format.

The array dimension is defined as,
n (the number of points) × 1

Query response is {<block>|<numeric1>,<numeric2>, ... ,<numeric n>}.

Related Command TRACe:POINts

Example DIM A(1:201)
OUTPUT @Hp4291;"TRAC TRX18,";A(*)

TRACe[:DATA]:VALue? <trace>,<numeric>

TRACe[:DATA]:VALue? <trace>,<numeric>

Outputs the each trace value at a specified point. (Query only)

Parameter	Description
<trace>	DTR, DTRCH{1 2}, MTR, MTRCH{1 2}, TR{1-17}, TR{1-17}CH{1 2}

TRACe[:DATA]:VALue?

Where,

DTR	: Data trace of the active channel.
DTRCH{1 2}	: Data trace of the channel{1 2}.
MTR	: Memory trace of the active channel.
MTRCH{1 2}	: Memory trace of the channel{1 2}.
TR1	: Data trace of the active channel.
TR{2-17}	: Memory trace of the active channel.
TR1CH{1 2}	: Data trace of the channel{1 2}.
TR{2-17}CH{1 2}	: Memory trace of the channel{1 2}.

Parameter	Description
<numeric>	The measurement point.

Query response is <numeric1>,<numeric2>.

Where,

<numeric1>	: Trace value.
<numeric2>	: Auxiliary trace value.

TRACe:POINts TR{18-21}[,<numeric>]

TRACe:POINts TR{18-21}[,<numeric>]

Defines the number of points of the user trace.

TRIGger Subsystem

The TRIGger subsystem is used to control the trigger event of the trigger system. For more information about triggering, see the *Programming Manual*.

Related Command INITiate:CONTinuous
 INITiate[:IMMediate]
 INITiate[:IMMediate]:AGain:ALL

TRIGger:EVENT:TYPE {POINT|SWEep}

TRIGger:EVENT:TYPE {POINT|SWEep}

Selects the trigger event mode. (TRIG EVENT [] under (Trigger))

Parameter	Description
POINT	Trigger event on point ¹
SWEep	Trigger event on sweep

Query Response is {POINT|SWE}.

¹ Available only when the trigger source is GPIB, Manual, or External trigger.

Example OUTPUT @Hp4291;"TRIG:EVENT:TYPE SWE"
 OUTPUT @Hp4291;"TRIG:EVENT:TYPE?"
 ENTER @Hp4291;A\$

TRIGger:SLOPe {POSitive|NEGative}

TRIGger:SLOPe {POSitive|NEGative}

Sets the trigger signal polarity of an external signal connected to the rear panel EXT TRIGGER input. (TRIG PLRTY pos neg under (Trigger))

Parameter	Description
POSitive	Positive trigger (low-to-high transition)
NEGative	Negative trigger (high-to-low transition)

Query Response is {POS|NEG}.

Example OUTPUT @Hp4291;"TRIG:SLOP POS"
 OUTPUT @Hp4291;"TRIG:SLOP?"
 ENTER @Hp4291;A\$

TRIGger:SOURce {BUS|EXTernal|INTernal|MANual}

TRIGger:SOURce {BUS|EXTernal|INTernal|MANual}

Selects the trigger source, which is common to both channels. (TRIGGER: [] under (Trigger))

TRIGger:SOURce

Parameter	Description
BUS	GPIB trigger (*TRG)
EXTernal	External trigger input from EXT TRIGGER input on the rear panel
INTernal	Internal trigger
MANual	Manual trigger from front panel key

Query Response is {BUS|EXT|INT|MAN}.

USKEY

(*Simple Command*) See “Simple Commands” later in this chapter.

Common Commands

*CLS

*CLS

Clears the Status Byte Register, and the Operation Event Status Register, the Standard Event Status Register, and the Instrument Event Status Register. (No query)

Example OUTPUT @Hp4291;"*CLS"

*ESE <numeric>

*ESE <numeric>

Sets the enable bits of the Standard Event Status Register.

Parameter	Description
<numeric>	0 to 255 (decimal expression of enable bits of the operation status register)

Query response is <numeric>.

Example OUTPUT @Hp4291;"*ESE 1"
 OUTPUT @Hp4291;"*ESE?"
 ENTER @Hp4291;A

*ESR?

*ESR?

Returns the contents of the Standard Event Status Register. (Query only)

Query response is <numeric>.

Example OUTPUT @Hp4291;"*ESR?"
 ENTER @Hp4291;A

*IDN?

*IDN?

Returns a string that represents the analyzer's ID.

Common Commands

Query response is *<manufacturer>*,*<model>*,*<serial no.>*,*<firmware rev.>*.

Where,

<manufacturer> HEWLETT-PACKARD.
<model> 4291A.
<serial no.> Serial Number, like JP1KA00101.
<firmware rev.> Firmware revision, like REV1.00.

Example OUTPUT @Hp4291;"*IDN?"
 ENTER @Hp4291;A\$

*OPC

*OPC

Sets the Operation Complete bit (bit 0 of the Standard Event Status Register) to 1 when it completes all pending operations.

*OPC? query returns an ASCII character 1 when all pending operations have been completed.

Query response is 1.

Example OUTPUT @Hp4291;"*OPC"
 OUTPUT @Hp4291;"*OPC?"
 ENTER @Hp4291;A

*OPT?

*OPT?

Queries the options installed. (Query only)

Query response is *<option>*s.

Where *<option>*s are,

001 Add dc bias.

Example OUTPUT @Hp4291;"*OPT?"
 ENTER @Hp4291;A\$

*PCB <numeric>

*PCB <numeric>

Specifies the address of a controller that is temporarily passing GPIB control to the analyzer.
 (No query)

Parameter	Description
<i><numeric></i>	0 to 30

Example OUTPUT @Hp4291;"*PCB 0"

***RST**

*RST

Resets the analyzer to its default values, (see the *Function Reference* for information on the default values), stops sweeping and taking data, and resets the HP instrument BASIC . (No query)

***SRE <numeric>**

*SRE <numeric>

Sets the contents of the Status Byte Enable Register.

Parameter	Description
<numeric>	0 to 255 (decimal expression of the contents of the Service Request Enable Register)

Query response is <numeric>.

Example OUTPUT @Hp4291;"*SRE 1"
 OUTPUT @Hp4291;"*SRE?"
 ENTER @Hp4291;A

***STB?**

*STB?

Returns the contents of the Status Byte Register. The content of the register's bit 6 is the master summary status bit. (Query only)

Query response is <numeric>.

Example OUTPUT @Hp4291;"*STB?"
 ENTER @Hp4291;A

***TRG**

*TRG

Triggers the analyzer when the trigger mode is set to BUS trigger. (No query)

Example To trigger a measurement,
 OUTPUT @Hp4291;"INIT:CONT ON"
 OUTPUT @Hp4291;"TRIG:SOUR BUS"
 OUTPUT @Hp4291;"*TRG"

***TST?**

*TST?

Executes an internal self-test and returns the test result. (Query only)

Query response is {0|1}.

Where,

0	: Pass
1	: Fail

Common Commands

Example OUTPUT @Hp4291;"*TST?"
 ENTER @Hp4291;A

*WAI

*WAI

Makes the analyzer wait until all previously sent commands are completed. (No query)

Simple Commands

BLIGHT {OFF|ON|0|1}

BLIGHT {OFF|ON|0|1}

Sets backlighting the LCD ON or OFF.

Parameter	Description
OFF or 0	Backlighting OFF
ON or 1	Backlighting ON

Query response is {0|1}.

```
Example          OUTPUT @Hp4291;"BLIGHT ON"
                  OUTPUT @Hp4291;"BLIGHT?"
                  ENTER @Hp4291;A
```

CENT <numeric>

CENT <numeric>

Defines the center value of the sweep range, or the center value of the segment to be edited in the list sweep table. (Center), or CENTER under (Sweep)

Parameter	Range	Unit
<numeric>	1000000(=1 M) to 1.8×10^9 (=1.8 G)	Hz (Frequency sweep)
	0.2×10^{-3} to 1	V (OSC level sweep)
	-40 to 40	V (DC-V sweep)
	-100×10^{-3} to 100×10^{-3}	A (DC-I sweep)

Query response is <numeric>.

```
Example          OUTPUT @Hp4291;"CENT 899.95MAHZ"
                  OUTPUT @Hp4291;"CENT?"
                  ENTER @Hp4291;A
```

CHAN1

CHAN1

Selects the channel 1 as the active channel. (Chan 1)

Query response is {0|1}.

Where, 0 : Channel 1 is not the active.
 1 : Channel 1 is the active.

CHAN2

CHAN2

Selects the channel 2 as the active channel. (**Chan 2**)

Query response is {0|1}.

Where, 0 : Channel 2 is not the active.
 1 : Channel 2 is the active.

CLEM

CLEM

Clears all memory traces. (No query)

CLOSE

CLOSE

Returns a file, which has been read/write-enabled using the ROPEN command or WOPEN command, to access-disabled status. If this command is executed before reading process using the READ? command completes, an error occurs.

Generally, this command is used in combination with the ROPEN command and READ? command or the WOPEN command and the WRITE command, as shown in Figure 11-3. (No query)

CONT

CONT

Initiate the trigger system continuously. (**CONTINUOUS** under **Trig**)

For more information about trigger system, see *Programming Manual*.

Query response is {0|1}.

Where, 0 : After measurement, the trigger sequence moves to the idle state.
 1 : After measurement, the trigger sequence moves to the wait-for-trigger state (continuously initiated).

CWD?

CWD?

Returns the name of the current directory. (Query only)

■ Query Response

{*string*} <new line><END>

DATMEM

DATMEM

Stores the current active measurement data in the selected memory of the active channel.

(**DATA—MEMORY** under **Display**); No query)

DFLT

DFLT

Sets the printing parameters to their default values. (DEFAULT SETUP under **Copy**); No query)

The table below lists the default values.

Command	Default value	Unit
DPI	75	dpi
FORMFEED	ON	
LANDSCAPE	OFF	
LMARG	1.0	inch
TMARG	1.0	inch
SKEY	OFF	

DISP {DATA|MEMO|DATM}

DISP {DATA|MEMO|DATM}

Selects the display trace type. (DISPLAY: DATA, MEMORY, DATA and MEMORY under **Display**)

Parameter	Description
DATA	Current data trace
MEMO	Memory trace
DATM	Current data and memory traces

Query response is {DATA|MEMO|DATM}.

Example OUTPUT @Hp4291;"DISP DATA"
 OUTPUT @Hp4291;"DISP?"
 ENTER @Hp4291;A\$

DISSMEMO {OFF|ON|0|1}

DISSMEMO {OFF|ON|0|1}

Sets the state of the active memory trace. (SEL'D MEM On off under **Display**)

Parameter	Description
OFF or 0	Erases the active memory trace.
ON or 1	Displays the active memory trace.

Query response is {0|1}.

Example OUTPUT @Hp4291;"DISSMEMO ON"
 OUTPUT @Hp4291;"DISSMEMO?"
 ENTER @Hp4291;A

DPI <numeric>

DPI <numeric>

Specifies the resolution of a printer used for printing by dpi. (DPI under Copy)

Parameter	Range	Unit
<numeric>	75 to 600	DPI(Dots Per Inch)

Query response is <numeric>.

```
Example          OUTPUT @Hp4291;"DPI 100"
                  OUTPUT @Hp4291;"DPI?"
                  ENTER @Hp4291;A
```

DUAC {OFF|ON|0|1}

DUAC {OFF|ON|0|1}

Selects the display of both measurement channels or the active channel only. (DUAL CHAN ON off under Display)

Parameter	Description
OFF or 0	Active channel only
ON or 1	Both channels

Query response is {0|1}.

```
Example          OUTPUT @Hp4291;"DUAC ON"
                  OUTPUT @Hp4291;"DUAC?"
                  ENTER @Hp4291;A
```

DUAM {IMPH|IRIM|AMPH|ARIM|LSQ|LPQ|CSD|CPD|DRLF|DRLT|DLFLT|DMLT|PRLF|PRCLT|PLFLT|PMLT}

DUAM {IMPH|IRIM|AMPH|ARIM|LSQ|LPQ|CSD|CPD|DRLF|DRLT|DLFLT|DMLT|PRLF|PRCLT|PLFLT|PMLT}

Selects parameters to be measure for both channel. (No query)

Parameter	Description
IMPH	Measure $ Z $ on channel 1, θ on channel 2 ($ Z -\theta$ under (Meas))
IRIM	Measure R on channel 1, X on channel 2 (R-X under (Meas))
AMPH	Measure $ Y $ on channel 1, θ on channel 2 ($ Y -\theta$ under (Meas))
ARIM	Measure G on channel 1, B on channel 2 (G-B under (Meas))
LSQ	Measure Ls on channel 1, Q on channel 2 (LS-Q under (Meas))
LPQ	Measure Lp on channel 1, Q on channel 2 (LP-Q under (Meas))
CSD	Measure Cs on channel 1, D on channel 2 (CS-D under (Meas))
CPD	Measure Cp on channel 1, D on channel 2 (CP-D under (Meas))
DRLF	Measure ε_r' on channel 1, ε_r'' on channel 2 ($\varepsilon_r'-\varepsilon_r''$ under (Meas))
DRLT	Measure ε_r' on channel 1, $\tan\delta$ on channel 2 ($\varepsilon_r'-\tan\delta$ under (Meas))
DLFLT	Measure ε_r'' on channel 1, $\tan\delta$ on channel 2 ($\varepsilon_r''-\tan\delta$ under (Meas))
DMLT	Measure $ \varepsilon_r $ on channel 1, $\tan\delta$ on channel 2 ($ \varepsilon_r -\tan\delta$ under (Meas))
PRLF	Measure μ_r' on channel 1, μ_r'' on channel 2 ($\mu_r'-\mu_r''$ under (Meas))
PRCLT	Measure μ_r' on channel 1, $\tan\delta$ on channel 2 ($\mu_r'-\tan\delta$ under (Meas))
PLFLT	Measure μ_r'' on channel 1, $\tan\delta$ on channel 2 ($\mu_r''-\tan\delta$ under (Meas))
PMLT	Measure $ \mu_r $ on channel 1, $\tan\delta$ on channel 2 ($ \mu_r -\tan\delta$ under (Meas))

Example OUTPUT @Hp4291;"DUAM IMPH"

FMT {LINY|LOGY|POL|SMIT|ADM|COMP}

FMT {LINY|LOGY|POL|SMIT|ADM|COMP}

Selects the display format. (LIN Y-AXIS , LOG Y-AXIS , POLAR CHART , SMITH CHART , ADMITTANCE CHART , COMPLEX PLANE under (Format))

Parameter	Description
LINY	Linear Y-axis (LIN Y-AXIS)
LOGY	Logarithmic Y-axis (LOG Y-AXIS)
POL	Polar chart (POLAR CHART)
SMIT	Smith chart (SMITH CHART)
ADM	Admittance chart (ADMITTANCE CHART)
COMP	Complex plane (COMPLEX PLANE)

Query response is {LINY|LOGY|POL|SMIT|ADM|COMP}.

Example OUTPUT @Hp4291;"FMT LOGY"

 OUTPUT @Hp4291;"FMT?"

 ENTER @Hp4291;A\$

FNAME? <numeric>

FNAME? <numeric>

Returns the file name corresponding to a specified number in the current directory. To each file, a number is assigned from 1 to “the number of the files” in alphabetical order. Use the FNUM? command to verify the number of the files in the current directory. (Query only)

Parameter	Description	Range
<numeric>	Specified file No.	1 to “the number of the files in the current directory”

■ Query Response

{string} <new line><^END>

FNUM?

FNUM?

Returns the number of the files in the current directory. (Query only)

■ Query Response

{numeric} <new line><^END>

FORMFEED {OFF|ON|0|1}

FORMFEED {OFF|ON|0|1}

Sets the printer ON or OFF for delivering printed paper each time printing an entire screen is finished. When the paper orientation is set to Landscape, the setting by this FORMFEED command will not take effect and the printer delivers printed paper screen by screen.

(FORMFEED ON off under **Copy**)

Parameter	Description
OFF or 0	Does not deliver printed paper
ON or 1	Delivers printed paper

Query response is {0|1}.

Example OUTPUT @Hp4291;"FORMFEED ON"
 OUTPUT @Hp4291;"FORMFEED?"
 ENTER @Hp4291;A

FREO

FREO

Blanks the displayed frequency notation for security purposes. Frequency notation cannot be restored except by sending the :SYSTem:PRESet or *RST command, or by turning the power OFF and ON. (FREQUENCY BLANK under **Display**)

Query response is {0|1}.

Where, 0 : FREO command was not issued.
 1 : FREO command was issued.

FSIZE? <string>

FSIZE? <string>

Returns the size of a specified file in bytes. If the file does not exist, this command returns -1.
(Query only)

Parameter	Description
<string>	File name of up to 12 characters including its extension (for the LIF format, up to 10 characters)

■ Query Response

{*numeric*} <new line><END>

HOLD

HOLD

Freezes the data trace on the display. the analyzer stops sweeping and taking data.

(SWEEP: HOLD under (Trigger))

Query response is {0|1}.

Where, 0 : Sweeping (not hold mode)
 1 : Hold mode

LANDSCAPE {OFF|ON|0|1}

LANDSCAPE {OFF|ON|0|1}

Sets the orientation of paper landscape or not, using ON or OFF. This setting takes effect for printers which support printing paper placed in the landscape orientation. Setting the paper orientation mode will invalidate the setting by the FORMFEED command.

(ORIENT [PORTRAIT] under (Copy))

Parameter	Description
OFF or 0	The orientation of paper is not set to Landscape. (Thus, Portrait)
ON or 1	The orientation of paper is set to Landscape.

Query response is {0|1}.

Example OUTPUT @Hp4291;"LANDSCAPE ON"
 OUTPUT @Hp4291;"LANDSCAPE?"
 ENTER @Hp4291;A

LMARG <numeric>

LMARG <numeric>

Specifies the left margin of printing by inch. (LFT MARGIN under (Copy))

Parameter	Range	Unit
<numeric>	0 to 5	inch

Query response is <numeric>.

Simple Commands

Example OUTPUT @Hp4291;"LMARG 5"
 OUTPUT @Hp4291;"LMARG?"
 ENTER @Hp4291;A

LISD {FBAS|OBAS}

LISD {FBAS|OBAS}

Selects the frequency base mode or order base mode in the frequency list sweep.

Parameter	Description
FBAS	frequency base mode
OBAS	order base mode

Query response is {FBAS|OBAS}.

Example OUTPUT @Hp4291;"LISD FBAS"
 OUTPUT @Hp4291;"LISD?"
 ENTER @Hp4291;A\$

MARDCENT

MARDCENT

Sets the stimulus parameter center value to the difference value between the marker and the Δ marker values. (MKR Δ →CENTER under (Marker→); No query)

MARDSPAN

MARDSPAN

Sets the SPAN to the difference between the marker and the Δ marker values. (MKR Δ →CENTER under (Marker→); No query)

MARK {OFF|ON|0|1}

MARK {OFF|ON|0|1}

Sets the marker to active (ON) or inactive (OFF). When the marker is turned off, the marker, sub-marker, and Δ marker are tuned to be off. ((MKR))

Parameter	Description
OFF or 0	Turns off the marker function.
ON or 1	Turns on the marker function.

Query response is {0|1}.

Example OUTPUT @Hp4291;"MARK ON"
 OUTPUT @Hp4291;"MARK?"
 ENTER @Hp4291;A

MARKCENT

MARKCENT

Sets the stimulus parameter center value to the stimulus parameter value of the marker.

(**MKR—CENTER** under **(Marker→)**; No query)

MARKREF

MARKREF

Makes the reference value equal to the marker's absolute value (regardless of the Δ marker value).

(**MKR—REFERENCE** under **(Scale Ref)** and **(Marker→)**; No query)

MARKSTAR

MARKSTAR

Sets the stimulus parameter start value to the stimulus value of the marker.

(**SEGMENT: MKR—START** under **(Sweep)**, or **MKR—START** under **(Marker→)**; No query)

MARKSTOP

MARKSTOP

Sets the stimulus parameter stop value to the stimulus value of the marker. (**MKR—STOP** under

(Sweep), or **MKR—STOP** under **(Marker→)**; No query)

MARZ

MARZ

Sets the stimulus parameter center value to the stimulus parameter value of the marker, and

changes the stimulus span value to "stimulus span \times zooming aperture." (**MKR ZOOM** under **(Marker→)**; No query)

MATH {**DATA|DMNM|DPLM|DDVM|DMLM**}

MATH {DATA|DMNM|DPLM|DDVM|DMLM}

Sets the data math operation. (**DATA MATH []** under **(Display)**)

Parameter	Description
DATA	Turns OFF all data math functions. (DATA MATH: DATA)
DMNM	Subtracts the memory trace from the data trace. (DATA-MEM)
DPLM	Adds the memory trace to the data trace. (DATA+MEM)
DDVM	Divides the data by the memory. (DATA/MEM)
DMLM	Multiplies the data trace by the memory trace. (DATA*MEM)

Query response is {DATA|DMNM|DPLM|DDVM|DMLM}.

```
Example          OUTPUT @Hp4291;"MATH DATA"
                  OUTPUT @Hp4291;"MATH?"
                  ENTER @Hp4291;A$
```

MEAS {IMAG|IPH|IRE|IIM|AMAG|APH|ARE|AIM|RCM|RCPH|RCR|RCIM|DCR|DCLF|DCLT|DCM|PRE|PLF|PLT|PMAG|CP|CS|LP|LS|D|Q|RP|RS}
 MEAS {IMAG|IPH|IRE|IIM|AMAG|APH|ARE|AIM|RCM|RCPH|RCR|RCIM|DCR|DCLF|DCLT|DCM|PRE|PLF|PLT|PMAG|CP|CS|LP|LS|D|Q|RP|RS}

Selects the parameters to be measured. (under **Meas**)

Parameter	Description
IMAG	Absolute magnitude value of impedance (IMPEDANCE: MAG(Z))
IPH	Phase value of impedance (PHASE(θ_z))
IRE	Resistance value (RESIST(R))
IIM	Reactance value (REACT(X))
AMAG	Absolute magnitude value of admittance (ADMITTANCE: MAG(Y))
APH	Phase value of admittance (PHASE(θ_y))
ARE	Conductance value (CONDUCT(G))
AIM	Suceptance value (SUSCEPT(B))
RCM	Absolute magnitude value of reflection coefficient (REFL. COEF: MAG(Γ))
RCPH	Phase value of reflection coefficient (PHASE(θ_Γ))
RCR	Real part of reflection coefficient (REAL(Γ_x))
RCIM	Imaginary part of reflection coefficient (IMAG(Γ_y))
DCR	Permittivity (Dielectric constant; PERMITTIVITY: REAL(ϵ_r'))
DCLF	Dielectric loss factor (LOSS FACTOR (ϵ_r''))
DCLT	Dielectric loss tangent (LOSS TANGENT ($\tan\delta$))
DCM	Absolute magnitude value of complex permittivity (MAG(ϵ_r))
PRE	Permeability (PERMEABILITY: REAL(μ_r'))
PLF	Loss factor of permeability (LOSS FACTOR(μ_r''))
PLT	Loss tangent of permeability (LOSS TANGENT($\tan\delta$))
PMAG	Absolute magnitude value of complex permeability (MAG(μ_r))
CP	Equivalent parallel capacitance (CAPCITANCE: PRL(C_p))
CS	Equivalent series capacitance (SER(C_s))
LP	Equivalent parallel inductance (INDUCTANCE: PRL(L_p))
LS	Equivalent series inductance (SER(L_s))
D	Dissipation factor (D FACTOR (D))
Q	Quality factor (Q FACTOR (Q))
RP	Equivalent parallel resistance (PRL(R_p))
RS	Equivalent series resistance (SER(R_s))

Query response is {IMAG|IPH|IRE|IIM|AMAG|APH|ARE|AIM|RCM|RCPH|RCR|RCIM|DCR|DCLF|DCLT|DCM|PRE|PLF|PLT|PMAG|CP|CS|LP|LS|D|Q|RP|RS}.

```
Example      OUTPUT @Hp4291;"MEAS IMAG"
             OUTPUT @Hp4291;"MEAS?"
             ENTER @Hp4291;A$
```

NUMG <numeric>

NUMG <numeric>

Triggers a user-specified number of sweeps and returns to the HOLD mode.

(NUMBER OF GROUPS under **Trigger**); No query)

Parameter	Description
<numeric>	Greater than 0 (if <numeric> is 0 or less than 0, it is set to 1.)

Example OUTPUT @Hp4291;"NUMG 10"

PEAKCENT

PEAKCENT

Searches for a peak using the marker and then changes the center to the stimulus parameter value of that peak. (PEAK→CENTER under **Marker→**); No query)

POIDTIME <numeric>

POIDTIME <numeric>

Sets the point delay time. (POINT DELAY TIME under **Sweep**)

Parameter	Range	Unit
<numeric>	0 to 3600 (resolution : 12.5 μ)	s

Query response is <numeric>.

PRSOFT {OFF|ON|0|1}

PRSOFT {OFF|ON|0|1}

Sets printing the softkeys displayed in the screen ON or OFF. (COPY SKEY on OFF under **Copy**)

Parameter	Description
OFF or 0	Does not print the soft keys
ON or 1	Print the soft keys

Query response is {0|1}.

Example OUTPUT @Hp4291;"PRSOFT ON"
 OUTPUT @Hp4291;"PRSOFT?"
 ENTER @Hp4291;A

READ?

READ?

Reads data from a file that has been read-enabled using the ROPEN command. The returned data is in the fixed length block format defined in IEEE488.2. The fixed length block format, as shown in Figure 11-2, consists of a header part indicating the data size and an actual data part. In the case of the 4291B, the number of digits to indicate the data size is 6 and the maximum length of the actual data part is 16 Kbytes. If a file contains data greater than 16 Kbytes, execute this command repeatedly to read it. Note that acceptable file formats for this command are the DOS format and the LIF format BDAT type.

Generally, this command is used in combination with the ROPEN command and the CLOSE command, as shown in Figure 11-3. (Query only)

■ Query Response

{*block*} <new line><END>

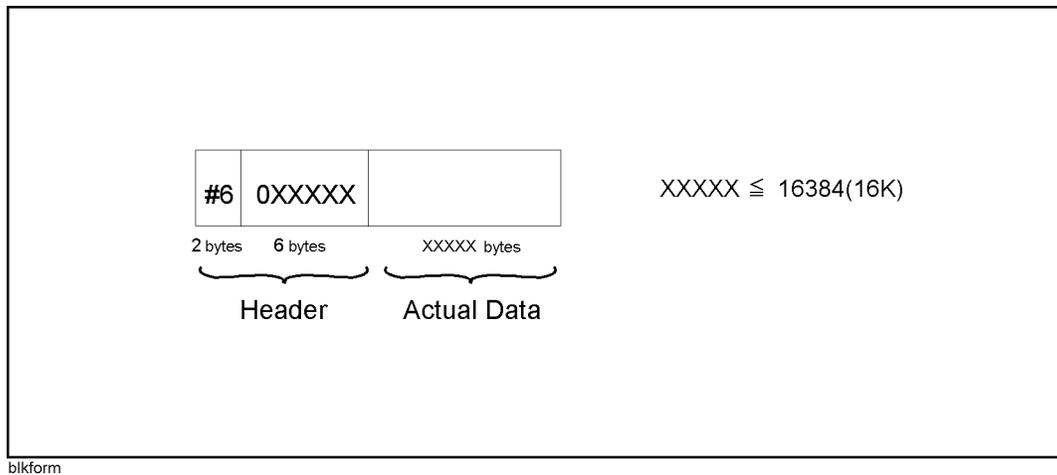


Figure 11-2. Fixed length block format

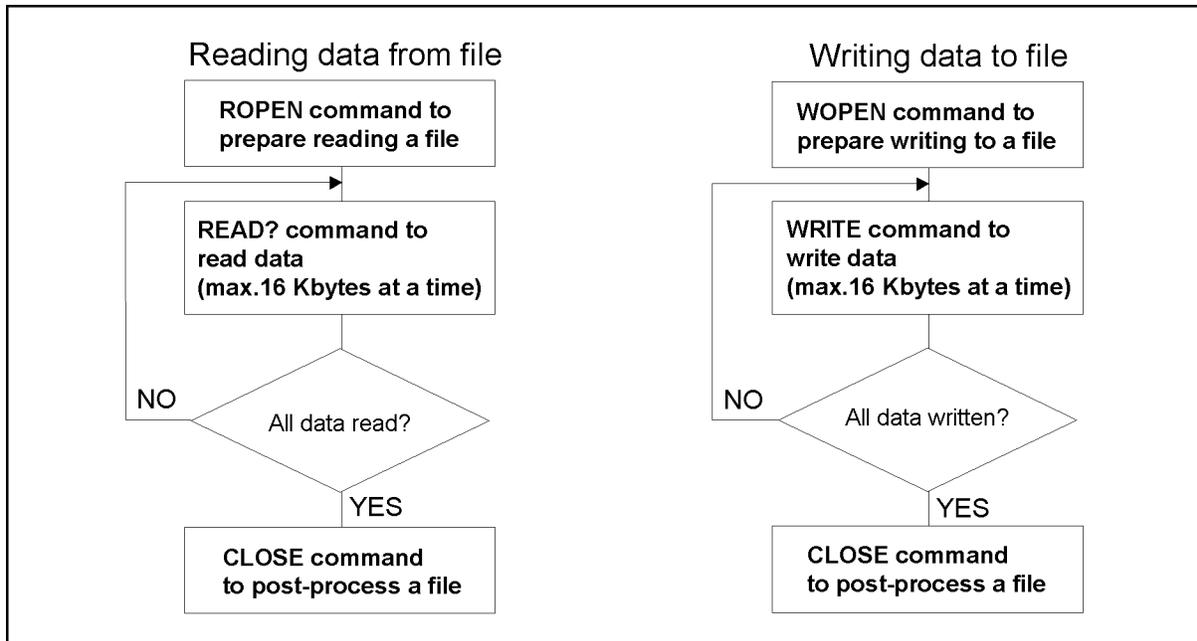
ROPEN <string>

ROPEN <string>

Makes a specified file read-enabled. If the file does not exist, an error occurs.

Generally, this command is used in combination with the READ? command and the CLOSE command, as shown in Figure 11-3. (No query)

Parameter	Description
<i><string></i>	File name of up to 12 characters including its extension (for the LIF format, up to 10 characters)



rwflow

Figure 11-3. Procedure of executing commands to read/write data

RESTMDISK

RESTMDISK

Restore the data from the backup to the memory disk.

Example OUTPUT @Hp4291;"RESTMDISK"
 OUTPUT @Hp4291

SAVDSTAC <string>

SAVDSTAC <string>

Saves the instrument state in the format so that the 4291A can recall. (4291A STATE under Save)

Example OUTPUT @Hp4291;"SAVDSTAC ""FIL_NAME"""

SAVDTIF <string>

SAVDTIF <string>

Saves the graphic image on the screen as an TIFF file.(GRAPHICS under Save)

Example OUTPUT @Hp4291;"SAVDTIF ""FIL_NAME"""

SELM <numeric>

SELM <numeric>

Selects the active memory trace. (SELECT MEMORY NO under **Display**)

Parameter	Range	Unit
<numeric>	1 to 18 (maximum number of the memory trace depends on number of points)	—

Query response is <numeric>.

SING

SING

Makes one sweep of the data and returns to the hold mode. (SINGLE under **Trigger**); No query)**SPAN** <numeric>

SPAN <numeric>

Sets the span of the stimulus range or the segment. ((SPAN) or SPAN under **Sweep**)

Parameter	Range	Unit
<numeric>	0 to 1.799×10^9 (= 1.799 G)	Hz (frequency sweep)
	0 to 0.9998×10^{-6} (= 999.8 mV)	V (OSC level sweep)
	0 to 80	V (DC-V sweep)
	0 to 2.0×10^{-5} (= 200mA)	A (DC-I sweep)

Query response is <numeric>.

```
Example      OUTPUT @Hp4291;"SPAN 1GHZ"
              OUTPUT @Hp4291;"SPAN?"
              ENTER @Hp4291;A
```

STAR <numeric>

STAR <numeric>

Sets the start value of the stimulus range or the segment. ((Start) or SEGMENT: START under **Sweep**)

Parameter	Range	Unit
<numeric>	1000000(= 1 M) to 1.8×10^9 (= 1.8 G)	Hz (Frequency sweep)
	0.2×10^{-3} to 1	V (OSC level sweep)
	-40 to 40	V (DC-V sweep)
	-100×10^{-3} to 100×10^{-3}	A (DC-I sweep)

Query response is <numeric>.

```
Example      OUTPUT @Hp4291;"STAR 100MAHZ"
              OUTPUT @Hp4291;"STAR?"
              ENTER @Hp4291;A
```

STOD{DISK|MEMO}

STOD{DISK|MEMO}

Sets the storage device. (**STOR DEV[]** under **Save**); No query;

Parameter	Description
STODDISK	Flexible disk drive
STODMEMO	Memory disk

Query response is {0|1}.

```
Example          OUTPUT @Hp4291;"STODDISK"
                  OUTPUT @Hp4291;"STODDISK?"
                  ENTER @Hp4291;A
```

STOP <numeric>

STOP <numeric>

Sets the stop value of the stimulus range or the segment. (**STOP**) or **STOP** under **Sweep**)

Parameter	Range	Unit
<numeric>	1000000(=1 M) to 1.8×10^9 (=1.8 G)	Hz (Frequency sweep)
	0.2×10^{-3} to 1	V (OSC level sweep)
	-40 to 40	V (DC-V sweep)
	-100×10^{-3} to 100×10^{-3}	A (DC-I sweep)

Query response is <numeric>.

```
Example          OUTPUT @Hp4291;"STOP 1.8GHZ"
                  OUTPUT @Hp4291;"STOP?"
                  ENTER @Hp4291;A
```

STORMDISK

STORMDISK

Backup the memory disk. (**BACK UP MEMO DISK** under **Save**)

```
Example          OUTPUT @Hp4291;"STORMDISK"
```

STYPE {LIN|LOG|LIST}

STYPE {LIN|LOG|LIST}

Selects the sweep type. (**SWEEP TYPE:LIN**, **LOG**, **LIST** under **Sweep**)

Simple Commands

Parameter	Description
LIN	Linear sweep (SWEEP TYPE:LIN)
LOG	Logarithmic sweep (LOG)
LIST	List sweep (, LIST)

Query response is {LINF|LOGF|LIST}.

```
Example          OUTPUT @Hp4291;"STYPE LIN"
                  OUTPUT @Hp4291;"STYPE?"
                  ENTER @Hp4291;A$
```

SWED {UP|DOWN}

SWED {UP|DOWN}

Toggles direction of sweep between up and down. (SWEEP DIR [] under (Sweep))

Parameter	Description
UP	Sweep direction UP.
DOWN	Sweep direction DOWN.

Query response is {UP|DOWN}.

SWEDTIME <numeric>

SWEDTIME <numeric>

Sets the sweep delay time. (SWEEP DELAY TIME under (Sweep))

Parameter	Range	Unit
<numeric>	0 to 3600 (resolution : 12.5 μ)	s

Query response is <numeric>.

SWES {FREQ|OLEV|DCV|DCI}

SWES {FREQ|OLEV|DCV|DCI}

Selects the sweep source. (SWP SRC: FREQ, OSC LEVEL, DC BIAS V, DC BIAS I under (Sweep))

Parameter	Description
FREQ	frequency sweep
OLEV	OSC level sweep
DCV	DC-V bias sweep
DCI	DC-I bias sweep

Query response is {FREQ|OLEV|DCV|DCI}.

TMARG <numeric>

TMARG <numeric>

Specifies the top margin of printing by inch. (TOP MARGIN under **Copy**)

Parameter	Range	Unit
<numeric>	0 to 5	inch

Query response is <numeric>.

Example OUTPUT @Hp4291;"TMARG 5"
 OUTPUT @Hp4291;"TMARG?"
 ENTER @Hp4291;A

USKEY

USKEY

Displays the user key labels of the soft keys. The user menu display returns to the ordinary measurement keys when the program ends.

Example OUTPUT @Hp4291;"USKEY"

WOPEN <string>[,<numeric>]

WOPEN <string>[,<numeric>]

If the specified file exists, this command makes it write-enabled; otherwise, creates a new file and makes it write-enabled. This command takes its arguments in a different way, depending on the file format. For a DOS format file you do not have to specify its file size, for a LIF format file you must. Specify the file size, 0 or greater, so that the file can contain the maximum number of bytes used. Note that only the BDAT type is available as the LIF file format.

The format and size of an existing file cannot be changed. Therefore, if you want to change them, delete the file itself using the MMEM:DEL command and then create a new file using this command.

This command is used in combination with the WRITE command and the CLOSE commands, as shown in Figure 11-3. (No query)

Parameter	Description
<string>	File name of up to 12 characters including its extension (for the LIF format, up to 10 characters)
<numeric>	File size (required only for the LIF format)

WRITE <block>

WRITE <block>

Writes data in a file that has been write-enabled using the WOPEN command. Written data must take the fixed length block format (see Figure 11-2) defined in IEEE488.2. The maximum length of data is 16 Kbytes. If data is greater than 16 Kbytes, execute this command repeatedly to write it. (No query)

Generally, this command is used in combination with the WOPEN command and the CLOSE command, as shown in Figure 11-3. (No query)

Parameter	Description
<block>	Data in the fixed length block format

Manual Changes

Introduction

This appendix contains the information required to adapt this manual to earlier versions or configurations of the analyzer than the current printing date of this manual. The information in this manual applies directly to the 4291B RF Impedance/Material Analyzer serial number prefix listed on the title page of this manual.

Manual Changes

To adapt this manual to your 4291B, see Table A-1 and Table A-2, and make all the manual changes listed opposite your instrument's serial number and firmware version.

Instruments manufactured after the printing of this manual may be different from those documented in this manual. Later instrument versions will be documented in a manual changes supplement that will accompany the manual shipped with that instrument. If your instrument's serial number is not listed on the title page of this manual or in Table A-1, it may be documented in a *yellow MANUAL CHANGES* supplement.

In additions to change information, the supplement may contain information for correcting errors (Errata) in the manual. To keep this manual as current and accurate as possible, Agilent Technologies recommends that you periodically request the latest *MANUAL CHANGES* supplement.

For information concerning serial number prefixes not listed on the title page or in the *MANUAL CHANGE* supplement, contact the nearest Agilent Technologies office.

Turn on the line switch or execute the *IDN? command by GPIB to confirm the firmware version. See the *GPIB Command Reference* manual for information on the *IDN? command.

Table A-1. Manual Changes by Serial Number

Serial Prefix or Number	Make Manual Changes

Table A-2. Manual Changes by Firmware Version

Version	Make Manual Changes
1.00, 1.01	Change 1

Serial Number

Agilent Technologies uses a two-part, ten-character serial number that is stamped on the serial number plate (see Figure A-1) attached to the rear panel. The first five digits and the letter are the serial prefix and the last five digits are the suffix.



Figure A-1. Serial Number Plate

Change 1

The firmware revision 1.00, 1.01 does not support the following commands. Please delete the descriptions about these commands in this manual.

CLOSE

CWD?

FNAME?

FNUM?

FSIZE?

READ?

ROPEN

WOPEN

WRITE

B

Complex Operation Sub Program

This appendix provides the complex operation subprogram for the instrument BASIC which does not has the complex operation function. This sub program is provided as a user defined function. To use this sub program, add the

Complex Opearation Sub Program

This section provides adding, subtracting, multiplying, and dividing sub programs for the complex array. The parameter syntax of these sub programs is common.

Parameter	Description
$A(*)$, $B(*)$	Arrays for operation. The denominator for the dividing is B array. A and B array must be a same size array.
$C(*)$	Array for the operation result. C must be a same size with A and B .
U	Upper limit of the array.

The base of the array is fixed at 1.

Therefore, the operation result of A and B is entered to the array C . See "Sample Program: Using Complex Operation Sub Program" for the usage of this sub program.

```
SUB Cmadd(A(*),B(*),C(*),U)
  FOR I=1 TO U
    FOR J=1 TO 2
      C(I,J)=A(I,J)+B(I,J)
    NEXT J
  NEXT I
SUBEND
```

Module B-1. Adding Complex Arrays

```
SUB Cmsub(A(*),B(*),C(*),U)
  FOR I=1 TO U
    FOR J=1 TO 2
      C(I,J)=A(I,J)-B(I,J)
    NEXT J
  NEXT I
SUBEND
```

Module B-2. Subtracting Complex Arrays

```

SUB Cmmul(A(*),B(*),C(*),U)
  FOR I=1 TO U
    C(I,1)=A(I,1)*B(I,1)-A(I,2)*B(I,2)
    C(I,2)=A(I,1)*B(I,2)+A(I,2)*B(I,1)
  NEXT I
SUBEND

```

Module B-3. Multiplying Complex Arrays

```

SUB Cmdiv(A(*),B(*),C(*),U)
  FOR I=1 TO U
    De=(B(I,1)^2-B(I,2)^2)
    IF De=0 THEN Err
    C(I,1)=(A(I,1)*B(I,1)+A(I,2)*B(I,2))/De
    C(I,2)=(A(I,2)*B(I,1)-A(I,1)*B(I,2))/De
  NEXT I
  GOTO Ext
Err:DISP "DIVISION BY 0"
Ext:!
SUBEND

```

Module B-4. Dividing Complex Arrays

Sample Program: Using Complex Operation Sub Program

This program shows the simplex example of using the complex operation sub program.

Disk



This program is included in the *Sample Program Disk*. Its filename is COMPLEX. This program is initialized for Instrument BASIC. The program for the external controller is not included.

```

100 ASSIGN @Hp4291 TO 800
110 DIM Ch1(1:201,1:2),Ch2(1:201,1:2),Res(1:201,1:2)
120 OUTPUT @Hp4291;"TRAC? DTRCH1"
130 ENTER @Hp4291;Ch1(*)
140 OUTPUT @Hp4291;"TRAC? DTRCH2"
150 ENTER @Hp4291;Ch2(*)
160 !
170 Nop=201
180 Cmadd(Ch1(*),Ch2(*),Res(*),Nop)
190 !Cmsub(Ch1(*),Ch2(*),Res(*),Nop)
200 !Cmmul(Ch1(*),Ch2(*),Res(*),Nop)
210 !Cmdiv(Ch1(*),Ch2(*),Res(*),Nop)
220 !
230 OUTPUT @Hp4291;"TRAC: COPY TR2, TR1"
240 OUTPUT @HP4291;"TRAC TR2, ";Res(*)
250 OUTPUT @Hp4291;"DISP: TRAC2: STAT ON"
260 END
270 SUB Cmadd(A(*),B(*),C(*),U)
280   FOR I=1 TO U
290     FOR J=1 TO 2
300       C(I,J)=A(I,J)+B(I,J)

```

```
310     NEXT J
320     NEXT I
330 SUBEND
340 SUB Cmsub(A(*),B(*),C(*),U)
350   FOR I=1 TO U
360     FOR J=1 TO 2
370       C(I,J)=A(I,J)-B(I,J)
380     NEXT J
390   NEXT I
400 SUBEND
410 SUB Cmmul(A(*),B(*),C(*),U)
420   FOR I=1 TO U
430     C(I,1)=A(I,1)*B(I,1)-A(I,2)*B(I,2)
440     C(I,2)=A(I,1)*B(I,2)+A(I,2)*B(I,1)
450   NEXT I
460 SUBEND
470 SUB Cmdiv(A(*),B(*),C(*),U)
480   FOR I=1 TO U
490     De=(B(I,1)^2-B(I,2)^2)
500     IF De=0 THEN Err
510     C(I,1)=(A(I,1)*B(I,1)+A(I,2)*B(I,2))/De
520     C(I,2)=(A(I,2)*B(I,1)-A(I,1)*B(I,2))/De
530   NEXT I
540   GOTO Ext
550 Err:DISP "DIVISION BY 0"
560 Ext: !
570 SUBEND
```


GPIB Command List by Function

This appendix summarizes the GPIB commands according to the equivalent front panel keys as follows.

- Chan 1 Chan 2
- Meas Format Display Scale Ref Bw/Avg Cal
- Sweep Source Trigger Start Stop Center Span
- Marker Marker→ Search Utility
- System Local Preset Copy Save Recall

Functions other than those controlled by the front panel operation can be controlled using the analyzer's GPIB commands. These are the GPIB only functions. In this appendix, the following GPIB commands for the GPIB only functions are also listed:

- Calibration related commands
- Fixture compensation related commands
- Data array related commands
- Data transfer format related commands
- Error related commands
- Key related commands
- Limit test related commands
- HP instrument BASIC related commands
- Marker related commands
- Measurement parameter related commands
- Status byte related commands
- System related commands
- Data math related commands
- Trigger related commands
- Wait commands
- User trace related commands
- 8-bit I/O related commands
- File transfer related commands

Front Panel Key List with Equivalent GPIB Commands

Front Panel Key	Equivalent SCPI Command	Simple Command
Chan 1 (When dual channel ON) (When dual channel OFF)	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {OFF 0} INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1}	CHAN1
Chan 2 (When dual channel ON) (When dual channel OFF)	INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {OFF 0} INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1}	CHAN2
Meas (When Format LIN Y-AXIS, or LOG Y-AXIS is selected) IMPEDANCE: MAG(Z) PHASE(θ_z) (When Format EXP PHASE on OFF) (When Format EXP PHASE ON off) RESIST(R) REACT(X) ADMITTANCE: MAG(Y) PHASE(θ_y) (When Format EXP PHASE on OFF) (When Format EXP PHASE ON off) CONDUCT(G) SUSCEPT(B) REFL.COEF: MAG(Γ) PHASE(θ_{Γ}) (When Format EXP PHASE on OFF) (When Format EXP PHASE ON off) RESIST(Rx) REACT(Ry)	CALCulate:MATH1:STATe OFF CALCulate:FORMat MLINear CALCulate:MATH1:STATe OFF CALCulate:FORMat PHASE CALCulate:MATH1:STATe OFF CALCulate:FORMat UPHase CALCulate:MATH1:STATe OFF CALCulate:FORMat REAL CALCulate:MATH1:STATe OFF CALCulate:FORMat IMAGinary CALCulate:MATH1[:EXPRession]:NAME ADM CALCulate:MATH1:STATe ON CALCulate:FORMat MLINear CALCulate:MATH1[:EXPRession]:NAME ADM CALCulate:MATH1:STATe ON CALCulate:FORMat PHASE CALCulate:MATH1[:EXPRession]:NAME ADM CALCulate:MATH1:STATe ON CALCulate:FORMat UPHase CALCulate:MATH1[:EXPRession]:NAME ADM CALCulate:MATH1:STATe ON CALCulate:FORMat REAL CALCulate:MATH1[:EXPRession]:NAME ADM CALCulate:MATH1:STATe ON CALCulate:FORMat IMAGinary CALCulate:MATH1[:EXPRession]:NAME RCO CALCulate:MATH1:STATe ON CALCulate:FORMat MLINear CALCulate:MATH1[:EXPRession]:NAME RCO CALCulate:MATH1:STATe ON CALCulate:FORMat PHASE CALCulate:MATH1[:EXPRession]:NAME RCO CALCulate:MATH1:STATe ON CALCulate:FORMat UPHase CALCulate:MATH1[:EXPRession]:NAME RCO CALCulate:MATH1:STATe ON CALCulate:FORMat REAL CALCulate:MATH1[:EXPRession]:NAME RCO CALCulate:MATH1:STATe ON CALCulate:FORMat IMAGinary	MEAS IMAG MEAS IPH MEAS IRE MEAS IIM MEAS AMAG MEAS APH MEAS ARE MEAS AIM MEAS RCM MEAS RCPH MEAS RCR MEAS RCIM

Front Panel Key	Equivalent SCPI Command	Simple Command
CAPACITANCE: PRL(Cp)	CALCulate:MATH1:STATe OFF CALCulate:FORMat CP	MEAS CP
SER(Cs)	CALCulate:MATH1:STATe OFF CALCulate:FORMat CS	MEAS CS
INDUCTNCE: PRL(Lp)	CALCulate:MATH1:STATe OFF CALCulate:FORMat LP	MEAS LP
SER(Ls)	CALCulate:MATH1:STATe OFF CALCulate:FORMat LS	MEAS LS
RESISTNCE: PRL(Rp)	CALCulate:MATH1:STATe OFF CALCulate:FORMat RP	MEAS RP
SER(Rs)	CALCulate:MATH1:STATe OFF CALCulate:FORMat RS	MEAS RS
D FACTOR(D)	CALCulate:MATH1:STATe OFF CALCulate:FORMat D	MEAS D
Q FACTOR(Q)	CALCulate:MATH1:STATe OFF CALCulate:FORMat Q	MEAS Q
PRMITTVTY: REAL($\epsilon r'$)	CALCulate:MATH1[:EXPRession]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:FORMat REAL	MEAS DCR
LOSS FACTR($\epsilon r''$)	CALCulate:MATH1[:EXPRession]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:FORMat LFACtor	MEAS DCLF
LOSS TNGNT($\tan\delta$)	CALCulate:MATH1[:EXPRession]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:FORMat LTANgent	MEAS DCLT
MAG($ \epsilon r $)	CALCulate:MATH1[:EXPRession]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:FORMat MLINear	MEAS DCM
PRMEABLY: REAL($\mu r'$)	CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON CALCulate:FORMat REAL	MEAS PRE
LOSS FACTR($\mu r''$)	CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON CALCulate:FORMat LFACtor	MEAS PLF
LOSS TNGNT($\tan\delta$)	CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON CALCulate:FORMat LTANgent	MEAS PLT
MAG($ \mu r $)	CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON CALCulate:FORMat MLINear	MEAS PMAG
DUAL CHAN SETTING → See Dual Channel Setting menu		
FIXTURE []		
IMPEDANCE []		
PERMITTVTY 16453	SYSTem:FIXTure HP16453	
PERMEABLY []		
SELECT FIXTURE → See Select Fixture menu		
MATERIAL SIZE → See Material Size menu		

Meas

Front Panel Key	Equivalent SCPI Command	Simple Command
(When Format POLAR CHART, SMITH CHART, ADMITTANCE CHART, or COMPLEX PLANE is selected) IMPEDANCE (Z) ADMITTANCE (Y) REFL.COEF (Γ) PERMITTVTY (ε) PERMEABLY (μ) DUAL CHAN SETTING → See Dual Channel Setting menu FIXTURE [] IMPEDANCE [] PERMITTVTY 16453 PERMEABLY [] SELECT FIXTURE → See Select Fixture menu MATERIAL SIZE → See Material Size menu	CALCulate:MATH1:STATe OFF CALCulate:MATH1[:EXPRession]:NAME ADM CALCulate:MATH1:STATe ON CALCulate:MATH1[:EXPRession]:NAME RCO CALCulate:MATH1:STATe ON CALCulate:MATH1[:EXPRession]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON SYSTem:FIXTure HP16453	
Select Fixture menu (When FIXTURE [] IMPEDANCE [] is selected) FIXTURE: NONE 16191 16192 16193 16194 USER SAVE USER FXTR KIT MODIFY [NONE] DEFINE EXTENSION LABEL FIXTURE KIT DONE (MODIFIED) (When FIXTURE [] PERMEABLY [] is selected) FIXTURE: 16454(S) 16454(L)	SYSTem:FIXTure NONE SYSTem:FIXTure HP16191 SYSTem:FIXTure HP16192 SYSTem:FIXTure HP16193 SYSTem:FIXTure HP16194 SYSTem:FIXTure UDEFine SYSTem:FIXTure:SAVE SYSTem:FIXTure:DISTance <numeric> SYSTem:FIXTure:LABel <string> (None) SYSTem:FIXTure HP16454S SYSTem:FIXTure HP16454L	

Front Panel Key	Equivalent SCPI Command	Simple Command
Dual Channel Setting menu		
Z - θ		DUAM IMPH
(When Format EXP PHASE on OFF)	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe OFF CALCulate:FORMat MLINear INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe OFF CALCulate:FORMat PHASe	
(When Format EXP PHASE ON off)	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe OFF CALCulate:FORMat MLINear INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe OFF CALCulate:FORMat UPHase	
R-X	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe OFF CALCulate:FORMat REAL INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe OFF CALCulate:FORMat IMAGinary	DUAM IRIM
Y - θ		DUAM AMPH
(When Format EXP PHASE on OFF)	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME ADM CALCulate:MATH1:STATe ON CALCulate:FORMat MLINear INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME ADM CALCulate:MATH1:STATe ON CALCulate:FORMat PHASe	
(When Format EXP PHASE ON off)	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME ADM CALCulate:MATH1:STATe ON CALCulate:FORMat MLINear INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME ADM CALCulate:MATH1:STATe ON CALCulate:FORMat UPHase	
G-B	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME ADM CALCulate:MATH1:STATe ON CALCulate:FORMat REAL INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME ADM CALCulate:MATH1:STATe ON CALCulate:FORMat IMAGinary	DUAM ARIM
LS-Q	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe {OFF 0} CALCulate:FORMat LS INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe OFF CALCulate:FORMat Q	DUAM LSQ

Meas

Front Panel Key	Equivalent SCPI Command	Simple Command
Lp-Q	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe {OFF 0} CALCulate:FORMat LP INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe OFF CALCulate:FORMat Q	DUAM LPQ
Cs-D	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe {OFF 0} CALCulate:FORMat CS INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe OFF CALCulate:FORMat D	DUAM CSD
Cp-D	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe {OFF 0} CALCulate:FORMat CP INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1:STATe OFF CALCulate:FORMat D	DUAM CPD
PRMITTVTY: $\epsilon r' - \epsilon r''$	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPReSSion]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:FORMat REAL INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPReSSion]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:FORMat LFACTOR	DUAM DRLF
$\epsilon r' - \tan \delta$	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPReSSion]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:FORMat REAL INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPReSSion]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:FORMat LTAngent	DUAM DRLT
$\epsilon r'' - \tan \delta$	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPReSSion]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:FORMat LFACTOR INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPReSSion]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:FORMat LTAngent	DUAM DLFLT
$ \epsilon r - \tan \delta$	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPReSSion]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:FORMat MLINear INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPReSSion]:NAME DCO CALCulate:MATH1:STATe ON CALCulate:FORMat LTAngent	DUAM DMLT

Front Panel Key	Equivalent SCPI Command	Simple Command
PERMEABLTY: $\mu r'$ - $\mu r''$	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON CALCulate:FORMat REAL INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON CALCulate:FORMat LFACtor	DUAM PRLF
$\mu r'$ -tan δ	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON CALCulate:FORMat REAL INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON CALCulate:FORMat LTANGent	DUAM PRLT
$\mu r''$ -tan δ	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON CALCulate:FORMat LFACtor INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON CALCulate:FORMat LTANGent	DUAM PLFLT
$ \mu r $ -tan δ	INSTRument[:SElect] CH1 or INSTRument:NSElect 1 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON CALCulate:FORMat MLNear INSTRument[:SElect] CH2 or INSTRument:NSElect 2 INSTRument:STATe {ON 1} CALCulate:MATH1[:EXPRession]:NAME PER CALCulate:MATH1:STATe ON CALCulate:FORMat LTANGent	DUAM PMLT
SINGLE PARAMETER	(None)	
Material Size menu (When FIXTURE [] PERMITTVTY 16453 is selected) THICKNESS DONE (MODIFIED) (When FIXTURE [] PERMEABLTY [] is selected) INNER DIAMETER OUTER DIAMETER HEIGHT DONE (MODIFIED)	CALCulate:MATH1:DIMension1 <numeric> CALCulate:MATH1:DIMension2 <in>,<out>,<thick> (<in>: INNER DIAMETER, <out>: OUTER DIAMETER, <thick>: HEIGHT)	

Format

Front Panel Key	Equivalent SCPI Command	Simple Command
Format		
(When Display TRACE [DATA&MEM])		
LIN Y-AXIS	DISPlay[:WINDow]:TRACe:GRATICule:FORMat RECTangle DISPlay[:WINDow]:TRACe:Y:SPACing LINear	FMT LINY
LOG Y-AXIS	DISPlay[:WINDow]:TRACe:GRATICule:FORMat RECTangle DISPlay[:WINDow]:TRACe:Y:SPACing LOGarithmic	FMT LOGY
POLAR CHART	DISPlay[:WINDow]:TRACe:GRATICule:FORMat POLar CALCulate:FORMat COMPLex	FMT POL
SMITH CHART	DISPlay[:WINDow]:TRACe:GRATICule:FORMat SMITH CALCulate:FORMat COMPLex	FMT SMIT
ADMITTANCE CHART	DISPlay[:WINDow]:TRACe:GRATICule:FORMat ADMittance CALCulate:FORMat COMPLex	FMT ADM
COMPLEX PLANE	DISPlay[:WINDow]:TRACe:GRATICule:FORMat CPLane CALCulate:FORMat COMPLex	FMT COMP
PHASE UNIT [DEG RAD]		
DEG	CALCulate:FORMat:UNIT:ANGLE DEG	
RAD	CALCulate:FORMat:UNIT:ANGLE RAD	
EXP PHASE ON off	(None, see Meas)	
(When Display TRACE [USER])		
Y-AXIS: LIN	DISPlay[:WINDow]:TRACe{18-21}:Y:SPACing LINear	
LOG	DISPlay[:WINDow]:TRACe{18-21}:Y:SPACing LOGarithmic	
X-AXIS: LIN	DISPlay[:WINDow]:TRACe{18-21}:X:SPACing LINear	
LOG	DISPlay[:WINDow]:TRACe{18-21}:X:SPACing LOGarithmic	

Front Panel Key	Equivalent SCPI Command	Simple Command
Display		
DUAL CHAN ON off ¹	INSTRument[:SElect] CH1 INSTRument:STATe {ON 1} INSTRument[:SElect] CH2 INSTRument:STATe {ON 1}	DUAC {ON 1}
DUAL CHAN on OFF ¹ (When channel 1 is active)	INSTRument[:SElect] CH2 INSTRument:STATe {OFF 0} INSTRument[:SElect] CH1 INSTRument:STATe {ON 1}	DUAC {OFF 0}
(When channel 2 is active)	INSTRument[:SElect] CH1 INSTRument:STATe {OFF 0} INSTRument[:SElect] CH2 INSTRument:STATe {ON 1}	
SPLIT DISP ON off ¹	DISPlay[:WINDow]:FORMat ULowEr	
SPLIT DISP on OFF ¹	DISPlay[:WINDow]:FORMat FBACk	
DISPLAY ALLOCATION		
ALL INSTRUMENT	DISPlay[:WINDow]:ALLocation INSTRument	
HALF INSTR HALF BASIC	DISPlay[:WINDow]:ALLocation HIHB	
ALL BASIC	DISPlay[:WINDow]:ALLocation BASic	
BASIC STATUS	DISPlay[:WINDow]:ALLocation BSTatus	
GRAPHICS: BASIC DRAM	DISPlay[:WINDow]:GRAPhics:STATe {ON 1}	
ALL MEMORY TRACE	DISPlay[:WINDow]:GRAPhics:STATe {OFF 0}	
DEFINE TRACE → See Define Trace menu		
DATA MATH → See Data Math menu		
EQUIV CKT MENU ¹ → See Equivalent Circuit menu		
TITLE	DISPlay[:WINDow]:TEXT10[:DATA] <string>	
LABEL MENU → See Label menu		
USER TRACE LABEL ² → See User Trace Label menu		
TRACE [DATA&MEM]	(None, DISPlay[:WINDow]:TRACe{1-17}:STAT {ON 1} displays data trace and meary traces.)	
TRACE [USER]	(None, DISPlay[:WINDow]:TRACe{18-21}:STAT {ON 1} displays user traces.)	
GRATICULE ON off ¹	DISP:TRAC:GRAT:GRID {ON OFF 1 0} ³	
ADJUST DISPLAY → See Adjust display menu		
FREQUENCY BLANK ¹	DISPlay:ANNotation:FREquency {OFF 0} SYSTem:SECurity[:STATe] {ON 1}	FREO

1 This label is not displayed when TRACE [USER].

2 This label is not displayed when TARCE [DATA&MEM].

3 DISP:TRAC:GRAT:GRID : DISPlay[:WINDow]:TRACe1:GRATicule:GRID[:STATe]

Display

Front Panel Key	Equivalent SCPI Command	Simple Command
Define Trace menu (When TRACE [DATA&MEM])		
DISPLAY: DATA	DISPlay[:WINDow]:TRACe1:STATe {ON 1} DISPlay[:WINDow]:TRACe2:STATe {OFF 0} . DISPlay[:WINDow]:TRACe17:STATe {OFF 0}	DISP DATA
MEMORY	DISPlay[:WINDow]:TRACe{2-17}:STATe {ON 1} DISPlay[:WINDow]:TRACe1:STATe {OFF 0}	DISP MEMO
DATA & MEMORY	DISPlay[:WINDow]:TRACe1:STATe {ON 1} DISPlay[:WINDow]:TRACe{2-17}:STATe {ON 1}	DISP DATM
DATA → MEMORY	TRACe:COPIY TR{2-17},TR1	DATMEM
SELECT MEMORY NO	(None)	SELM <numeric>
SEL'D MEM ON off	DISPlay[:WINDow]:TRACe{2-17}:STATe {ON 1}	DISSMEMO {ON 1}
SEL'D MEM on OFF	DISPlay[:WINDow]:TRACe{2-17}:STATe {OFF 0}	DISSMEMO {OFF 0}
CLEAR MEMORIES	DISPlay[:WINDow]:TRACe2:CLEAr . DISPlay[:WINDow]:TRACe17:CLEAr	CLEM
(When TRACE [USER])		
DATA → USR	TRACe:COPIY TR{18-21},TR1	
MEMORY → USR	TRACe:COPIY TR{18-21},TR2	
SELECT UTRC [1]		
USER TRC 1	(None, :TRACe18 selects trace 1)	
USER TRC 2	(None, :TRACe19 selects trace 2)	
USER TRC 3	(None, :TRACe20 selects trace 3)	
USER TRC 4	(None, :TRACe21 selects trace 4)	
SEL'D UTRC ON off		
(When SELECT UTRC [1])	DISPlay[:WINDow]:TRACe18:STATe {ON 1}	
(When SELECT UTRC [2])	DISPlay[:WINDow]:TRACe19:STATe {ON 1}	
(When SELECT UTRC [3])	DISPlay[:WINDow]:TRACe20:STATe {ON 1}	
(When SELECT UTRC [4])	DISPlay[:WINDow]:TRACe21:STATe {ON 1}	
SEL'D UTRC on OFF		
(When SELECT UTRC [1])	DISPlay[:WINDow]:TRACe18:STATe {OFF 0}	
(When SELECT UTRC [2])	DISPlay[:WINDow]:TRACe19:STATe {OFF 0}	
(When SELECT UTRC [3])	DISPlay[:WINDow]:TRACe20:STATe {OFF 0}	
(When SELECT UTRC [4])	DISPlay[:WINDow]:TRACe21:STATe {OFF 0}	
CLEAR ALL UTRC	DISPlay[:WINDow]:TRACe{18-21}:CLEAr	

Front Panel Key	Equivalent SCPI Command	Simple Command
Data Math menu DATA MATH:DATA DATA+ MEM DATA-MEM DATA/MEM DATA*MEM DEFAULT GAIN & OFS OFFSET MKR→OFFSET OFFSET AUX OFFSET GAIN	CALCulate:MATH2:STATe OFF CALCulate:MATH2[:EXPRession]:NAME ADD CALCulate:MATH2:STATe ON CALCulate:MATH2[:EXPRession]:NAME SUB CALCulate:MATH2:STATe ON CALCulate:MATH2[:EXPRession]:NAME DIV CALCulate:MATH2:STATe ON CALCulate:MATH2[:EXPRession]:NAME MUL CALCulate:MATH2:STATe ON DATA[:DATA] GAIN,1 DATA[:DATA] OFFS,0 DATA[:DATA] AOFF,0 DATA[:DATA] OFFS,MARKer DATA[:DATA] OFFS,<numeric> DATA[:DATA] AOFF,<numeric> DATA[:DATA] GAIN,<numeric>	MATH DATA MATH DPLM MATH DMNM MATH DDVM MATH DMLM
Equivalent Circuit menu SELECT EQV CKT [A] A B C D E DISP EQV PARM [ON] DISP EQV PARM [OFF] DEFINE EQV PARAMS R1 C1 L1 C0 CALCULATE EQV PARAMS SIMULATE F-CHRST	CALCulate:EVALuate:EPARameters:CIRCUit A CALCulate:EVALuate:EPARameters:CIRCUit B CALCulate:EVALuate:EPARameters:CIRCUit C CALCulate:EVALuate:EPARameters:CIRCUit D CALCulate:EVALuate:EPARameters:CIRCUit E DISPlay[:WINDow]:TEXT39:STAT {ON 1} DISPlay[:WINDow]:TEXT39:STAT {OFF 0} DATA[:DATA] EQR1,<numeric> DATA[:DATA] EQC1,<numeric> DATA[:DATA] EQL1,<numeric> DATA[:DATA] EQC0,<numeric> CALCulate:EVALuate:EPARameters DISPlay[:WINDow]:TEXT39:STATe {ON 1} CALCulate:EVALuate:EPARameters:SIMulation	
Label menu LABEL COLOR X POS Y POS LABEL NUMBER CLEAR ALL LABEL	DISPlay[:WINDow]:TEXT{11-30}[:DATA] <string> DISPlay[:WINDow]:TEXT{11-30}:COLor <numeric> DISPlay[:WINDow]:TEXT{11-30}:LOCate <x>,<y> ¹ (<x>:Y POS , <y>:X POS) (None, DISPlay[:WINDow]:TEXT11 selects label number 0, . DISPlay[:WINDow]:TEXT30 selects label number 19.) DISPlay[:WINDow]:TEXT{11-30}:CLEar	

1 <x>,<y> : <numeric(x)>,<numeric(y)>

Front Panel Key	Equivalent SCPI Command	Simple Command
Scale Ref AUTO SCALE SCALE/DIV ² REFERENCE POSITION ² REFERENCE VALUE ² MKR→REFERENCE ² LEFT VALUE ³ RIGHT VALUE ³ TOP VALUE BOTTOM VALUE AXIS [COUPLE] ³ AXIS [UNCOUPLE] ³ SCALE FOR [DATA] ² SCALE FOR [MEMO] ² D&M SCALE [COUPLE] ² D&M SCALE [UNCOUPLE] ² REFERENCE X VALUE ² REFERENCE Y VALUE ²	DISP:TRAC{1-21}:Y[:SCALE]:AUTO ONCE ¹ DISP:TRAC{1-17}:Y[:SCALE]:PDIVision <numeric> ¹ DISP:TRAC{1-17}:Y[:SCALE]:RPOsition <numeric> ¹ DISP:TRAC{1-17}:Y[:SCALE]:RLEVel <numeric> ¹ DISP:TRAC{1-17}:Y[:SCALE]:RLEVel MARKer ¹ DISP:TRAC{18-21}:X[:SCALE]:LEFT <numeric> ¹ DISP:TRAC{18-21}:X[:SCALE]:RIGHt <numeric> ¹ DISP:TRAC{1-21}:Y[:SCALE]:TOP <numeric> ¹ DISP:TRAC{1-21}:Y[:SCALE]:BOTTom <numeric> ¹ DISP:TRAC{18-21}:GRATicle:AXIS:COUPle {ON 1} ¹ DISP:TRAC{18-21}:GRATicle:AXIS:COUPle {OFF 0} ¹ (None, "DISP:TRAC1 ... " selects data) (None, "DISP:TRAC{2-17} ... " selects memory) DISP:TRAC:Y[:SCALE]:COUPle {ON 1} ¹ DISP:TRAC:Y[:SCALE]:COUPle {OFF 0} ¹ DISP:TRAC{1-17}:X[:SCALE]:RLEVel <numeric> ¹ DISP:TRAC{1-17}:Y[:SCALE]:RLEVel <numeric> ¹	
Bw/Avg AVERAGING RESTART AVERAGING ON off AVERAGING FACTOR POINT AVG ON off POINT AVG FACTOR	SENS:e:AVERAge2:CLEar SENS:e:AVERAge2[:STATe] {OFF ON 0 1} SENS:e:AVERAge2:COUNT <numeric> SENS:e:AVERAge1[:STATe] {OFF ON 0 1} SENS:e:AVERAge1:COUNT <numeric>	
Cal CALIBRATE MENU OPEN SHORT LOAD LOW-LOSS CAPACITOR CAL POINTS [FIXED] CAL POINTS [USER] DONE: CAL RESUME CAL SEQUENCE	SENS:e:CORRection1:COLLect[:ACQuire] STANdard1 SENS:e:CORRection1:COLLect[:ACQuire] STANdard2 SENS:e:CORRection1:COLLect[:ACQuire] STANdard3 SENS:e:CORRection1:COLLect[:ACQuire] STANdard4 SENS:e:CORRection1:COLLect:FPOints FIXEd SENS:e:CORRection1:COLLect:FPOints USER SENS:e:CORRection1:COLLect:SAVE (None)	

1 DISP:TRAC : DISPlay[:WINDow]:TRACe

2 This label is not displayed when TRACE [USER].

3 This label is not displayed when TRACE [DATA&MEM].

Scale Ref Bw/Avg Cal

Front Panel Key	Equivalent SCPI Command	Simple Command
FIXTURE COMPEN		
COMPEN MENU		
OPEN (When the impedance test fixture is selected)	SENSe:CORRection2:COLLect[:ACQuire] STANdard1	
(When the permittivity test fixture is selected)	SENSe:CORRection2:COLLect[:ACQuire] STANdard4	
SHORT (When impedance test fixture is selected)	SENSe:CORRection2:COLLect[:ACQuire] STANdard2	
(When permittivity test fixture is selected)	SENSe:CORRection2:COLLect[:ACQuire] STANdard5	
(When permeability test fixture is selected)	SENSe:CORRection2:COLLect[:ACQuire] STANdard7	
LOAD (When impedance test fixture is selected)	SENSe:CORRection2:COLLect[:ACQuire] STANdard3	
(When permittivity test fixture is selected)	SENSe:CORRection2:COLLect[:ACQuire] STANdard6	
COMP POINT [FIXED]	SENSe:CORRection2:COLLect:FPOints FIXed	
COMP POINT [USER]	SENSe:CORRection2:COLLect:FPOints USER	
DONE: COMPEN	SENSe:CORRection2:COLLect:SAVE	
RESUME COMP SEQ	(None)	
OPEN ON off	SENSe:CORRection2:OPEN {OFF ON 0 1}	
SHORT ON off	SENSe:CORRection2:SHORT {OFF ON 0 1}	
LOAD ON off	SENSe:CORRection2:LOAD {OFF ON 0 1}	
CAL KIT [7mm]		
CAL KIT:7mm	SENSe:CORRection1:CKIT APC7	
USER KIT	SENSe:CORRection1:CKIT UDEFined	
SAVE USER KIT	SENSe:CORRection1:CKIT:SAVE	
MODIFY [7mm]		
→ See Modify calkit menu		
COMPEN KIT []		
COMP KIT: TEFLON ¹	SENSe:CORRection2:CKIT2 TEFLon	
USER KIT ¹	SENSe:CORRection2:CKIT2 UDEFined	
SAVE COMPEM KIT		
(When the impedance test fixture is selected)	SENSe:CORRection2:CKIT[1]:SAVE	
(When the permittivity test fixture is selected)	SENSe:CORRection2:CKIT2:SAVE	
MODIFY []		
→ See Modify compenkit menu		
PORT EXTENSION		
EXTENSION ON off	SENSe:CORRection1:EDELay:STATe {OFF ON 0 1}	
EXTENSION VALUE	SENSe:CORRection1:EDELay[:TIME] <numeric>	

¹ This softkey label is only for the permittivity test fixture.

Front Panel Key	Equivalent SCPI Command	Simple Command
Modify calkit menu DEFINE STANDARD OPEN: CONDUCT(G) CAP.(C) SHORT: RESIST.(R) INDUCT.(L) LOAD: RESIST.(R) REACT.(X) STD DONE (DEFINED) LABEL KIT KIT DONE (MODIFIED)	SENSE:CORRection1:CKIT:STANdard1:G <numeric> SENSE:CORRection1:CKIT:STANdard1:C <numeric> SENSE:CORRection1:CKIT:STANdard2:R <numeric> SENSE:CORRection1:CKIT:STANdard2:L <numeric> SENSE:CORRection1:CKIT:STANdard3:R <numeric> SENSE:CORRection1:CKIT:STANdard3:X <numeric> (None) SENSE:CORRection1:CKIT:LABel <string> (None)	
Modify compenkit menu (When the impedance test fixture is selected) DEFINE STANDARD OPEN: CONDUCT(G) CAP.(C) SHORT: RESIST.(R) INDUCT.(L) LOAD: RESIST.(R) INDUCT.(L) STD DONE (DEFINED) LABEL KIT KIT DONE (MODIFIED) (When the permittivity test fixture is selected) DEFINE STANDARD LOAD: ϵ_r REAL ϵ_r LOSS THICKNESS STD DONE (DEFINED) LABEL KIT KIT DONE (MODIFIED)	SENSE:CORRection2:CKIT[1]:STANdard1:G <numeric> SENSE:CORRection2:CKIT[1]:STANdard1:C <numeric> SENSE:CORRection2:CKIT[1]:STANdard2:R <numeric> SENSE:CORRection2:CKIT[1]:STANdard2:L <numeric> SENSE:CORRection2:CKIT[1]:STANdard3:R <numeric> SENSE:CORRection2:CKIT[1]:STANdard3:L <numeric> (None) SENSE:CORRection2:CKIT[1]:LABel <string> (None) SENSE:CORRection2:CKIT2:STANdard6:PREal <num> ¹ SENSE:CORRection2:CKIT2:STANdard6:PLFactor <num> ¹ SENSE:CORRection2:CKIT2:STANdard6:THICKness <num> ¹ (None) SENSE:CORRection2:CKIT2:LABel <string> (None)	

1 <num> : <numeric>

Sweep

Front Panel Key	Equivalent SCPI Command	Simple Command
Sweep		
SWEEP TIME [AUTO]		
SWEEP TIME AUTO	SENSe:SWEEp:TIME:AUTO {ON 1}	
h:m:s	SENSe:SWEEp:TIME <numeric>	
POINT DELAY TIME	SENSe:SWEEp:DWELl1 <numeric> SENSe:SWEEp:DWELl1:AUTO {ON 1}	POIDTIME <numeric>
SWEEP DELAY TIME	SENSe:SWEEp:DWELl2 <numeric> SENSe:SWEEp:DWELl2:AUTO {ON 1}	SWEDTIME <numeric>
NUMBER OF POINTS	SENSe:SWEEp:POINts <numeric>	
COUPLED CH ON off	INSTRument:COUPlE ALL	
COUPLED CH on OFF	INSTRument:COUPlE NONE	
SWEEP MENU → See Sweep menu		
SWEEP DIR [UP]		SWED UP
(frequency or OSC level sweep)	SOURce1:SWEEp:DIRectiOn UP	
(DC-I/DC-V sweep)	SOURce2:SWEEp:DIRectiOn UP	
SWEEP DIR [DOWN]		SWED DOWN
(frequency or OSC level sweep)	SOURce1:SWEEp:DIRectiOn DOWN	
(DC-I/DC-V sweep)	SOURce2:SWEEp:DIRectiOn DOWN	
LIST MENU		
LIST DISP: FREQ BASE	DISPlay[:WINDow]:TRACe:X:SPACing LINear	LISD FBAS
ODER BASE	DISPlay[:WINDow]:TRACe:X:SPACing OBASe	
EDIT LIST		
SEGMENT	SENSe:LIST:SEGMENT <numeric>	
EDIT → See Edit menu	SENSe:LIST:SEGMENT:EDIT	
DELETE	SENSe:LIST:SEGMENT:DELEte	
ADD	SENSe:LIST:SEGMENT:ADD	
CLEAR LIST	SENSe:LIST:CLEAr	
LIST DONE	SENSe:LIST:SAVE	

Front Panel Key	Equivalent SCPI Command	Simple Command
Sweep menu		
SWP SRC: FREQ	SENSe:FREQuency:MODE SWEEp SOURce1:VOLTage:MODE FIXEd SOURce2:CURRent:MODE FIXEd SOURce2:VOLTage:MODE FIXEd	SWES FREQ
OSC LEVEL	SENSe:FREQuency:MODE FIXEd SOURce1:VOLTage:MODE SWEEp SOURce2:CURRent:MODE FIXEd SOURce2:VOLTage:MODE FIXEd	SWES OLEV
DC BIAS V (optional)	SENSe:FREQuency:MODE FIXEd SOURce1:VOLTage:MODE FIXEd SOURce2:CURRent:MODE FIXEd SOURce2:VOLTage:MODE SWEEp	SWES DCV
DC BIAS I (optional)	SENSe:FREQuency:MODE FIXEd SOURce1:VOLTage:MODE FIXEd SOURce2:CURRent:MODE SWEEp SOURce2:VOLTage:MODE FIXEd	SWES DCI
SWP TYPE: LIN		STYPE LIN
(When SWP SRC: FREQ)	SENSe:SWEEp:SPACing LINear DISPlay[:WINDow]:TRACe:X:SPACing LINear	
(When OSC LEVEL)	SOURce1:SWEEp:SPACing LINear DISPlay[:WINDow]:TRACe:X:SPACing LINear	
(When DC BIAS V, or DC BIAS I)	SOURce2:SWEEp:SPACing LINear DISPlay[:WINDow]:TRACe:X:SPACing LINear	
LOG		STYPE LOG
(When SWP SRC: FREQ)	SENSe:SWEEp:SPACing LOGarithmic DISPlay[:WINDow]:TRACe:X:SPACing LOGarithmic	
(When OSC LEVEL)	SOURce1:SWEEp:SPACing LOGarithmic DISPlay[:WINDow]:TRACe:X:SPACing LOGarithmic	
(When DC BIAS V, or DC BIAS I)	SOURce2:SWEEp:SPACing LOGarithmic DISPlay[:WINDow]:TRACe:X:SPACing LOGarithmic	
LIST	SENSe:FREQuency:MODE LIST SOURce1:VOLTage:MODE LIST SOURce2:CURRent:MODE FIXEd SOURce2:VOLTage:MODE FIXEd SENSe:SWEEp:SPACing LINear	STYPE LIST
Edit menu		
SEGMENT: MKR→START	SENSe:LIST:SEGMENT:FREQuency:STARt MARKer	
MKR→STOP	SENSe:LIST:SEGMENT:FREQuency:STOP MARKer	
NUMBER of POINTS	SENSe:LIST:SEGMENT:POINts <numeric>	
OSC LEVEL		
(When OSC UNIT[AMPERE])	SENSe:LIST:SEGMENT:CURRent <numeric>	
(When OSC UNIT[dBm])	SENSe:LIST:SEGMENT:POWer <numeric>	
(When OSC UNIT[VOLT])	SENSe:LIST:SEGMENT:VOLTage <numeric>	
AVERAGING ON POINT	SENSe:LIST:SEGMENT:AVERage:COUNt <numeric>	
MORE		
SEGMENT: START	SENSe:LIST:SEGMENT:FREQuency:STARt <numeric>	STAR
STOP	SENSe:LIST:SEGMENT:FREQuency:STOP <numeric>	STOP
CENTER	SENSe:LIST:SEGMENT:FREQuency:CENTer <numeric>	CENT
SPAN	SENSe:LIST:SEGMENT:FREQuency:SPAN <numeric>	SPAN
SEGMENT QUIT	SENSe:LIST:SEGMENT:QUIT	
SEGMENT DONE	SENSe:LIST:SEGMENT:SAVE	

Source

Front Panel Key	Equivalent SCPI Command	Simple Command
Source		
OSC LEVEL		
(When OSC UNIT[VOLT])	SOUR:VOLT <numeric> ¹	
(When OSC UNIT[AMPERE])	SOUR:CURR <numeric> ²	
(When OSC UNIT[dBm])	SOUR:POW <numeric> ³	
OSC UNIT[VOLT]		
VOLT	(None, defined by SOUR:{VOLT CURR POW} node)	
AMPERE	(None, defined by SOUR:{VOLT CURR POW} node)	
dBm	(None, defined by SOUR:{VOLT CURR POW} node)	
CW FREQ	SOURce1:FREQuency[:CW]:FIXed <numeric>	
DC BIAS ON off (optional)		
(When BIAS SRC [VOLTAGE])	SOURce2:VOLTagE:STATe {OFF ON 0 1}	
(When BIAS SRC [CURRENT])	SOURce2:CURRent:STATe {OFF ON 0 1}	
DC BIAS MENU		
BIAS SRC [VOLTAGE]	SOURce2:VOLTagE:ALC[:STATe] {ON 1}	
BIAS SRC [CURRENT]	SOURce2:CURRent:ALC[:STATe] {ON 1}	
BIAS VOLTAGE	SOUR2:VOLT <numeric> ⁴	
BIAS CUR LIMIT	SOURce2:CURRent:LIMit[:AMPLitude] <numeric>	
BIAS CURRENT	SOUR2:CURR <numeric> ⁵	
BIAS VOLT LIMIT	SOURce2:VOLTagE:LIMit[:AMPLitude] <numeric>	

1 SOUR:VOLT : SOURce1:VOLTagE[:LEVel][:IMMediate][:AMPLitude]

2 SOUR:CURR : SOURce1:CURRent[:LEVel][:IMMediate][:AMPLitude]

3 SOUR:POW : SOURce1:POWer[:LEVel][:IMMediate][:AMPLitude]

4 SOUR2:VOLT : SOURce2:VOLTagE[:LEVel][:IMMediate][:AMPLitude]

5 SOUR2:CURR : SOURce2:CURRent[:LEVel][:IMMediate][:AMPLitude]

Front Panel Key	Equivalent SCPI Command	Simple Command
<p>Trigger</p> <p>SWEEP:HOLD</p> <p>SINGLE</p> <p>NUMBER of GROUPS</p> <p>CONTINUOUS</p> <p>TRIGGER:[FREE RUN]</p> <p>TRIGGER:FREE RUN</p> <p>EXTERNAL</p> <p>MANUAL</p> <p>(When TRIGGER:[FREE RUN], or TRIGGER:[EXTERNAL])</p> <p>(When TRIGGER:[MANUAL])</p> <p>TRIG EVENT [ON SWEEP]</p> <p>TRIG EVENT [ON POINT]</p> <p>TRIG PLRTY POS neg</p> <p>MEASURE RESTART</p>	<p>INITiate:CONTInuous { OFF 0} ABORt</p> <p>INITiate:CONTInuous { OFF 0} ABORt SENSe:SWEep:COUNT 1 INITiate[:IMMediate]</p> <p>INITiate:CONTInuous { OFF 0} SENSe:SWEep:COUNT <numeric> INITiate[:IMMediate]</p> <p>INITiate:CONTInuous { ON 1}</p> <p>TRIGger:SOURce INternal</p> <p>TRIGger:SOURce EXTernal</p> <p>TRIGger:SOURce MANual</p> <p>(None)</p> <p>TRIGger:EVENT:TYPE SWEep</p> <p>TRIGger:EVENT:TYPE POINT</p> <p>TRIGger:SLOPe { POSitive NEGative}</p> <p>INITiate[:IMMediate]:AGain:ALL</p>	<p>HOLD</p> <p>SING</p> <p>NUMG</p> <p>CONT</p>
<p>Start</p> <p>(When sweep source is frequency)</p> <p>(When sweep source is OSC level)</p> <p>(When sweep source is dc bias voltage)</p> <p>(When sweep source is dc bias current)</p>	<p>SENSe:FREQuency:STARt <numeric></p> <p>SOURce1:VOLTage:STARt <numeric></p> <p>SOURce2:VOLTage:STARt <numeric></p> <p>SOURce2:CURRent:STARt <numeric></p>	<p>STAR</p>
<p>Stop</p> <p>(When sweep source is frequency)</p> <p>(When sweep source is OSC level)</p> <p>(When sweep source is dc bias voltage)</p> <p>(When sweep source is dc bias current)</p>	<p>SENSe:FREQuency:STOP <numeric></p> <p>SOURce1:VOLTage:STOP <numeric></p> <p>SOURce2:VOLTage:STOP <numeric></p> <p>SOURce2:CURRent:STOP <numeric></p>	<p>STOP</p>
<p>Center</p> <p>(When sweep source is frequency)</p> <p>(When sweep source is OSC level)</p> <p>(When sweep source is dc bias voltage)</p> <p>(When sweep source is dc bias current)</p>	<p>SENSe:FREQuency:CENTer <numeric></p> <p>SOURce1:VOLTage:CENTer <numeric></p> <p>SOURce2:VOLTage:CENTer <numeric></p> <p>SOURce2:CURRent:CENTer <numeric></p>	<p>CENT</p>
<p>Span</p> <p>(When sweep source is frequency)</p> <p>(When sweep source is OSC level)</p> <p>(When sweep source is dc bias voltage)</p> <p>(When sweep source is dc bias current)</p>	<p>SENSe:FREQuency:SPAN <numeric></p> <p>SOURce1:VOLTage:SPAN <numeric></p> <p>SOURce2:VOLTage:SPAN <numeric></p> <p>SOURce2:CURRent:SPAN <numeric></p>	<p>SPAN</p>

Marker

Front Panel Key	Equivalent SCPI Command	Simple Command
Marker (rotary knob) SUB MKR → See Sub-marker menu CLEAR SUB MKR → See Sub-marker menu PRESET MKRS MKR ON [DATA] MKR ON [MEMORY] MKR [UNCOUPLE] MKR [COUPLE] MKR [CONT] MKR [DISCRETE] ΔMODE MENU → See Delta mode menu	DISPLAY[:WINDow]:TRACe:MARKer1:ALL:STATe {ON 1} CALCulate:EVALuate:Y[1]:XPOSITION <numeric> DISPLAY[:WINDow]:TRACe1:MARKer1:ALL DEFault CALCulate:EVALuate:ON1 "TR1" CALCulate:EVALuate:ON1 "TR{2-17}" CALCulate:EVALuate:COUPlE {OFF 0} CALCulate:EVALuate:COUPlE {ON 1} CALCulate:EVALuate:INTerpolate {ON 1} CALCulate:EVALuate:INTerpolate {OFF 0}	MARK {ON 1}
Sub-marker menu SUB MKR 1 2 3 4 5 6 7	DISPLAY[:WINDow]:TRACe:MARKer2:STATe {OFF ON 0 1} DISPLAY[:WINDow]:TRACe:MARKer3:STATe {OFF ON 0 1} DISPLAY[:WINDow]:TRACe:MARKer4:STATe {OFF ON 0 1} DISPLAY[:WINDow]:TRACe:MARKer5:STATe {OFF ON 0 1} DISPLAY[:WINDow]:TRACe:MARKer6:STATe {OFF ON 0 1} DISPLAY[:WINDow]:TRACe:MARKer7:STATe {OFF ON 0 1} DISPLAY[:WINDow]:TRACe:MARKer8:STATe {OFF ON 0 1}	
Delta mode menu ΔMKR FIXED ΔMKR TRACKING ΔMKR ΔMODE OFF ΔMKR STIMULUS FIXED ΔMKR VALUE FIXED ΔMKR AUX VALUE	DISPLAY[:WINDow]:TRACe:MARKer:RELAtive {ON 1} DISP:TRAC:MARK:REL:REF MARKer ¹ DISPLAY[:WINDow]:TRACe:MARKer:RELAtive {ON 1} DISP:TRAC:MARK:REL:REF FIXed ¹ DISPLAY[:WINDow]:TRACe:MARKer:RELAtive {ON 1} DISP:TRAC:MARK:REL:REF TRACked ¹ DISPLAY[:WINDow]:TRACe:MARKer:RELAtive {OFF 0} CALCulate:EVALuate:REFerence:X <numeric> CALCulate:EVALuate:REFerence:Y1 <numeric> CALCulate:EVALuate:REFerence:Y2 <numeric>	

1 DISP:TRAC:MARK:REL:REF : DISPLAY[:WINDow]:TRACe:MARKer:RELAtive:REFerence

Front Panel Key	Equivalent SCPI Command	Simple Command
Marker→	DISPlay[:WINDow]:TRACe:MARKer1:ALL:STATe {ON 1}	
MKR→CENTER (When sweep source is frequency) (When sweep source is OSC level) (When sweep source is dc bias voltage) (When sweep source is dc bias current)	SENSe:FREQUency:CENTer MARKer SOURce1:VOLTage:CENTer MARKer SOURce2:VOLTage:CENTer MARKer SOURce2:CURRent:CENTer MARKer	MARKCENT
MKR→START (When sweep source is frequency) (When sweep source is OSC level) (When sweep source is dc bias voltage) (When sweep source is dc bias current)	SENSe:FREQUency:STARt MARKer SOURce1:VOLTage:STARt MARKer SOURce2:VOLTage:STARt MARKer SOURce2:CURRent:STARt MARKer	MARKSTAR
MKR→STOP (When sweep source is frequency) (When sweep source is OSC level) (When sweep source is dc bias voltage) (When sweep source is dc bias current)	SENSe:FREQUency:STOP MARKer SOURce1:VOLTage:STOP MARKer SOURce2:VOLTage:STOP MARKer SOURce2:CURRent:STOP MARKer	MARKSTOP
MKR→REFERENCE	DISP:TRAC{1-21}:Y:RLEV MARK ¹	MARKREF
MKR ZOOM (When sweep source is frequency) (When sweep source is OSC level) (When sweep source is dc bias voltage) (When sweep source is dc bias current)	SENSe:FREQUency:SPAN MZAPerture SOURce1:VOLTage:SPAN MZAPerture SOURce2:VOLTage:SPAN MZAPerture SOURce2:CURRent:SPAN MZAPerture	MARZ
PEAK→CENTER (When sweep source is frequency) (When sweep source is OSC level) (When sweep source is dc bias voltage) (When sweep source is dc bias current)	SENSe:FREQUency:CENTer TPEak SOURce1:VOLTage:CENTer TPEak SOURce2:VOLTage:CENTer TPEak SOURce2:CURRent:CENTer TPEak	PEAKCENT
CROSS CHAN ON off	CALCulate:EVALuate:EFFect:ON {1 2}	
MKRA→SPAN (When sweep source is frequency) (When sweep source is OSC level) (When sweep source is dc bias voltage) (When sweep source is dc bias current)	SENSe:FREQUency:SPAN DMARKer SOURce1:VOLTage:SPAN DMARKer SOURce2:VOLTage:SPAN DMARKer SOURce2:CURRent:SPAN DMARKer	MARDSPAN
MKRA→CENTER (When sweep source is frequency) (When sweep source is OSC level) (When sweep source is dc bias voltage) (When sweep source is dc bias current)	SENSe:FREQUency:CENTer DMARKer SOURce1:VOLTage:CENTer DMARKer SOURce2:VOLTage:CENTer DMARKer SOURce2:CURRent:CENTer DMARKer	MARDCENT
ZOOMING APERTURE	DATA[:DATA] MZAP, <numeric>	

¹ DISP:TRAC{1-21}:Y:RLEV MARK : DISPlay[:WINDow]:TRACe{1-21}:Y[:SCALE]:RLEVel MARKer

Search

Front Panel Key	Equivalent SCPI Command	Simple Command
<p>Search</p> <p>SEARCH: MAX (When SEARCH TRK on OFF) (When SEARCH TRK ON off)</p> <p>MIN (When SEARCH TRK on OFF) (When SEARCH TRK ON off)</p> <p>TARGET (When SEARCH TRK on OFF) (When SEARCH TRK ON off)</p> <p>TARGET SEARCH LEFT SEARCH RIGHT SUB MKR → See Sub-marker menu</p> <p>PEAK (When SEARCH TRK on OFF) (When SEARCH TRK ON off)</p> <p>SEARCH TRK ON off SEARCH TRK on OFF</p> <p>WIDTHS [OFF] SEARCH IN SEARCH OUT WIDTHS ON off WIDTH VALUE MKRVAL/($\sqrt{2}$) MKRVAL*($\sqrt{2}$) MKRVAL/2 FIXED VALUE</p> <p>SEARCH RANGE MENU → See Search range menu</p>	<p>DISPlay[:WINDow]:TRACe:MARKer:STATe {ON 1}</p> <p>CALCulate:EVALuate:Y[1]:XPOSITION:MAXimum</p> <p>CALCulate:EVALuate:Y[1]:XPOSITION:TRACk MAXimum</p> <p>CALCulate:EVALuate:Y[1]:XPOSITION:MINimum</p> <p>CALCulate:EVALuate:Y[1]:XPOSITION:TRACk MINimum</p> <p>CALCulate:EVALuate:Y[1]:XPOSITION:TARGet <numeric></p> <p>CALCulate:EVALuate:Y[1]:XPOSITION:TRACk TARGet</p> <p>CALCulate:EVALuate:Y[1]:XPOSITION:TARGet <numeric></p> <p>CALCulate:EVALuate:Y[1]:XPOSITION:LTARGet</p> <p>CALCulate:EVALuate:Y[1]:XPOSITION:RTARGet</p> <p>CALCulate:EVALuate:Y[1]:XPOSITION:PEAK → See Peak menu</p> <p>CALCulate:EVALuate:Y[1]:XPOSITION:TRACk PEAK</p> <p>(None, see SEARCH: MAX, MIN, TARGET)</p> <p>CALCulate:EVALuate:Y[1]:XPOSITION:TRACk OFF</p> <p>CALCulate:EVALuate:WIDTH:XPOSITION:IN</p> <p>CALCulate:EVALuate:WIDTH:XPOSITION:OUT</p> <p>CALCulate:EVALuate:WIDTH:STATe {OFF ON 0 1}</p> <p>CALCulate:EVALuate:WIDTH:Y DIVS2</p> <p>CALCulate:EVALuate:WIDTH:Y MULS2</p> <p>CALCulate:EVALuate:WIDTH:Y DIV2</p> <p>CALCulate:EVALuate:WIDTH:Y FIXed, <numeric></p>	
<p>Sub-marker menu</p> <p>SUB MKR 1 2 3 4 5 6 7</p>	<p>DISPlay[:WINDow]:TRACe:MARKer2:STATe {OFF ON 0 1}</p> <p>DISPlay[:WINDow]:TRACe:MARKer3:STATe {OFF ON 0 1}</p> <p>DISPlay[:WINDow]:TRACe:MARKer4:STATe {OFF ON 0 1}</p> <p>DISPlay[:WINDow]:TRACe:MARKer5:STATe {OFF ON 0 1}</p> <p>DISPlay[:WINDow]:TRACe:MARKer6:STATe {OFF ON 0 1}</p> <p>DISPlay[:WINDow]:TRACe:MARKer7:STATe {OFF ON 0 1}</p> <p>DISPlay[:WINDow]:TRACe:MARKer8:STATe {OFF ON 0 1}</p>	

Front Panel Key	Equivalent SCPI Command	Simple Command
Peak menu PEAK NEXT PEAK NEXT PEAK LEFT NEXT PEAK RIGHT PEAK DEF MENU → See Peak definition menu SUB MKR → See Sub-marker menu	CALCulate:EVALuate:Y[1]:XPOSITION:PEAK CALCulate:EVALuate:Y[1]:XPOSITION:NPEak CALCulate:EVALuate:Y[1]:XPOSITION:LPEak CALCulate:EVALuate:Y[1]:XPOSITION:RPEak	
Peak definition menu THRESHOLD ON off THRESHOLD VALUE MKR→THRESHOLD PEAK PLRTY POS neg PEAK DEF: ΔX PEAK DEF: ΔY MKR→PEAK DELTA	CALC: EVAL: PEAK: THR: STAT {OFF ON 0 1} ¹ CALCulate:EVALuate:PEAK:THReshold <numeric> CALCulate:EVALuate:PEAK:THReshold MARKer CALCulate:EVALuate:PEAK:POLarity {POSitive NEGative} CALCulate:EVALuate:PEAK:EXCursion: X <numeric> CALCulate:EVALuate:PEAK:EXCursion[:Y] <numeric> CALCulate:EVALuate:PEAK:EXCursion: X DMARKer	
Search range menu PART SRCH ON off MKRΔ→SEARCH RNG MKR→LEFT RNG MKR→RIGHT RNG	CALCulate:EVALuate:BAND:FULL[:STATe] {OFF ON 0 1} CALCulate:EVALuate:BAND:SPAN DMARKer CALCulate:EVALuate:BAND:STARt MARKer CALCulate:EVALuate:BAND:STOP MARKer	

¹ CALC: EVAL: PEAK: THR: STAT : CALCulate:EVALuate:PEAK:THReshold:STATe

Utility

Front Panel Key	Equivalent SCPI Command	Simple Command
Utility	DISPlay[:WINDow]:TRACk:MARKer1:ALL:STATe {ON 1}	
MKR LIST ON off	DISPlay[:WINDow]:TEXT9:STATe {OFF ON 0 1}	
STATISTICS ON off	CALCulate:EVALuate:MSStatistics[:STATe] {OFF ON 0 1}	
	CALCulate:EVALuate:MSStatistics:DATA?	
SMTH/POLAR MENU		
REAL IMAG	CALCulate:EVALuate:R:FORMat RIMaginary	
LIN MAG PHASE	CALCulate:EVALuate:R:FORMat MLIPhase	
LOG MAG PHASE	CALCulate:EVALuate:R:FORMat MLOPhase	
R+jX	CALCulate:EVALuate:R:FORMat RX	
G+jB	CALCulate:EVALuate:R:FORMat GB	
SWR PHASE	CALCulate:EVALuate:R:FORMat SWRPhase	
MKR X AXIS [STIM]		
STIM	DISPlay[:WINDow]:TRACe:MARKer:UNIT SPARameter	
TIME	DISPlay[:WINDow]:TRACe:MARKer:UNIT TIME	
1/(2 π F)	DISPlay[:WINDow]:TRACe:MARKer:UNIT IOMega	
LEVEL MON [OFF]		
MONITOR: OFF	CALCulate:EVALuate:ON2 "OFF"	
AC-V	CALCulate:EVALuate:ON2 "ACV"	
AC-I	CALCulate:EVALuate:ON2 "ACC"	
DC-V (option)	CALCulate:EVALuate:ON2 "DCV"	
DC-I (option)	CALCulate:EVALuate:ON2 "DCC"	

Front Panel Key	Equivalent SCPI Command	Simple Command
System		
IBASIC		
Step	PROGrama[:SELEcted]:EXECute "STEP" ¹	
Continue	PROGrama[:SELEcted]:STATe CONTInue or; PROGrama[:SELEcted]:EXECute "CONT" ¹	
Run	PROGrama[:SELEcted]:STATe RUN or; PROGrama[:SELEcted]:EXECute "RUN" ¹	
Pause	PROGrama[:SELEcted]:STATe PAUSE or; PROGrama[:SELEcted]:EXECute "PAUSE" ¹	
Stop	PROGrama[:SELEcted]:STATe STOP or; PROGrama[:SELEcted]:EXECute "STOP" ¹	
Edit	PROGrama[:SELEcted]:EXECute "EDIT" ¹	
ASSIGN @Hp4291	(None)	
OUTPUT @Hp4291	(None)	
ENTER @Hp4291	(None)	
END	(None)	
GOTO LINE	(None)	
RECALL LINE	(None)	
END EDIT	(None)	
ON KEY LABELS		
[USER DEFINE]		
CAT		
SAVE		
RE-SAVE		
GET		
PURGE		
INITIALIZE		
MSI [INTERNAL/MEMORY]		
SCRATCH		
RENumber		
LIST		

¹ PROGRAM:EXPLICIT node can be used in the same way. See "PROGRAM Subsystem" in Chapter 9.

System

Front Panel Key	Equivalent SCPI Command	Simple Command
COMMAND ENTRY		
SELECT LETTER		
SPACE		
BACK SPACE		
ERASE TITLE		
DONE		
CANCEL		
CLEAR I/O		
RESET		
PROGRAM MENU		
USER FILE NAME		
PREV FILES		
NEXT FILES		
STORE DEV [DISK]		
MEMORY PARTITION	(None)	
SET CLOCK → See Set clock menu		
BEEPER MENU		
BEEP DONE ON off	SYSTem:BEEPer1:STATe {OFF ON 0 1}	
BEEP WARN ON off	SYSTem:BEEPer2:STATe {OFF ON 0 1}	
LIMIT MENU		
LIMIT LINE ON off	CALCulate:LIMit:LINE {OFF ON 0 1}	
LIMIT TEST ON off	CALCulate:LIMit:STATe {OFF ON 0 1}	
BEEP FAIL [OFF]		
BEEP: OFF	CALCulate:LIMit:BEEPer[:STATe] {OFF 0}	
PASS	CALCulate:LIMit:BEEPer:CONDition PASS CALCulate:LIMit:BEEPer[:STATe] {ON 1}	
FAIL	CALCulate:LIMit:BEEPer:CONDition FAIL CALCulate:LIMit:BEEPer[:STATe] {ON 1}	
EDIT LIMIT LINE → See Edit limit line menu		
LIMIT LINE OFFSETS		
STIMULUS OFFSET	CALCulate:LIMit:CONTRol:OFFSet <numeric>	
AMPLITUDE OFFSET	CALCulate:LIMit:OFFSet <numeric>	
MKR→ AMP.OFS.	CALCulate:LIMit:OFFSet MARKer	
LOGGING ON off	(None)	
SERVICE MENU → See Service Manual.		

Front Panel Key	Equivalent SCPI Command	Simple Command
Set clock menu TIME HH:MM:SS HOUR MIN SEC ENTER CANCEL DATE MM/DD/YY DATE DD/MM/YY MONTH DAY YEAR ENTER CANCEL DATE MODE:MonDayYear DayMonYear	SYSTem:TIME <hour>,<minute>,<second> (Enter <hour> parameter) (Enter <minute> parameter) (Enter <second> parameter) (None) (None) SYSTem:DATE <year>,<month>,<day> SYSTem:DATE <year>,<month>,<day> (Enter <month> parameter) (Enter <day> parameter) (Enter <year> parameter) (None) (None) SYSTem:DATE:MODE MDY SYSTem:DATE:MODE DMY	
Edit limit line menu SEGMENT EDIT STIMULUS MKR→STIMULUS UPPER LIMIT LOWER LIMIT DELTA LIMIT MIDDLE VALUE MKR→MIDDLE DONE DELETE ADD CLEAR LIST CLEAR LIST YES NO DONE	CALCulate:LIMit:SEGMENT <numeric> CALCulate:LIMit:SEGMENT:EDIT CALCulate:LIMit:SEGMENT:CONTRol[:DATA] <numeric> CALCulate:LIMit:SEGMENT:CONTRol[:DATA] MARKer CALCulate:LIMit:SEGMENT:UPPer <numeric> CALCulate:LIMit:SEGMENT:LOWer <numeric> CALCulate:LIMit:SEGMENT:DELTA <numeric> CALCulate:LIMit:SEGMENT:MIDDLE <numeric> CALCulate:LIMit:SEGMENT:MIDDLE MARKer CALCulate:LIMit:SEGMENT:SAVE CALCulate:LIMit:SEGMENT:DELEte CALCulate:LIMit:SEGMENT:ADD CALCulate:LIMit:CLEAr (None) CALCulate:LIMit:SAVE	

Local Preset Copy

Front Panel Key	Equivalent SCPI Command	Simple Command
<p>Local</p> <p>SYSTEM CONTROLLER (None)</p> <p>ADDRESSABLE ONLY (None)</p> <p>SET ADDRESSES ADDRESS:4291 (None)</p> <p>ADDRESS:CONTROLLER SYSTem:COMMunicate:GPIB:CONTroller:ADDRes <num>¹</p>		
<p>Preset</p> <p>SYSTem:PRESet</p>		
<p>Copy</p> <p>PRINT [STANDARD] HCOPY[:IMMediate]</p> <p>COPY ABORT HCOPY:ABORt</p> <p>COPY SKEY HCOPY:DEvIce:SKEY {OFF ON 0 1}</p> <p>COPY TIME ON off HCOPY:ITEm:TDSTamp:STATe {OFF ON 0 1}</p> <p>PRINT SETUP → See Print setup menu</p> <p>ORIENT [PORTRAIT] HCOPY:DEvIce:LANDScape {OFF ON 0 1}</p> <p>FORMFEED HCOPY:DEvIce:FORMFeed {OFF ON 0 1}</p> <p>LIST VALUES → See Screen menu DISPlay[:WINDow]:TEXT1:PAGE 1 DISPlay[:WINDow]:TEXT1:STATe {ON 1}</p> <p>OPERATING PARAMETERS → See Screen menu DISPlay[:WINDow]:TEXT2:PAGE 1 DISPlay[:WINDow]:TEXT2:STATe {ON 1}</p> <p>CAL KIT DEFINITION → See Screen menu DISPlay[:WINDow]:TEXT3:PAGE 1 DISPlay[:WINDow]:TEXT3:STATe {ON 1}</p> <p>COMPEN KIT DEFINITION → See Screen menu DISPlay[:WINDow]:TEXT4:PAGE 1 DISPlay[:WINDow]:TEXT4:STATe {ON 1}</p> <p>LIST SWEEP TABLE</p> <p>DISPLAY: LIST → See Screen menu (When DISP MODE: ST & SP) DISPlay[:WINDow]:TEXT5:PAGE 1 DISPlay[:WINDow]:TEXT5:STATe {ON 1}</p> <p>(When CTR & SPAN) DISPlay[:WINDow]:TEXT6:PAGE 1 DISPlay[:WINDow]:TEXT6:STATe {ON 1}</p> <p>DISP MODE: ST & SP (None, see DISPLAY LIST)</p> <p>CTR & SPAN (None, see DISPLAY LIST)</p> <p>LIMIT TEST TABLE</p> <p>DISPLAY: LIST → See Screen menu (When DISP MODE: UPR & LWR) DISPlay[:WINDow]:TEXT7:PAGE 1 DISPlay[:WINDow]:TEXT7:STATe {ON 1}</p> <p>(When MID & DLT) DISPlay[:WINDow]:TEXT8:PAGE 1 DISPlay[:WINDow]:TEXT8:STATe {ON 1}</p> <p>DISP MODE: ST & SP (None, see DISPLAY LIST)</p> <p>MID & DLT (None, see DISPLAY LIST)</p>		<p>PRSOFT</p> <p>LANDSCAPE</p> <p>FORMFEED</p>

1 <num> : <numeric>

Front Panel Key	Equivalent SCPI Command	Simple Command
Print setup menu		
PRINT STANDARD	HCOPY:DEvice:COLor {OFF 0}	
COLOR	HCOPY:DEvice:COLor {ON 1}	
PRINT COLOR [FIXED]	HCOPY:DEvice:CMAP:COLor FIXed	
PRINT COLOR [VARIABLE]	HCOPY:DEvice:CMAP:COLor VARiable	
DPI	HCOPY:DEvice:DPI	
DEFAULT SETUP	HCOPY:DEfault	DFLT
TOP MARGIN	HCOPY:DEvice:TOPMarg	TMARG
LFT MARGIN	HCOPY:DEvice:LEFTMarg	LMARG
Screen menu		
PRINT [STANDARD]	HCOPY:DEvice:LANGuage PCL HCOPY{:IMMediate}	
COPY ABORT	HCOPY:ABORt	
COPY TIME ON off	HCOPY:ITEM:TDSamp:STATe {OFF ON 0 1}	
NEXT PAGE	DISPlay[:WINDow]:TEXT{1-8}:PAGE UP	
PREV PAGE	DISPlay[:WINDow]:TEXT{1-8}:PAGE DOWN	
RESTORE DISPLAY	DISPlay[:WINDow]:TEXT{1-8}:STATe {OFF 0}	

Save

Front Panel Key	Equivalent SCPI Command	Simple Command
Save		
STATE		
(When STOR DEV [MEMORY])	MMEemory:STORe:STATe <file_name>,"MEMORY"	
(When STOR DEV [DISK])	MMEemory:STORe:STATe <file_name>,"DISK"	
DATA ONLY		
SAVE BINARY	MMEemory:STORe:TRACe SEL,<file_name>,<msus>	
(When STOR DEV [MEMORY])	MMEemory:STORe:TRACe SEL,<file_name>,"MEMORY"	
(When STOR DEV [DISK])	MMEemory:STORe:TRACe SEL,<file_name>,"DISK"	
SAVE ASCII	MMEemory:STORe:DINTerchange:TRACe SEL,<file_name>,<msus>	
DEFINE SAVE DATA → See Define save data menu		
STOR DEV []		
GRAPHICS	MMEemory:STORe:DINTerchange:TIFf <file_name>,<msus>	SAVDTIF <file_name>
4291A STATE		SAVDSTAC <file_name>
RE-SAVE FILE		
BACK UP MEMO DISK		STORMDISK
FILE UTILITIES → See File utilities menu		

Front Panel Key	Equivalent SCPI Command	Simple Command
Define save data menu RAW ON off RAW on OFF CAL ON off CAL on OFF DATA ON off DATA on OFF MEM ON off MEM on OFF DATA TRACE ON off DATA TRACE on OFF MEM TRACE ON off MEM TRACE on OFF USER TRACE ON off USER TRACE on OFF	MMEMory:STORe:ITEM:TRACe:SElect RAW MMEMory:STORe:ITEM:TRACe:DELeTe RAW MMEMory:STORe:ITEM:TRACe:SElect CCO MMEMory:STORe:ITEM:TRACe:DELeTe CCO MMEMory:STORe:ITEM:TRACe:SElect DATA MMEMory:STORe:ITEM:TRACe:DELeTe DATA MMEMory:STORe:ITEM:TRACe:SElect MEM MMEMory:STORe:ITEM:TRACe:DELeTe MEM MMEMory:STORe:ITEM:TRACe:SElect DTR MMEMory:STORe:ITEM:TRACe:DELeTe DTR MMEMory:STORe:ITEM:TRACe:SElect MTR MMEMory:STORe:ITEM:TRACe:DELeTe MTR MMEMory:STORe:ITEM:TRACe:SElect UTR MMEMory:STORe:ITEM:TRACe:DELeTe UTR	
File utilities menu PURGE FILE file_name (When STOR DEV [MEMORY]) (When STOR DEV [DISK]) PREV FILES NEXT FILES CREATE DIRECTORY CHANGE DIRECTORY COPY FILE file_name (When STOR DEV [MEMORY]) (When STOR DEV [DISK]) PREV FILES NEXT FILES INITIALIZE INITIALIZE MEMORY: YES (When FORMAT [LIF]) (When FORMAT [DOS]) INITIALIZE DISK: YES (When FORMAT [LIF]) (When FORMAT [DOS]) NO	MMEMory:DELeTe <file_name>,"MEMORY" MMEMory:DELeTe <file_name>,"DISK" (None) (None) MMEMory:CREate:DIRectory <string> MMEMory:CDIRectory <string> MMEM:COpy <file_name>,"MEMORY",<string>,<msus> ¹ MMEM:COpy <file_name>,"DISK",<string>,<msus> ¹ (None) (None) MMEMory:INITialize "MEMORY",LIF MMEMory:INITialize "MEMORY",DOS MMEMory:INITialize "DISK",LIF MMEMory:INITialize "DISK",DOS (None)	

1 MMEM:COPY : MMEMory:COPY

Recall

Front Panel Key	Equivalent SCPI Command	Simple Command
FORMAT [LIF]	(None, see INITIALIZE MEMORY: YES , INITIALIZE DISK: YES)	
FORMAT [DOS]	(None, see INITIALIZE MEMORY: YES , INITIALIZE DISK: YES)	
STOR DEV [MEMORY]	(None, see INITIALIZE MEMORY: YES)	STODMEMO
STOR DEV [DISK]	(None, see INITIALIZE DISK: YES)	STODDISK
<p>Recall</p> <p>file_name</p> <p>(When STOR DEV [MEMORY], state)</p> <p>(When STOR DEV [MEMORY], data)</p> <p>file_name</p> <p>(When STOR DEV [DISK], state)</p> <p>(When STOR DEV [DISK], data)</p> <p>file_name</p> <p>file_name</p> <p>PREV FILES</p> <p>NEXT FILES</p> <p>STOR DEV [MEMORY]</p> <p>STOR DEV [DISK]</p>	<p>MMEemory:LOAD:STATe <file_name>,"MEMORY"</p> <p>MMEemory:LOAD:TRACe SEL,<file_name>,"MEMORY"</p> <p>MMEemory:LOAD:STATe <file_name>,"DISK"</p> <p>MMEemory:LOAD:TRACe SEL,<file_name>,"DISK"</p> <p>(None)</p> <p>(None)</p> <p>(None, see file_name)</p> <p>(None, see file_name)</p>	

GPIB Only Functions and the GPIB Commands

Function	Equivalent SCPI Command
Calibration related commands Query the calibration has performed	SENSe:CORRection1:STATe?
Fixture compensation related commands Set the fixture compensation standard arrays Clear the fixture compensation standard arrays	DATA:DEFine {OADM SIMP LIMP},{<numeric> DATA DTR TR1} DATA:DELete {OADM SIMP LIMP}
Data arrays related commands Query the data existing arrays	MMEMory:ITEM:TRACe:CATalog?
Data transfer related commands Transfer calibration coefficient array Transfer compensation coefficient array Transfer compensation open standard array Transfer compensation short standard array Transfer compensation load standard array Transfer raw data array Transfer data array Transfer level monitor array Transfer data of the specified point Transfer memory array Transfer memory of the specified point Transfer the level monitor value of the specified point Transfer the data trace array (active channel) (channel 1) (channel 2) Transfer the data trace value of the specified point (active channel) (channel 1) (channel 2) Transfer the memory trace array (active channel) (channel 1) (channel 2)	DATA[:DATA] CCO{11-33},<block> DATA[:DATA] CMP{1-3},<block> DATA[:DATA] OADM,<block> DATA[:DATA] SIMP,<block> DATA[:DATA] LIMP,<block> DATA[:DATA] RAW,<block> DATA[:DATA] DATA,<block> DATA[:DATA] MON,<block> DATA[:DATA]:VALue? DATA,<numeric> DATA[:DATA]? MEM ¹ DATA[:DATA]:VALue? MEM,<numeric> ¹ DATA[:DATA]:VALue? MON,<numeric> TRACe[:DATA] DTR,<block> TRACe[:DATA] TR1,<block> TRACe[:DATA] DTRCH1,<block> TRACe[:DATA] TRICH1,<block> TRACe[:DATA] DTRCH2,<block> TRACe[:DATA] TRICH2,<block> TRACe[:DATA]:VALue? DTR,<numeric> TRACe[:DATA]:VALue? TR1,<numeric> TRACe[:DATA]:VALue? DTRCH1,<numeric> TRACe[:DATA]:VALue? TRICH1,<numeric> TRACe[:DATA]:VALue? DTRCH2,<numeric> TRACe[:DATA]:VALue? TRICH2,<numeric> TRACe[:DATA] MTR ¹ TRACe[:DATA] TR{2-17} TRACe[:DATA] TR{2-17}CH1,<block> TRACe[:DATA] MTRCH1,<block> ¹ TRACe[:DATA]:VALue? TR{2-17}CH2,<block> TRACe[:DATA]MTRCH2,<block> ¹

¹ Selected memory trace

GPIB Only Functions and the GPIB Commands

Function	Equivalent SCPI Command
Data transfer related commands (Continued) Transfer the memory trace value of the specified point (active channel) (channel 1) (channel 2) Transfer stimulus array Transfer stimulus of the specified point	TRACe[:DATA]:VALue? TR{2-17} TRACe[:DATA]:VALue? MTR,<numeric> ¹ TRACe[:DATA]:VALue? TR{2-17}CH1,<numeric> TRACe[:DATA]:VALue? MTRCH1,<numeric> ¹ TRACe[:DATA]:VALue? TR{2-17}CH2,<numeric> TRACe[:DATA]:VALue? MTRCH2,<numeric> ¹ DATA[:DATA]? SPAR DATA[:DATA]:VALue? SPAR,<numeric>
Data transfer format related commands Set the data transfer format	FORMAT[:DATA] { ASCii REAL,32 REAL,64 PACKed,32}
Error related commands Query error	SYSTem:ERRor?
Key related commands Operate front panel key codes Lock front panel key	SYSTem:KEY <numeric> SYSTem:KLOCK {OFF ON 0 1}
Limit test related commands Query the limit test fail points Query the limit test result list Query the limit test results on marker Query the number of the limit test fail point	DATA[:DATA]? LFA DATA[:DATA]? LLIS DATA[:DATA]? LMAR DATA:POINT? LFA
HP instrument BASIC control commands Query the program name Create the program Scratch the program Set the contents of the program variables and arrays Set the contents of the string variables and arrays (No function)	PROGram:CATalog? PROGram[:SELEcted]:DEFine <block> ² PROGram[:SELEcted]:DELete[:SELEcted] or, PROGram[:SELEcted]:DELete:ALL ² PROGram[:SELEcted]:NUMBer <var>,<numeric1>,<numeric2>,... ²³ PROGram[:SELEcted]:STRing <var>,<string1>,<string2>,... ²⁴ PROGram[:SELEcted]:MALLocate {<numeric> DEFault} or, PROGram[:SELEcted]:NAME <string> ²
Marker Related commands Query marker values (complex) (real part) (imaginary part) Move the marker to the specified stimulus Move the marker to the specified point	CALCulate:EVALuate:Y{[1] 2-8}:DATA? {CH1 CH2} CALCulate:EVALuate:Y{[1] 2-8}:VALue1? {CH1 CH2} CALCulate:EVALuate:Y{[1] 2-8}:VALue2? {CH1 CH2} CALCulate:EVALuate:Y{[1] 2-8}:XPOSITION <numeric> CALCulate:EVALuate:Y{[1] 2-8}:XPOSITION:POINT <numeric>

1 Selected memory trace

2 PROGram:EXPLict node can be used in the same way. See "PROGram Subsystem" in Chapter 3.

3 More than one <numeric> parameters can be sent.

4 More than one <string> parameters can be sent.

GPIB Only Functions and the GPIB Commands

Function	Equivalent SCPI Command
Marker Related commands (Continued) Query the Δ marker values Query the bandwidth parameters Query the marker statistics	CALCulate:EVALuate:REFerence:DATA? CALCulate:EVALuate:WIDTh:DATA? CALCulate:EVALuate:MSStatistics:DATA?
Measurement parameter related commands Query the available conversion	CALCulate:MATH1[:EXPReSSion]:CATaLog?
Status byte related commands Clear the status bytes Set the status byte enable register Query the contents of status byte register Set the standard event status enable register Query the contents of the ESR ¹ Set OPC ² bit when all operation completed Set the instrument event status enable register Query the instrument event status register Set the operation status enable register Set the operation status event register Query the operation status condition register Set the negative transition filter Set the positive transition filter Set the questionable status enable register Query the questionable status event register Query the questionable status condition register	*CLS *SRE *STB? *ESE *ESR? *OPC STATus:INSTrument:ENABle <numeric> STATus:INSTrument[:EVENT]? STATus:OPERation:ENABle STATus:OPERation[:EVENT]? STATus:OPERation:CONDition? STATus:OPERation:NTRansition STATus:OPERation:PTRansition STATus:QUEStionable:ENABle STATus:QUEStionable[:EVENT]? STATus:QUEStionable:CONDition?
System related commands Query the firmware version Query the option installed Reset the system Execute the internal test Set the address to pass control back Query whether the external reference is connected Query the power-on test result LCD back light on/off	*IDN? or DIAGnostic:FREVision? *OPT? *RST *TST? *PCB DIAGnostic:EREference:STATe? DIAGnostic:INIT:RESult? DISPlay:BACKlight
Data math related commands Query the available data math expression	CALCulate:MATH2[:EXPReSSion]:CATaLog?
Trigger related commands Trigger when the bus trigger is selected	*TRG
Wait commands Pause until preceding command completed	*WAI
User trace related commands Transfer the user trace arrays Set the number of points of the user trace	TRACe[:DATA] {TRX{18-21} TRY{18-21}} TRACe:POINts TR{18-21}{,<numeric>}

1 standard event status register

2 Operation Complete

GPIB Only Functions and the GPIB Commands

Function	Equivalent SCPI Command
8 bit I/O related commands Read data from 8 bit I/O Write data from 8 bit I/O	SYSTem:COMMunicate:PARallel[:RECeive]:DATA? SYSTem:COMMunicate:PARallel:TRANsmit:DATA <numeric>
File transfer commands Returns a file to access-disabled status Returns the name of the current directory Returns the file name Returns the number of the files in the directory Returns the size of a specified file in bytes Reads data from a file Makes a specified file read-enabled Makes a specified file write-enabled Writes data in a file	CLOSE CWD? FNAME? <numeric> FNUM? FSIZE? <string> READ? ROPEN <string> WOPEN <string>[,<numeric>] WRITE <block>

SCPI Conformance Information

This chapter provides information about SCPI conformance. It also provides a list of all the GPIB commands implemented in the analyzer.

SCPI Version

The 4291B conforms to the SCPI 1993 standard.

GPIB Commands Compatible to SCPI

Conventions	Definition
< >	Words or characters enclosed in < > brackets are used to symbolize a variable parameter.
[]	Words enclosed in the [] brackets can be omitted.
{ }	Several codes are enclosed in the { } brackets and one of these codes can be selected. For example, {OFF ON 0 1} means OFF, ON, 0, or 1. {1-4} means 1, 2, 3, or 4.

Upper case characters represent the command that must appear exactly as shown with no embedded spaces. Lower case characters can be omitted.

The following two tables list the SPCI-compatible commands implemented in the 4291B. Table D-1 lists the IEEE 488.2 common commands. Table D-2 lists the instrument control commands.

Table D-1. IEEE 488.2 Common Commands

*CLS	*RST
*ESE <numeric>	*SRE <numeric>
*ESR?	*STB?
*IDN?	*TST?
*OPC	*WAI
*PCB <numeric>	

In Table D-2, the commands noted (original) are not part of the SCPI definition, but are unique to the 4291B.

GPIB Only Functions and the GPIB Commands

Table D-2. Instrument Control Commands

COMMAND	PARAMETER	NOTES
ABORt		[no query]
[CALCulate Subsystem]		
CALCulate		
:EVALuate		
:BAND		
:FULL		
:STATe	{OFF ON 0 1}	
:SPAN	DMARKer	
:STARt	MARKer	
:STOP	MARKer	
:COUPle	{OFF ON 0 1}	
:EFFect		
:ON	{1 2}	
:EPARAmeters		[no query] (original)
:CIRCuit	{A B C D E}	(original)
:SIMulation		[no query] (original)
:INTerpolate	{OFF ON 0 1}	
:MSTATistics		
:DATA?		[query only]
:STATe	{OFF ON 0 1}	
:ON1	"TR{1-21}"	
:ON2	{"OFF" "ACV" "ACC" "DCV" "DCC"}	
:PEAK		
:EXCURsion		
:X	{<numeric> DMARKer}	
:Y	{<numeric> DMARKer}	
:POLarity	{POSitive NEGative}	
:THReshold	{<numeric> MARKer}	
:STATe	{OFF ON 0 1}	
:R		
:FORMat	{RIMaginary MLIPhase MLOPhase RX GB SWRPhase}	
:REFerence		(original)
:DATA?		[query only] (original)
:X	<numeric>	(original)
:Y[1]	<numeric>	(original)
:Y2	<numeric>	(original)
:WIDTh		
:DATA?		[query only] (original)
:STATe	{OFF ON 0 1}	(original)
:XPOSition		
:IN		[no query] (original)
:OUT		[no query] (original)
:Y	{DIVS2 MULS2 DIV2 FIXed[,<numeric>]}	(original)
:Y{1-8}	(Y[1]: Main marker, Y{2-8}: Sub marker)	
:DATA?	[CH1 CH2]	[query only] (original)
:VALue{1 2}?	[CH1 CH2]	[query only] (original)
:XPOSition	<numeric>	
:LPEak ¹		
:LTARget ¹		
:MAXimum ¹		
:MINimum ¹		
:NPEak ¹		
:PEAK ¹		
:POINt	<numeric>	
:RPEak ¹		
:RTARget ¹		
:TARGet ¹	<numeric>	
:TRACK ¹	{MAXimum MINimum TARGet PEAK OFF}	

¹ These nodes are only for Y[1]:XPOSition

GPIB Only Functions and the GPIB Commands

Table D-2. Instrument Control Commands (continued)

COMMAND	PARAMETER	NOTES
CALCulate (Continued)		
:FORMat	{MLINear PHASe UPHase REAL IMAGinary LFACTor LTANgent CP CS LP LS D Q RP RS COMPLex}	(original)
:UNIT		
:ANGLE	{DEG RAD}	
:LIMit		
:BEEPer		
:CONDition	{PASS FAIL}	(original)
[:STATe]	{OFF ON 0 1}	(original)
:CLEAr		[no query] (original)
:CONTrol		
:OFFSet	<numeric>	(original)
:LINE	{OFF ON 0 1}	(original)
:OFFSet	{<numeric> MARKer}	(original)
:SAVE		[no query] (original)
:SEGment	<numeric>	(original)
:ADD		[no query] (original)
:CONTrol		
[:DATA]	{<numeric> MARKer}	(original)
:DELete		[no query] (original)
:DELta	<numeric>	(original)
:EDIT		[no query] (original)
:LOWer	<numeric>	(original)
:MIDDLE	{<numeric> MARKer}	(original)
:SAVE		[no query] (original)
:UPPer	<numeric>	(original)
:STATe	{OFF ON 0 1}	
:MATH1		
:DIMension1	<numeric>	
:DIMension2	<numeric(in)>,<numeric(out)>,<numeric(hei)> ¹	
[:EXPReSSion]		
:CATalog?		[query only]
:NAME	{ADM DCO PER RCO}	
:STATe	{OFF ON 0 1}	
:MATH2		
[:EXPReSSion]		
:CATalog?		[query only]
:NAME	{SUB ADD DIV MUL}	
:STATe	{OFF ON 0 1}	

¹ <numeric(in)>: inner dimension, <numeric(out)>: outer dimension, <numeric(hei)>: height

GPIB Only Functions and the GPIB Commands

Table D-2. Instrument Control Commands (continued)

COMMAND	PARAMETER	NOTES
[DATA Subsystem]		
DATA		
:DATA	{AOFF GAIN MZAP}, <numeric> OFFS, {<numeric> MARKer} <array>, {<block> <numeric1>, <numeric2>, ..., <numeric n2>} ¹ {EQC0 EQC1 EQL1 EQR1}, <numeric>	(original)
:DATA?	LFA LLIS LMAR MEM SPAR	[query only] [query only] [query only] [query only] [query only]
:VALue?	{SPAR DATA MEM MON}, <numeric>	[query only]
	(AOFF : aux offset CCO{11-33} : calibration coefficient CMP{1-3} : compensation coefficient DATA : data array (complex) EQ{C0 C1 L1 R1} : equivalent circuit parameters GAIN : gain LFA : limit test fail points LLIS : limit test result list LMAR : limit test result on marker LIMP : fixture compensation load standard array MEM : selected memory array (complex) MON : monitor data array MZAP : zooming aperture OADM : fixture compensation open standard array OFFS : offset RAW : raw data array SPAR : stimulus array SIMP : fixture compensation short standard array)	
:DEFine	{OADM SIMP LIMP}, {<numeric> DATA DTR TR1}	
:DELete	{OADM SIMP LIMP}	[no query]
:POINT?	LFA	[query only]
[DIAGnostic Subsystem]		
DIAGnostic		
:EREFerence		[query only]
:STATe?		[query only]
:FREVision?		[query only]
:INIT		
:RESult?		[query only]
:SERVice		
:BUS		
:DC	<numeric>	
:FREQuency	<numeric>	
:STATe	{OFF ON 0 1}	
:CCONstant		
:DCBias	{OFF ON 0 1}	
:OLEVel	{NORMal FRONT OFF}	
:DCBias		
:MODE	{AUTO MANual}	
:OFFSet	{AUTO POSitive NEGative}	
:POLarity	{AUTO POSitive NEGative}	
:STATe	{OFF ON 0 1}	

1 <array> is {CCO{11-33}|CMP{1-3}|DATA|RAW|OADM|SIMP|LIMP|MON}.

GPIB Only Functions and the GPIB Commands

Table D-2. Instrument Control Commands (continued)

COMMAND	PARAMETER	NOTES
DIAGnostic		
:SERVice		
:DCBias (Continued)		
:GDAC		
:MODE	{AUTO MANual}	
:VALue	<numeric>	
:IDAC		
:MODE	{AUTO MANual}	
:VALue	<numeric>	
:VDAC		
:MODE	{AUTO MANual}	
:VALue	<numeric>	
:DWELL		
:TRD		
:MPX	<numeric>	
:IF		
:GAIN		
:MODE	{AUTO MANual}	
:X		
:XI	{DB0 DB6 DB12}	
:XV	{DB0 DB6 DB12}	
:Y	{DB0 DB6 DB12 DB18}	
:Z	{DB0 DB2 DB4 DB18}	
:MODE	{OFF ON 0 1}	
:SOURce		
:ATTenuator	{AUTO DB0 DB10 DB20 DB30 DB40 DB50 DB60}	
:LEVel		
:DAC		
:MODE	{AUTO MANual}	
:VALue	<numeric>	
:MODE	{AUTO MANual}	
:STATE	{OFF ON 0 1}	
:SYNThesizer		
:FLOCAl		
:MODE	{AUTO SINGle TRIPle}	
:FN		
:MODE	{AUTO NARRow WIDE}	
:FREQuency		
:OFFSet	<numeric>	
:STEP		
:DAC		
:MODE	{AUTO MANual}	
:VALue	<numeric>	
:LOOP	{OPEN CLOSE}	
:MODE	{AUTO MANual}	
:OUTPut	{OFF ON 0 1}	
:POLarity	{AUTO POSitive NEGative}	
:TRANsducer		
:GAIN	{DB0 DB10}	
:GIATtenuator	{DB0 DB18}	
:GVATtenuator	{DB0 DB18}	
:THEAd	{HIMP LIMP HTHI HTLI NONE}	
:MODE	{AUTO MANual}	
:RANGe	{AUTO EXPand NORMal}	
:SENSe		
:FUNCTION	{IMPedance VOLTage CURRent VIRatio}	
:TEST	<numeric>	[no query]
:CONTINUE		[no query]
:EXECute		[no query]
:RESult?	<numeric>	[query only]
:SPEC	{CUSTomer PRODuction}	

GPIB Only Functions and the GPIB Commands

Table D-2. Instrument Control Commands (continued)

COMMAND	PARAMETER	NOTES
[DISPlay Subsystem]		
DISPlay		
:ANNotation		
:FREQuency	{OFF ON 0 1}	
:BACKlight	{OFF ON 0 1}	
:BRIGHtness	<numeric>	
:CMAp		
:COLor{1-14} ¹		
:DEFault		[no query] (original)
:HSL	<numeric(Hue)>,<numeric(Sat)>,<numeric(Lum)>	
:DEFault		[no query]
:LOAD		[no query] (original)
:STORe		[no query] (original)
:CONTrast	<numeric>	
[:WINDow]		
:ALLocation	{INSTrument HIHB BASic BSTatus}	
:FORMat	{FBACk ULOWer}	(original)
:GRAPhics		
:STATe	{OFF ON 0 1}	
:TEXT{1-40} ²		
:CLEar	(only for TEXT{11-30})	[no query]
:COLor	<numeric> (only for TEXT{11-30})	
[:DATA]	<string> (only for TEXT{10-38})	
:LOCate	<numeric(x)>[,<numeric(y)>] (only for TEXT{11-30})	
:PAGE	{UP DOWN <numeric>} (except TEXT{9-40})	
:STATe	{OFF ON 0 1} (except TEXT{10-38}; TEXT{10-38} is always ON)	
:TRACe{1-21}	(1:DataTrace, 2-17:MemoryTrace, 18-21:UserTrace)	
:CLEar	(only for TRACe{2-21})	[no query] (original)
:GRATicule		
:AXIS		
COUple	{OFF ON 0 1} (only for TRACe{18-21})	
:FORMat	{RECTangle POLar SMITH ADMittance CPLane}	
:GRID		
[:STATe]	{OFF ON 0 1} (only for TRACe{1-17})	
:MARKer{1-8}	(1:MainMarker, 2-8:SubMarker)	
:ALL ³	DEFault	(original)
:STATe	{OFF ON 0 1}	(original)
:RELative ³	{OFF ON 0 1}	(original)
:REFerence	{FIXed MARKer TRACkEd}	(original)
:STATe	{OFF ON 0 1} (only for MARKer{2-8})	(original)
:UNIT	{SPARAmeter TIME OMega} (only for TRACe{1-17})	(original)
:STATe	{OFF ON 0 1}	
:X		
[:SCALe]		
:LEFt	<numeric> (only for TRACe{18-21})	
:RIGHt	<numeric> (only for TRACe{18-21})	
:RLEVel	<numeric> (except TRACe{18-21})	
:SPACing	{LINear LOGarithmic OBASe} ⁴	(original)
:UNIT	<string> (only for TRACe{18-21})	(original)

1 1:Ch1Data, 2:Ch1Mem/LimitLine, 3:Ch2Data, 4:Ch2Mem/LimitLine, 5:Graticule, 6:Warning, 7:Text/Marker, 8:IBASIC, 9-14:Pen1-6

2 1:ListValue, 2:OperationParameter, 3:CalStd, 4:CompenStd, 5:STArT&STOP (List), 6:CENTer&SPAN (List), 7:UPPer&LOWer (LimitTest), 8:MID&DELT (LimitTest), 9:MarkerList, 10:Title, 11-30:Label, 31-34:UserTraceHeadline, 35-38:UserTraceFootnote, 39:EquivCircuitParameter, 40:EquivalentCircuit

3 Only for MARKer[1]

4 OBASe is only for TRACe{1-17}:X:SPACing

GPIB Only Functions and the GPIB Commands

Table D-2. Instrument Control Commands (continued)

COMMAND	PARAMETER	NOTES
[INSTRUMENT Subsystem]		
INSTRUMENT		
:COUPLE	{ALL NONE}	(original)
:NSELECT	{1 2}	
[:SELECT]	{CH1 CH2}	
:STATE	{OFF ON 0 1}	
[MMEMORY Subsystem]		
MMEMORY		
:CDIRECTORY	<string>	[no query]
:COPY	{<string(s)>,<string(m_s)>,<string(d)>,<string(m_d)>} ¹	[no query]
:CREATE		
:DIRECTORY	<string>	[no query] (original)
:DELETE	<string(file_name)>[,<string(m)>]	[no query]
:INITIALIZE	<string>,{LIF DOS}	[no query]
:LOAD		
:STATE	<string(file_name)>[,<string(m)>]	(original)
:TRACE	SEL,<string(file_name)>[,<string(m)>]	
:STORE		
:DINTERCHANGE		
:TIFF	<string(file_name)>[,<string(m)>]	[no query]
:TRACE	SEL,<string(file_name)>[,<string(m)>]	
:ITEM		
:TRACE		
:CATALOG?		[query only] (original)
:DELETE	{CCO DATA DTR UTR MEM MTR RAW}	[no query] (original)
:SELECT	{CCO DATA DTR UTR MEM MTR RAW}	[no query] (original)
:STATE	<string(file_name)>[,<string(m)>]	(original)
:TRACE	SEL,<string(file_name)>[,<string(m)>]	
[PROGRAM Subsystem]		
PROGRAM		
:CATALOG?		[query only]
[:SELECTED]		
:DEFINE	<block>	
:DELETE		
[:SELECTED]		[no query]
:ALL		[no query]
:EXECUTE	<string>	[no query]
:MALLOCATE	{<numeric> DEFAULT}	
:NAME	<string>	
:NUMBER	<var>,<numeric1>,<numeric2>,...,<numeric n>	
:STATE	{RUN PAUSE STOP CONTINUE}	
:STRING	<var>,<string1>,<string2>,...,<string n>	
:WAIT		
:EXPLICIT		
:DEFINE	"PROG",<block>	
:DELETE	"PROG"	[no query]
:EXECUTE	"PROG",<string>	[no query]
:MALLOCATE	"PROG",{<numeric> DEFAULT}	
:NAME	"PROG",<string>	
:NUMBER	"PROG",<var>,<numeric1>,<numeric2>,...,<numeric n>	
:STATE	"PROG",{RUN PAUSE STOP CONTINUE}	
:STRING	"PROG",<var>,<string1>,<string2>,...,<string n>	
:WAIT	"PROG"	

¹ (s) means the source file, and (d) means the destination file.

(m_s) means the source mass storage, and (m_d) means the destination mass storage.

GPIB Only Functions and the GPIB Commands

Table D-2. Instrument Control Commands (continued)

COMMAND	PARAMETER	NOTES
[SENSe Subsystem]		
SENSe		
:AVERage1		
:COUNt	<numeric>	
[:STATe]	{OFF ON 0 1}	
:AVERage2		
:CLEAr		[no query] (original)
:COUNt	<numeric>	
[:STATe]	{OFF ON 0 1}	
:CORRection1		
:CKIT	{APC7 UDEFined}	(original)
:LABel	<string>	(original)
:SAVE		[no query]
:STANdard1		
:C	<numeric>	(original)
:G	<numeric>	(original)
:STANdard2		
:L	<numeric>	(original)
:R	<numeric>	(original)
:STANdard3		
:R	<numeric>	(original)
:X	<numeric>	(original)
:COLLect		
[:ACQuire]	{STANdard1 STANdard2 STANdard3 STANdard4}	
:FPOints	{FIXed USER}	(original)
:SAVE		(original)
:EDELay		
:STATe	{OFF ON 0 1}	
[:TIME]	<numeric>	
[:STATe]?		[query only]
:CORRection2		
:CKIT[1]		
:LABel	<string>	
:SAVE		
:STANdard1		
:C	<numeric>	(original)
:G	<numeric>	(original)
[:SELect]	{LIST LPARAmeter}	(original)
:STANdard2		
:L	<numeric>	(original)
:R	<numeric>	(original)
[:SELect]	{LIST LPARAmeter}	(original)
:STANdard3		
:L	<numeric>	(original)
:R	<numeric>	(original)
[:SELect]	{LIST LPARAmeter}	(original)
:CKIT2	{TEFLon UDEFined}	(original)
:LABel	<string>	(original)
:SAVE		[no query]
:STANdard6		
:PERal	<numeric>	(original)
:PLFactor	<numeric>	(original)
:THICKness	<numeric>	(original)

GPIB Only Functions and the GPIB Commands

Table D-2. Instrument Control Commands (continued)

COMMAND	PARAMETER	NOTES
SENSe		
:CORRection2 (Continued)		
:COLLect		
[:ACQuire]	STANdard{1-7}	
:FPOints	{FIXed USER}	
:SAVE		
:OPEN		
[:STATe]	{OFF ON 0 1}	
:SHORt		
[:STATe]	{OFF ON 0 1}	
:LOAD		
[:STATe]	{OFF ON 0 1}	
:FREQuency		
:CENTer	{<numeric> DMARKer MARKer TPEak}	
:MODE	{FIXed LIST SWEep}	
:SPAN	{<numeric> DMARKer MZAPerture}	
:STARt	{<numeric> MARKer}	
:STOP	{<numeric> MARKer}	
:LIST		
:CLEar		[no query] (original)
:SAVE		[no query] (original)
:SEGMENT	< numeric >	(original)
:ADD		[no query] (original)
:AVERage		
:COUNT	< numeric >	(original)
:CURRent	< numeric >	(original)
:DELeTe		[no query] (original)
:EDIT		[no query] (original)
:FREQuency		
:CENTer	< numeric >	(original)
:SPAN	< numeric >	(original)
:STARt	{<numeric> MARKer}	(original)
:STOP	{<numeric> MARKer}	(original)
:POINts	< numeric >	(original)
:POWer	< numeric >	(original)
:QUIT		[no query] (original)
:SAVE		[no query] (original)
:VOLTage	< numeric >	(original)
:SWEep		
:COUNT	< numeric >	
:DWELl{1 2}	< numeric >	
:AUTO	{OFF ON 0 1}	
:POINts	< numeric >	
:SPACing	{LlNear LOGarithmic}	
:TIME	< numeric >	
:AUTO	{OFF ON 0 1}	

GPIB Only Functions and the GPIB Commands

Table D-2. Instrument Control Commands (continued)

COMMAND	PARAMETER	NOTES
[SOURce Subsystem]		
SOURce1	(1:ac-source)	
:FREQuency		
[:CW :FIXed]	<numeric>	
:{CURRent POWer}		
[:LEVel]		
[:IMMediate]		
[:AMPLitude]	<numeric>	
:SWEep		
:DIRection	{UP DOWN}	
:SPACing	{LINear LOGarithmic}	
:VOLTag		
:CENTer	{<numeric> DMARker MARKer TPEak}	
[:LEVel]		
[:IMMediate]		
[:AMPLitude]	<numeric>	
:MODE	{FIXed LIST SWEep}	
:SPAN	{<numeric> DMARker MZAPErture}	
:STARt	{<numeric> MARKer}	
:STOP	{<numeric> MARKer}	
SOURce2	(2:dc-source, option 001 only)	
:{CURRent VOLTag}		
:ALC		
[:STATe]	{OFF ON 0 1}	
:CENTer	{<numeric> DMARker MARKer TPEak}	
[:LEVel]		
[:IMMediate]		
[:AMPLitude]	<numeric>	
:LIMit		
[:AMPLitude]	<numeric>	
:MODE	{FIXed SWEep}	
:SPAN	{<numeric> DMARker MZAPErture}	
:STARt	{<numeric> MARKer}	
:STATe	{OFF ON 0 1}	(original)
:STOP	{<numeric> MARKer}	
:SWEep		
:DIRection	{UP DOWN}	
:SPACing	{LINear LOGarithmic}	

GPIB Only Functions and the GPIB Commands

Table D-2. Instrument Control Commands (continued)

COMMAND	PARAMETER	NOTES
[STATus Subsystem] STATus		
:INSTRument		
:ENABLE	<numeric>	(original)
[:EVENT]?		[query only] (original)
:OPERation		
:CONDition?		[query only]
:ENABLE	<numeric>	
[:EVENT]?		[query only]
:NTRansition	<numeric>	
:PTRansition	<numeric>	
:PRESet		[no query]
:QUEStionable		
:CONDition?		[query only]
:ENABLE	<numeric>	
[:EVENT]?		[query only]
[SYSTem Subsystem] SYSTem		
:BEEPer{ 1 2}		
:STATe	{OFF ON 0 1}	
:COMMunicate		
:GPIB		
:CONTRoller		
:ADDReSS	<numeric>	(original)
:PARAllel		
[:RECEive]		
:DATA?		[query only] (original)
:TRANsmit		
:DATA	<numeric>	(original)
:DATE	<numeric>, <numeric>, <numeric>	
:MODE	{MDY DMY}	(original)
:ERRor?		[query only]
:FIXTure	{NONE HP16191 HP16192 HP16193 HP16453 HP16454S HP16454L UDEFined}	(original)
:DISTance	<numeric>	(original)
:LABel	<string>	(original)
:SAVE		[no query] (original)
:KEY	<numeric>	
:KLOCK	{OFF ON 0 1}	
:PRESet		[no query]
:SECurity		
[:STATe]	{ON 1}	
:TIME	<numeric(hour)>, <numeric(min)>, <numeric(sec)>	
:VERSion?		[query only]

GPIB Only Functions and the GPIB Commands

Table D-2. Instrument Control Commands (continued)

COMMAND	PARAMETER	NOTES
[TRACe Subsystem]		
TRACe		
:COPY	TR{2-17},TR1 ¹	[no query]
	TR{18-21},TR{1-17} ¹	[no query]
[:DATA]	<trace>,{<block> <numeric1>,<numeric2>,...,<numeric n2>} ²	
	{TRX{18-21} TRY{18-21}},{<block> <numeric1>,...,<numeric n>} ³	
:VALue?	<trace>,<numeric> ²	[query only]
:POINts	TR{18-21}[,<numeric>]	
[TRIGger Subsystem]		
TRIGger		
:EVENT		
:TYPE	{POINT SWEep}	(original)
:SLOPe	{POSitive NEGative}	
:SOURce	{BUS EXTernal INTernal MANual}	

1 TR1 : data trace

TR{2-17} : memory trace{1-16}

TR{18-21} : user trace{1-4}

2 <trace> is {DTR|DTRCH{1|2}|MTR|MTRCH{1|2}|TR{1-17}|TR{1-17}CH{1|2}}

Where,

DTR : data trace of active channel

DTRCH{1|2} : data trace of channel{1|2}

MTR : selected memory trace of active channel

MTRCH{1|2} : selected memory trace of Channel{1|2}

TR1 : data trace of active channel

TR{2-17} : memory trace of active channel

TR1CH{1|2} : data trace of channel{1|2}

TR{2-17}CH{1|2} : memory trace of channel{1|2}

3 TRX{18-21} : x-array of user trace{1-4}

TRY{18-21} : y-array of user trace{1-4}

GPIB Only Functions and the GPIB Commands

Simple Commands

The following table lists the simple commands implemented in the 4291B. The simple commands are unique to the analyzer.

Table D-3. Simple Commands

BLIGHT	{OFF ON 0 1}	
CHAN1		[query returns {0 1}]
CHAN2		[query returns {0 1}]
CENT	<numeric>	
CLEM		[no query]
CONT		[query returns {0 1}]
DATMEM		[no query]
DFLT		[no query]
DISP	{DATA MEMO DATM}	
DISSMEMO	{OFF ON 0 1}	
DPI	<numeric>	
DUAC	{OFF ON 0 1}	
DUAM	{IMPH IRIM APPH ARIM LSQ LPQ CSD CPD DMPH DRIM PMPH PRIM}	
FMT	{LINY LOGY POL SMIT ADM COMP}	
FORMFEED	{OFF ON 0 1}	
FREQ		[query returns {0 1}]
HOLD		[query returns {0 1}]
LANDSCAPE	{OFF ON 0 1}	
LISD	{FBAS OBAS}	
LMARG	<numeric>	
MARDCENT		[no query]
MARDSPAN		[no query]
MARK	{OFF ON 0 1}	
MARKCENT		[no query]
MARKREF		[no query]
MARKSTAR		[no query]
MARKSTOP		[no query]
MARZ		[no query]
MATH	{DATA DMNM DPLM DDVM DMLM}	
MEAS	{IMAG IPH IRE IIM AMAG APH ARE AIM RCM RCPH RCR RCIM DCM DCPH DCR DCIM PHMA PPH PRE PIM CP CS LP LS D Q RP RS}	
NUMG	<numeric>	[no query]
PEAKCENT		[no query]
POIDTIME	<numeric>	
PRSOFT	{OFF ON 0 1}	
RESTMDISK		[no query]

GPIB Only Functions and the GPIB Commands

Table D-3. Simple Commands (continued)

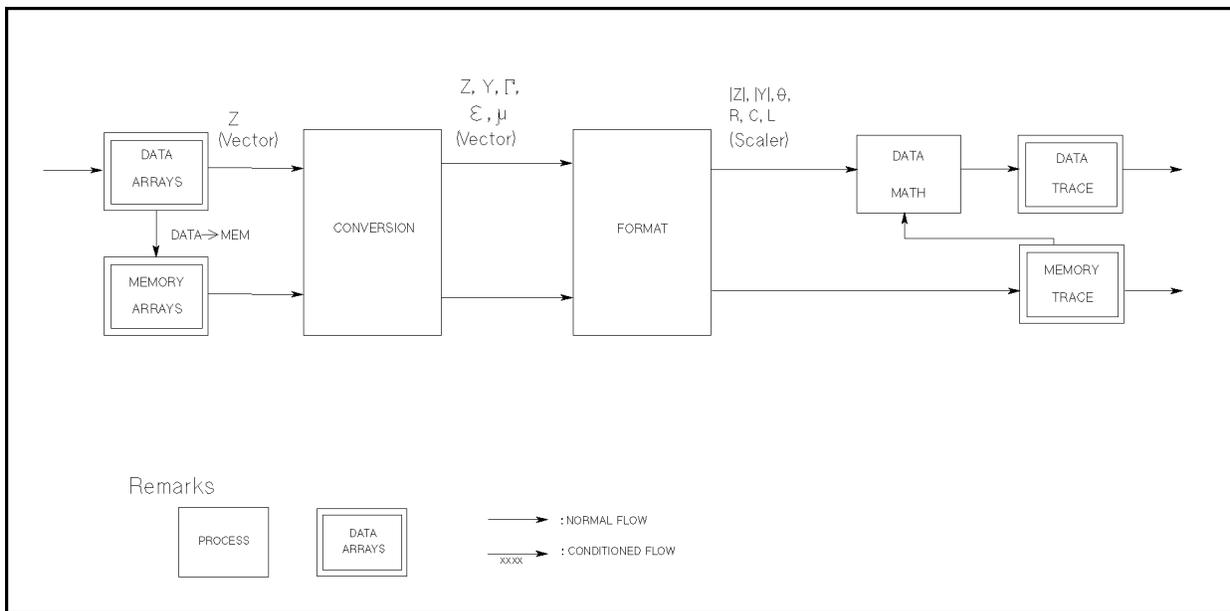
SAVDSTAC	<string>	[no query]
SAVDTIF	<string>	[no query]
SELM	<numeric>	
SING		[no query]
SPAN	<numeric>	
STAR	<numeric>	
STODDISK		[query returns {0 1}]
STODMEMO		[query returns {0 1}]
STOP	<numeric>	
STORMDISK		[no query]
STYPE	{LIN LOG LIST}	
SWED	{UP DOWN}	
SWEDTIME	<numeric>	
SWES	{FREQ OLEV DCV DCI}	
TMARG	<numeric>	
USKEY		[no query]

Measurement Parameter Settings Using GPIB Commands

This appendix shows the operation of the measurement parameter settings using the GPIB commands.

Each measurement parameter of the 4291B goes through the following flow:

Figure E-1 shows the data formatting. The Z (Vector) in the data array is converted to an admittance or reflection coefficient (Vector), and is formatted for each measurement parameter, Z, Y, R, C, L and so on (Scalar).



C640D001

Figure E-1. Data Formatting Inside 4291B

The measurement parameter setting depends on the CONVERSION and FORMAT, which are controlled by the following GPIB commands.

CONVERSION	CALCulate:MATH1[:EXPRession]:NAME {ADM DCO PER RCO} CALCulate:MATH1:STATe {OFF ON 0 1}
FORMAT	CALCulate:FORMat {MLINear PHASe UPHase REAL IMAGinary LFACTor LTANGent CP CS LP LS D Q RP RS COMPLex}

Table E-1. GPIB Commands Setting Measurement Parameters (1/2)

FORMAT	CONVERSION		
	CALC:MATH:STAT OFF	CALC:MATH:STAT ON CALC:MATH:NAME ADM	CALC:MATH:STAT ON CALC:MATH:NAME RCO
CALC:FORM MLIN	IMPEDANCE: MAG(Z)	ADMITTANCE: MAG(Y)	REFL.COEF: MAG(Γ)
CALC:FORM PHAS ¹	PHASE(θ _z)	PHASE(θ _y)	PHASE(θ _Γ)
CALC:FORM UPH ²	PHASE(θ _z)	PHASE(θ _y)	PHASE(θ _Γ)
CALC:FORM REAL	RESIST(R)	CONDUCT(G)	RESIST(Γ _x)
CALC:FORM IMAG	REACT(X)	SUSCEPT(B)	REACT(Γ _y)
CALC:FORM LFAC ³	—	—	—
CALC:FORM LTAN ³	—	—	—
CALC:FORM CP	CAPACITANCE: PRL(C _p)	CAPACITANCE: PRL(C _p)	CAPACITANCE: PRL(C _p)
CALC:FORM CS	SER(C _s)	SER(C _s)	SER(C _s)
CALC:FORM LP	INDUCTANCE: PRL(L _p)	INDUCTANCE: PRL(L _p)	INDUCTANCE: PRL(L _p)
CALC:FORM LS	SER(L _s)	SER(L _s)	SER(L _s)
CALC:FORM Q	Q FACTOR(Q)	Q FACTOR(Q)	Q FACTOR(Q)
CALC:FORM D	D FACTOR(D)	D FACTOR(D)	D FACTOR(D)
CALC:FORM RP	RESISTANCE: PRL(R _p)	RESISTANCE: PRL(R _p)	RESISTANCE: PRL(R _p)
CALC:FORM RS	SER(R _s)	SER(R _s)	SER(R _s)
CALC:FORM COMP ⁴	—	—	—

1 This command is used to set the **Format** EXP PHASE on OFF

2 This command is used to set the **Format** EXP PHASE ON off

3 This command is used to select the permittivity and permeability parameters. See Table E-2.

4 This command is used to set the **Format** POLAR CHART, SMITH CHART, ADMITTANCE CHART, COMPLEX PLANE.

Example To select ADMITTANCE: MAG(|Y|),

```
OUTPUT @Hp4291;"CALC:MATH:STAT ON"
OUTPUT @Hp4291;"CALC:MATH:NAME ADM"
OUTPUT @Hp4291;"CALC:FORM MLIN"
```

Table E-2. GPIB Commands Setting Measurement Parameters (2/2)

FORMAT	CONVERSION	
	CALC:MATH:STAT ON CALC:MATH:NAME DCO	CALC:MATH:STAT ON CALC:MATH:NAME PER
CALC:FORM MLIN	MAG($ e_r $)	MAG($ \mu_r $)
CALC:FORM PHAS ¹	—	—
CALC:FORM UPH ¹	—	—
CALC:FORM REAL	PERMITTVTY: REAL(e_r')	PRMEABLYTY: REAL(μ_r')
CALC:FORM IMAG ¹	—	—
CALC:FORM LFAC	LOSS FACTR (e_r'')	LOSS FACTR (μ_r'')
CALC:FORM LTAN	LOSS TNGNT ($\tan\delta$)	LOSS TNGNT ($\tan\delta$)
CALC:FORM CP ¹	—	—
CALC:FORM CS ¹	—	—
CALC:FORM LP ¹	—	—
CALC:FORM LS ¹	—	—
CALC:FORM Q ¹	—	—
CALC:FORM D ¹	—	—
CALC:FORM RP ¹	—	—
CALC:FORM RS ¹	—	—
CALC:FORM COMP ²	—	—

1 This command is used to select the impedance (also admittance and refraction factor) parameters. See Table E-1.

2 This command is used to set the **Format** COMPLEX PLANE .

Error Messages

This section lists the error messages that are displayed on the analyzer display or transmitted by the instrument over GPIB. Each error message is accompanied by an explanation, and suggestions are provided to help in solving the problem. Where applicable, references are provided to the related chapter of the appropriate manual. The messages are listed in numerical order.

In the explanation of many error commands, section numbers of the IEEE standard 488.2 are included. Refer to them for additional information about an error with these IEEE section numbers.

-430 **Query DEADLOCKED**

A condition causing a deadlocked query error occurred (see IEEE 488.2, 6.3.1.7). For example, both input buffer and output buffer are full and the analyzer cannot continue.

-420 **Query UNTERMINATED**

A condition causing an unterminated query error occurred (see IEEE 488.2, 6.3.2.2). For example, the analyzer was addressed to talk and an incomplete program message was received by the controller.

-410 **Query INTERRUPTED**

A condition causing an interrupted query error occurred (see IEEE 488.2, 6.3.2.3). For example, a query followed by DAB or GET before a response was completely sent.

-400 **Query errors**

This is the generic query error that the analyzer cannot detect more specific errors. This code indicates only that a query error as defined in IEEE 488.2, 11.5.1.1.7 and 6.3 has occurred.

-350 **Queue overflow**

A specific code entered into the queue in lieu of the code that caused the error. This code indicates that there is no room in the queue and an error occurred but was not recorded.

-330 **Self-test failed**

A self-test failed. Contact your nearest Agilent Technologies office or see the *Service Manual* for troubleshooting.

-311 **Memory error**

An error was detected in the analyzer's memory.

Error Messages**-310 System error**

Some error, termed “system error” by the analyzer, has occurred.

-286 Program runtime error

A program runtime error of the HP instrument BASIC has occurred. To get a more specific error information, use the ERRM\$ or ERRN command of the HP instrument BASIC.

-285 Program syntax error

A syntax error appears in a downloaded program. The syntax used when parsing the downloaded program is device-specific.

-284 Program currently running

Certain operations dealing with programs may be illegal while the program is running. For example, deleting a running program might not be possible.

-283 Illegal variable name

An attempt was made to reference a nonexistent variable in a program.

-282 Illegal program name

The name used to reference a program was invalid. For example, redefining an existing program, deleting a nonexistent program, or in general, referencing a nonexistent program.

-281 Cannot create program

An attempt to create a program was unsuccessful. A reason for the failure might include not enough memory.

-280 Program error

A downloaded program-related execution error occurred. This error message is used when the analyzer cannot detect the more specific errors described for errors –281 through –289.

-257 File name error

A legal program command or query could not be executed because the file name on the device media was in error. For example, an attempt was made to copy to a duplicate file name. The definition of what constitutes a file name error is device-specific.

-256 File name not found

A legal program command could not be executed because the file name on the device media was not found: for example, an attempt was made to read or copy a nonexistent file.

-250 Mass storage error

A mass storage error occurred. This error message is used when the analyzer cannot detect the more specific errors described for errors –251 through –259.

-241 Hardware missing

A legal program command or query could not be executed because of missing analyzer hardware. For example, an option was not installed.

-240 Hardware error

A legal program command or query could not be executed because of a hardware problem in the analyzer. Definition of what constitutes a hardware problem is completely device-specific. This error message is used when the analyzer cannot detect the more specific errors described for errors –241 through –249.

-231 Data questionable

Measurement accuracy is suspect.

-230 Data corrupt or stale

Possibly invalid data. New reading started but not completed since last access.

-225 Data out of memory

The analyzer has insufficient memory to perform the requested operation.

-224 Illegal parameter value

Used where exact value, from a list of possibilities, was expected.

-223 Too much data

A legal program data element of block, expression, or string type was received that contained more data than the analyzer could handle due to memory or related device-specific requirements.

-222 Data out of range

A legal program data element was parsed but could not be executed because the interpreted value was outside the legal range as defined by the analyzer (see IEEE 488.2, 11.5.1.1.5).

-221 Settings conflict

A legal program data element was parsed but could not be executed due to the current device state (see IEEE 488.2, 6.4.5.3 and 11.5.1.1.5).

-220 Parameter error

Indicates that a program data element related error occurred. This error message is used when the analyzer cannot detect the more specific errors described for errors –221 through –229.

-213 Init ignored

A request for a measurement initiation was ignored as another measurement was already in progress.

-211 Trigger ignored

A GET, *TRG, or triggering signal was received and recognized by the analyzer but was ignored because of analyzer timing considerations. For example, the analyzer was not ready to respond.

-210 Trigger error

A trigger related error occurred. This error message is used when the analyzer cannot detect the more specific errors described for errors –211 through –219.

Error Messages**-200 Execution error**

This is the generic syntax error that the analyzer cannot detect more specific errors. This code indicates only that an execution error as defined in IEEE 488.2, 11.5.1.1.5 has occurred.

-168 Block data not allowed

A legal block data element was encountered but was not allowed by the analyzer at this point in parsing.

-161 Invalid block data

A block data element was expected, but was invalid for some reason (see IEEE 488.2, 7.7.6.2). For example, an END message was received before the length was satisfied.

-160 Block data error

This error, as well as errors –161 and –168, are generated when analyzing the syntax of a block data element. This particular error message is used if the analyzer cannot detect a more specific error.

-158 String data not allowed

A string data element was encountered but was not allowed by the analyzer at this point in parsing.

-151 Invalid string data

A string data element was expected, but was invalid for some reason (see IEEE 488.2, 7.7.5.2). For example, an END message was received before the terminal quote character.

-150 String data error

This error, as well as errors –151 and –158, are generated when analyzing the syntax of a string data element. This particular error message is used if the analyzer cannot detect a more specific error.

-148 Character data not allowed

A legal character data element was encountered where prohibited by the analyzer.

-144 Character data too long

The character data element contains more than twelve characters (see IEEE 488.2, 7.7.1.4).

-141 Invalid character data

Either the character data element contains an invalid character or the particular element received is not valid for the header.

-140 Character data error

This error, as well as errors –141 through –148, are generated when analyzing the syntax of a character data element. This particular error message is used if the analyzer cannot detect a more specific error.

-138 Suffix not allowed

A suffix was encountered after a numeric element that does not allow suffixes.

-134 Suffix too long

The suffix contained more than 12 characters (see IEEE 488.2, 7.7.3.4).

-131 Invalid suffix

The suffix does not follow the syntax described in IEEE 488.2, 7.7.3.2, or the suffix is inappropriate for the analyzer.

-130 Suffix error

This error, as well as errors –131 through –139, are generated when parsing a suffix. This particular error message is used if the analyzer cannot detect a more specific error.

-128 Numeric data not allowed

A legal numeric data element was received, but the analyzer does not accept it in this position for a header.

-124 Too many digits

The mantissa of a decimal numeric data element contains more than 255 digits excluding leading zeros (see IEEE 488.2, 7.7.2.4.1).

-123 Exponent too large

The magnitude of the exponent was larger than 32000 (see IEEE 488.2, 7.7.2.4.1).

-121 Invalid character in number

An invalid character for the data type being parsed was encountered. For example, an alpha character in a decimal numeric or a “9” in octal data.

-120 Numeric data error

This error, as well as errors –121 through –129, are generated when parsing a data element that appears to be numeric, including the nondecimal numeric types. This particular error message is used if the analyzer cannot detect a more specific error.

-114 Header Suffix out of range

The value of a numeric suffix attached to a program mnemonic makes the header invalid.

-113 Undefined header

The header is syntactically correct, but it is undefined for the analyzer. For example, *XYZ is not defined for the analyzer.

-112 Program mnemonic too long

The header contains more than twelve characters (see IEEE 488.2, 7.6.1.4.1).

Error Messages**-111 Header separator error**

A character that is not a legal header separator was encountered while parsing the header. For example, no white space followed the header, thus *SRE4 is an error.

-110 Command header error

An error was detected in the header. This error message is used when the analyzer cannot detect the more specific errors described for errors –111 through –119.

-109 Missing parameter

Fewer parameters were received than required for the header. For example, the *SRE command requires one parameter, so receiving only *SRE is not allowed.

-108 Parameter not allowed

More parameters were received than expected for the header. For example, the *SRE command only accepts one parameter, so receiving *SRE 4,16 is not allowed.

-105 GET not allowed

A Group Execute Trigger (GET) was received within a program message (see IEEE 488.2, 7.7).

-104 Data type error

The parser recognized an unallowed data element. For example, numeric or string data was expected but block data was encountered.

-103 Invalid separator

The parser was expecting a separator and encountered an illegal character. For example, the semicolon was omitted after a program message unit, *RST:INIT.

-102 Syntax error

An unrecognized command or data type was encountered. For example, a string was received when the analyzer was not expecting to receive a string.

-101 Invalid character

A syntax element contains a character that is invalid for that type. For example, a header containing an ampersand (SENS&).

-100 Command error

This is a generic syntax error that the analyzer cannot detect more specific errors. This code indicates only that a command error, as defined in IEEE 488.2, 11.5.1.1.4, has occurred.

+0 No error

The error queue is empty. Every error in the queue has been read (SYSTEM:ERRor? query) or the queue was cleared by power-on or the *CLS command.

6 ADDITIONAL STANDARDS NEEDED

Error-correction coefficients cannot be computed until all the necessary standards have been measured. Execute all **OPEN**, **SHORT**, **LOAD** calibration (**SENSe:CORRection1:COLLect[:ACQuire] {STAN1|STAN2|STAN3}**) before press **DONE: CAL** (**SENSe:CORRection1:COLLect:SAVE**).

7 CALIBRATION REQUIRED

No valid calibration coefficients were found when you attempted to perform fixture compensation. See *Users Guide* for information on how to perform calibration.

9 NO CALIBRATION CURRENTLY IN PROGRESS

The **RESUME CAL SEQUENCE** softkey (No GPIB command) is not valid unless a calibration is in progress. Start a new calibration. See “**Cal** key” in the *Operation Manual*.

10 CALIBRATION ABORTED

The calibration in progress was terminated due to a change of the stimulus parameter or calibration measurement points. For example,

- Changing **CAL POINT [FIXED]** between **CAL POINT [USER]** (**SENSe:CORRection1:COLLect:FPOints {FIXed|USER}**).

11 COMPENSATION REQUIRED

No valid fixture compensation coefficients were found when you attempted to turn fixture compensation ON (**OPEN ON off** —**SENSe:CORRection2:OPEN ON**, **SHORT ON off** —**SENSe:CORRection2:SHORT ON**, **LOAD ON off** —**SENSe:CORRection2:LOAD ON**). See *Users Guide* for information on how to perform compensation.

12 NO COMPENSATION CURRENTLY IN PROGRESS

The **RESUME COMP SEQ** softkey (No GPIB command) is not valid unless a fixture compensation is in progress. Start a new calibration. See “**Cal** key” in the *Function Reference*.

13 COMPENSATION ABORTED

The compensation in progress was terminated due to a change of the stimulus parameter or calibration measurement points. For example,

- Changing **COMP POINT [FIXED]** between **COMP POINT [USER]** (**SENSe:CORRection2:COLLect:FPOints {FIXed|USER}**) before pressing **DONE: COMPEN** (**SENSe:CORRection2:COLLect:SAVE**).

14 NOT ALLOWED IN DC BIAS SWEEP

The calibration **CAL POINTS [USER]** (**SENSe:CORRection1:COLLect:FPOints USER**) or compensation in **COMP POINT [USER]** (**SENSe:CORRection2:COLLect:FPOints USER**) cannot be executed in the DC-V/DC-I sweep.

Error Messages**15 COMPENSATION STD LIST UNDEFINED**

(*GPIB only*) You cannot execute `SENSe:CORRection2:CKIT[1]:STANdard{1-3}[:SElect]` LIST when the fixture compensation standard array is not defined.

16 CAN'T CHANGE-HIGH TEMP TEST HEAD CONNECTED

You cannot execute `CAL POINTS [FIXED]` (`SENSe:CORRection1:COLLect:FP0ints FIXed`) or `COMP POINT [FIXED]` (`SENSe:CORRection2:COLLect:FP0ints FIXed`) when the high temperature test head is connected.

22 PRINTER: not on, not connected, out of paper

The printer does not respond to control. Check the supply to the printer, online status, sheets, and so on.

30 NO VALID MEMORY TRACE

If memory traces are to be displayed or otherwise used, a data trace must first be stored to memory.

31 CAN'T CALCULATE EQUIVALENT PARAMETERS

Data is not match to the equivalent circuit and cannot calculate the parameters.

32 MUST BE MORE THAN 2 POINTS FOR ANALYSIS

`CALCULATE EQV PARAMS` (`CALCulate:EVALuate:EPARameters`) is pressed when the NOP (number of points) is 2. Set the number of measurement points to the number lager than 2.

33 MEM-TRACE MEMORY FULL

Another memory trace cannot be saved because the total NOP of memory traces exceeds (801×3).

40 PHASE LOCK LOOP UNLOCKED

Sever error. Contact your nearest Agilent Technologies office.

46 TOO MUCH DATA

(*GPIB only*) Either there is too much binary data to send to the analyzer when the data transfer format is binary, or the amount of data is greater than the number of points.

47 NOT ENOUGH DATA

(*GPIB only*) The amount of data sent to the analyzer is less than that expected when the data transfer format is binary.

48 OPTION NOT INSTALLED

(*GPIB only*) This error occurs when an GPIB command which is optional command is sent and the analyzer is not installed the option. Please confirm options installed to the analyzer using `*OPT?` command (see "`*OPT?`" in Chapter 11.)

56 TOO MANY SEGMENTS

The maximum number of segments for the limit line table is 18.

66 CURRENT EDITING SEGMENT SCRATCHED

The current editing the table of list sweep or the limit line is scratched. It is occur when the operation other than editing the table is executed before terminate editing the table (SENSE:LIST:SAVE, or CALCulate:LIMit:SAVE)

67 COMMAND IGNORED - SEGMENT NOT DONE YET

(*GPIB only*) The GPIB command the analyzer received is ignored, because the segment is editing . Send CALCulate:LIMit:SEGMENT:SAVE (limit segment done) or SENSE:LIST:SEGMENT:SAVE (segment done) to terminate editing segment.

68 SEGMENT START/STOP OVERLAPPED

Segments are not allowed to be overlapped. Reenter appropriate value for start or stop value of segments to avoid that segment is not overlapped.

69 TOO MANY SEGMENTS OR POINTS

Frequency list mode is limited to 15 segments or 801 points.

70 TOO SMALL POINTS OR TOO LARGE STOP

STOP+SPAN/(NOP-1) is out of sweep range. Increase NOP or change STOP value to lower frequency to avoid this error.

74 CAN'T CHANGE- ANOTHER CONTROLLER ON BUS

The analyzer cannot assume the mode of system controller until the system controller is removed from the bus or relinquishes the bus.

75 NO TEST HEAD CONNECTED

Check the test head connection.

76 UNKNOWN TEST HEAD CONNECTED

The test head get wrong. Contact your nearest Agilent Technologies office.

77 INVALID MATERIAL SIZE

(For the permeability measurement) The material size definition is wrong. The outer diameter must be larger than the inner.

78 MATERIAL SIZE UNDEFINED

(For the permittivity and permeability measurement) The **DONE (MODIFIED)** key (under **Meas MATERIAL SIZE**) is pressed or it is attempt to select the measurement parameter key when the material sizes are empty. Define the material size before press these keys.

Error Messages**79 NOT AVAILABLE FOR THIS FORMAT**

(For the permittivity and permeability measurement) You cannot execute **POLAR CHART**, **SMITH CHART**, and **ADMITTANCE CHART** (`DISPlay[:WINDow]:TRACe1:GRATicule:FORMat {POLar|SMITH|ADMittance}`).

80 NOT AVAILABLE FOR THIS FIXTURE

(*GPIB only*) You cannot execute `CALCulate:MATH1[:EXPRession]:NAME {DCO|PER}` when the `SYSTem:FIXTure {NONE|HP16191|HP16192|HP16193|HP16194}` is selected.

87 NO DATA TRACE

The **MARKER ON [DATA]** (`CALCulate:EVALuate:ON1 "TR1"`) is selected when the data trace is not displayed.

88 NO MEMORY TRACE

The **MARKER ON [MEMORY]** (`CALCulate:EVALuate:ON1 "TR{2-17}"`) is selected when the memory trace is not displayed.

89 NO MARKER DELTA - SPAN NOT SET

The **MKRA—SPAN** softkey (`SENSe:FREQuency:SPAN DMARker`, or `SOURce{1|2}:{VOLTage|CURRent}:SPAN DMARker'`) requires that the Δ marker mode be turned ON.

90 NO MARKER DELTA - RANGE NOT SET

The **MKRA—SEARCH RNG** softkey (`CALCulate:EVALuate:BAND:SPAN DMARker`) requires that Δ marker is turned ON.

92 NO ACTIVE MARKER

(*GPIB only*) The `marker→` command cannot be execute when no marker is displayed on the screen. Turn on the marker before executing the `marker→` commands.

93 CAN'T CHANGE WHILE DUAL CHAN OFF

The cross channel (`CALCulate:EVALuate:EFFect:ON 1`) cannot be turned on when dual channel is off. Turn on the dual channel before the cross channel is turned on.

94 NO FIXED DELTA MARKER

The Δ marker cannot move (`ΔMKR STIMULUS`, `FIXED ΔMKR VALUE` or `FIXED Δ AUX VALUE — CALCulate:EVALuate:REFerence:{X|Y1|Y2} <numeric>` cause the error) because,

- The Δ marker is not turned on. → Turn the Δ marker ON (`DISPlay[:WINDow]:TRACe:MARKer:RELative ON`).
- The only fixed Δ marker can move by `FIXED ΔMKR VALUE` or `FIXED Δ AUX VALUE`. → Press `FIXED ΔMKR` (`DISPlay[:WINDow]:TRACe:MARKer:RELative:REFerence FIXed`).

95 FREQUENCY SWEEP ONLY

Cannot select MKR X AXIS $[1/(2\pi F)]$ in OSC level sweep, or DC-V/DC-I sweep.

104 SAVE ERROR

A serious error, for example physically damaged disk surface, is detected on saving a file.

105 RECALL ERROR: INSTR STATE PRESET

A serious error, for example corrupted data, is detected on recalling a file, and this forced the analyzer to be PRESET.

107 NO STATE/DATA/IBASIC FILES ON DISK

(Front-panel key only) The RE-SAVE FILE, COPY FILE, PURGE FILE, PROGRAM MENU or **Recall** key pressed, but there are no files with extensions (“_D” or “_S” for LIF format, or “.STA” or “.DAT” for DOS format) on the floppy disk.

108 CAN'T SAVE GRAPHICS WHEN COPY IN PROGRESS

If you attempt to save graphics when a print is in progress, this error message is displayed. Wait until print is complete, then save graphics again.

110 NO STATE/DATA/IBASIC FILES ON MEMORY

(Front-panel key only) The RE-SAVE FILE, COPY FILE, PURGE FILE, PROGRAM MENU or **Recall** key pressed, but there are no files with extensions (“_D” or “_S” for LIF format, or “.STA” or “.DAT” for DOS format) on the memory disk.

113 NO DATA TRACE DISPLAYED

The SCALE FOR [DATA] (DISPlay[:WINDow]:TRACe1:Y[:SCALe]) is selected when the data trace is not displayed.

114 NO MEMORY TRACE DISPLAYED

The SCALE FOR [MEMORY] (DISPlay[:WINDow]:TRACe{2-17}:Y[:SCALe]) is selected when the memory trace is not displayed.

115 LIF-DOS COPY NOT ALLOWED

If you try to copy a file between the memory disk and the floppy disk when the format of the memory disk is different from the format of the floppy disk, this message is displayed.

118 LIST TABLE EMPTY OR INSUFFICIENT TABLE

The frequency list is empty. To implement the list frequency mode, add segments to the list table.

119 FREQUENCY SWEEP ONLY

Equivalent circuit function is executed in OSC level sweep, DC-I sweep, DC-V sweep. The equivalent circuit function is available in frequency sweep only.

Error Messages**120 CAN'T CHANGE IN LIST SWEEP**

When list sweep is selected, the following parameters are not allowed to be changed:

- Stimulus center, span, start, stop
- Number of Point
- OSC level

Modify the list table to change these parameters in the list sweep.

121 NOT ALLOWED IN FREQUENCY SWEEP

SWEEP DIR [] (SOURCE{1|2}:SWEEP:DIRection DOWN) is pressed in frequency sweep. Sweep direction down is only available for OSC level sweep, DC-V, or DC-I sweep.

127 DC BIAS OVERLOAD

Hardware failure. Do not input external DC BIAS. If this message keeps on being displayed, contact your nearest HP service office.

132 BACKUP DATA LOST

Data checksum error on the battery backup memory has occurred. The battery is recharged for approximately 10 minutes after power was turned ON.

140 ON POINT NOT ALLOWED FOR THE CURRENT TRIG

The trigger event mode cannot be changed to the ON POINT mode because the current trigger source setting does not allow the ON POINT mode. The trigger event ON POINT mode is available for only MANUAL, EXTERNAL, and BUS trigger sources.

141 INSUFFICIENT MEMORY

If a lot of tasks is executed at same time, memory might be insufficient for a while. (For example, running HP instrument BASIC program, printing a screen, and sending or receiving data array by GPIB are required at same time.) Please wait until finishing some tasks then execute the next task.

148 INVALID DATE

The date entered to set the real time clock is invalid. Reenter correct date.

158 UNIT STRING TOO LONG

(GPIB only) DISPLAY[:WINDow]:TRACe{18-21}:X:UNIT <string> or DISPLAY[:WINDow]:TRACe{18-21}:Y:UNIT <string> commands can send <string> up to 4 characters.

159 NO VALID USER TRACE

The marker cannot be used in user trace because the selected user trace is OFF.

160 INVALID X-AXIS VALUE FOR LOG

User trace cannot be displayed in log scale, because,

- The right value and left value of the X-axis is same.
- The X-axis range is defined from - value to + value.

Change the X-axis right/left value.

189 NOT ALLOWED IN SVC MODE

Dual channel cannot be displayed in the service mode.

190 CORR. CONST. DATA LOST; DEFAULT DATA IS USED

This message is displayed when the correction constants EEPROM data is lost and turned on in the service mode. See the *Service Manual* for troubleshooting.

198 POWER ON TEST FAILED

An internal test fails in the power on sequence (the power on self-test fails). Contact your nearest Agilent Technologies office or see the *Service Manual* for troubleshooting.

199 EEPROM WRITE ERROR

Data cannot be stored properly into the EEPROM on the A1 CPU, when performing the display background adjustment or updating correction constants in the EEPROM using the adjustment program. See the *Service Manual* for troubleshooting.

203 EEPROM CHECK SUM ERROR

An “internal test 1: A1 CPU” fails. The data (Correction Constants and so on) stored in the A1 CPU’s EEPROM are invalid. See the *Service Manual* for troubleshooting.

204 DSP CHIP TEST FAILED

An “internal test 1: A1 CPU” fails. The A1 CPU’s DSP (Digital Signal Processor) does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

205 F-BUS TIMER CHIP TEST FAILED

An “internal test 1: A1 CPU” fails. The A1 CPU’s F-BUS (Frequency Bus) timer does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

206 RTC CHIP TEST FAILED

An “internal test 1: A1 CPU” fails. The A1 CPU’s RTC (Real Time Clock) does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

207 KEY CHIP TEST FAILED

An “internal test 1: A1 CPU” fails. The A1 CPU’s front keyboard control chip does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

208 FDC CHIP TEST FAILED

An “internal test 1: A1 CPU” fails. The A1 CPU’s FDC (Flexible Disk drive control) chip does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

209 HP-IB CHIP TEST FAILED

An “internal test 1: A1 CPU” fails. The A1 CPU’s GPIB chip does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

Error Messages**210 DIN CHIP TEST FAILED**

An “internal test 1: A1 CPU” fails. The A1 CPU’s DIN control chip does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

211 CPU INTERNAL SRAM R/W ERROR

An “internal test 2: A1 VOLATILE MEMORY” fails. The A1 CPU’s internal SRAM does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

212 CPU BACKUP SRAM R/W ERROR

An “internal test 2: A1 VOLATILE MEMORY” fails. The A1 CPU’s BACKUP SRAM does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

213 DSP SRAM R/W ERROR

An “internal test 2: A1 VOLATILE MEMORY” fails. The DSP’s SRAM on the A1 CPU does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

214 DUAL PORT SRAM R/W ERROR

An “internal test 2: A1 VOLATILE MEMORY” fails. The DSP’s dual port SRAM on the A1 CPU does not work properly. Replace the A1 CPU with a new one. See the *Service Manual* for troubleshooting.

215 POST REGULATOR OUTPUT VOLTAGE OUT OF SPEC

An “internal test 4: A2 POST REGULATOR” fails. A power supply voltage of the A2 post-regulator is out of its limits. See the *Service Manual* for troubleshooting.

216 GND LEVEL OUT OF SPEC

An “internal test 4: A2 POST REGULATOR” fails. The voltage of the GND (Ground) at the DC bus node 26 is out of its limits. See the *Service Manual* for troubleshooting.

217 FAN POWER OUT OF SPEC

An “internal test 4: A2 POST REGULATOR” fails. The voltage of the fan power supply at the DC bus node 11 is out of its limits. See the *Service Manual* for troubleshooting.

218 FAILURE FOUND FROM A/D MUX TO A/D CONVERTER

An “internal test 5: A6 A/D CONVERTER” fails. A trouble is found on the signal path from the A/D multiplexer to A/D converter on the A6 receiver IF. See the *Service Manual* for troubleshooting.

219 REF OSC TEST FAILED

An “internal test 6: A5 REFERENCE OSC” fails. The reference oscillator on the A5 synthesizer does not work properly. See the *Service Manual* for troubleshooting.

220 FRACTIONAL N OSC TEST FAILED

An “internal test 7: A5 FRACTIONAL N OSC” fails. The fractional N oscillator on the A5 synthesizer does not work properly. See the *Service Manual* for troubleshooting.

221 STEP OSC TEST FAILED

An “internal test 8: A5 STEP OSC” fails. The step oscillator on the A5 synthesizer does not work properly. See the *Service Manual* for troubleshooting.

222 1st LO OSC TEST FAILED

An “internal test 9: A4A1 1ST LO OSC” fails. The 1st LO OSC (first local oscillator) on the A4A1 1st LO does not work properly. See the *Service Manual* for troubleshooting.

223 2nd LO OSC TEST FAILED

An “internal test 10: A3A2 2ND LO” fails. The 2nd LO OSC (second local oscillator) on the A3A2 2nd LO does not work properly. See the *Service Manual* for troubleshooting.

224 A3 DIVIDER OUTPUT FREQUENCY OUT OF SPEC

An “internal test 11: A3A1 DIVIDER” fails. The output frequency of the divider circuit on the A3A1 ALC is out of its limits. See the *Service Manual* for troubleshooting.

225 3rd LO OSC TEST FAILED

An “internal test 12: A6 3RD LO OSC” fails. The 3rd LO OSC (third local oscillator) on the A6 receiver IF does not work properly. See the *Service Manual* for troubleshooting.

226 SOURCE OSC TEST FAILED

An “internal test 13: A3A1 SOURCE OSC” fails. The source oscillator on the A3A1 ALC does not work properly. See the *Service Manual* for troubleshooting.

227 SAMPLE FREQUENCY OUT OF SPEC

An “internal test 14: A6 SEQUENCER” fails. The sampling frequency of the sample/hold circuit on the A6 receiver IF is out of its limits.

228 SOURCE LEVEL TEST FAILED

An “internal test 15: SOURCE LEVEL” fails. See the *Service Manual* for troubleshooting.

229 DC BIAS TEST FAILED

An “internal test 16: DC BIAS” fails. See the *Service Manual* for troubleshooting.

230 FLOPPY DISK DRIVE FAILURE FOUND

An “external test 18: DSK DR FAULT ISOL'N” fails. The A53 built-in FDD (floppy disk drive) does not work properly. Replace the A53 FDD with a new one. See the *Service Manual* for troubleshooting.

231 POWER SWEEP LINEARITY TEST FAILED

An “external test 19: POWER SWEEP LINEARITY” fails. See the *Service Manual* for troubleshooting.

Error Messages**232 SOURCE LEVEL TEST FAILED**

An “external test 20: SOURCE LEVEL” fails. See the *Service Manual* for troubleshooting.

233 OUTPUT ATTENUATOR TEST FAILED

An “external test 21: OUTPUT ATTENUATOR” fails. See the *Service Manual* for troubleshooting.

234 TRD LOSS TEST FAILED

An “external test 22: TRD LOSS” fails. See the *Service Manual* for troubleshooting.

235 TRD ISOL’N I TO V TEST FAILED

An “external test 28: TRD ISOL’N I TO V” fails. See the *Service Manual* for troubleshooting.

236 TRD ISOL’N V TO I TEST FAILED

An “external test 29: TRD ISOL’N V TO I” fails. See the *Service Manual* for troubleshooting.

237 HIGH Z HEAD TEST FAILED

An “external test 30: HIGH Z HEAD” fails. See the *Service Manual* for troubleshooting.

238 LO Z HEAD TEST FAILED

An “external test 31: LOW Z HEAD” fails. See the *Service Manual* for troubleshooting.

239 FRONT ISOL’N TEST FAILED

An “external test 25: FRONT ISOL’N” fails. See the *Service Manual* for troubleshooting.

240 CABLE ISOL’N TEST FAILED

An “external test 27: ” fails. See the *Service Manual* for troubleshooting.

241 RECEIVER GAIN TEST FAILED

An “external test 22: RECEIVER GAIN” fails. See the *Service Manual* for troubleshooting.

242 RECEIVER GAIN OUT OF SPEC

An “external test 25: FRONT ISOL’N” fails. A6 receiver IF gain is incorrect. See the *Service Manual* for troubleshooting.

243 A6 GAIN TEST FAILED

An “external test 23: A6 GAIN” fails. See the *Service Manual* for troubleshooting.

244 A6 VI NORMALIZER TEST FAILED

An “external test 24: A6 VI NORMALIZER” fails. See the *Service Manual* for troubleshooting.

245 MAX VCXO LEVEL OUT OF SPEC

Maximum VCXO level is incorrect, in performing an “adjustment test 36: 3RD VCXO LEVEL ADJ” or an “adjustment test 39: SOURCE VCXO LEVEL ADJ”. See the *Service Manual* for troubleshooting.

246 **VCXO TUNING VOLTAGE OUT OF LIMIT**

VCXO tuning voltage is incorrect, in performing an “adjustment test 36: 3RD VCXO LEVEL ADJ” or an “adjustment test 39: SOURCE VCXO LEVEL ADJ”. See the *Service Manual* for troubleshooting.

248 **HIGH TMP HIGH Z HEAD TEST FAILED**

An “external test 32: HIGH TMP HIGH Z HEAD TEST FAILED” fails. See the *Service Manual* for troubleshooting.

249 **HIGH TMP LOW Z HEAD TEST FAILED**

An “external test 33: HIGH TMP LOW Z HEAD TEST FAILED” fails. See the *Service Manual* for troubleshooting.

Index

8

8 bit I/O, 11-110, C-36

A

ABOR, 5-2

ABORT, 1-5

accessing calibration coefficient arrays, 3-4

accessing compensation coefficient arrays,
3-4

active channel, 2-3

active controller, 1-5

admittance, 2-9

admittance chart, 2-10

ALL BASIC, 10-2

ALL INSTRUMENT, 10-2

arrow key

⏴, 10-14

⌂, 10-14

⏵, 10-14

⏴, 10-14

⏶, 10-14

ASCII data transfer, 3-7

automating the measurement procedure
impedance, 2-3

AUTOREC, 9-3

auto scale, 2-13

AUTOST, 9-3

AUTOST, 9-1

autostart, 9-3

averaging, 5-5

averaging factor, 11-76

B

back space, 10-4

⏪, 10-14

BACKUP MEMORY DISK, 9-2

bandwidth center value, 11-12

bandwidth parameter, 11-12, C-35

bandwidth peak value, 11-12

bandwidth value, 11-12

BASIC STATUS, 10-2

beeper, 11-109

binary data transfer, 3-8

BIN sorting, 6-3

<block>, 1-10

bus trigger, 5-2

C

calibration, 2-6

calibration coefficient, 3-3

calibration coefficient array, 11-28, C-33

calibration state, 2-6

⌨, 10-14

CAT, 10-7

character

deleting, 10-4

inserting, 10-4

character entry key, 10-14

clearing register, 4-9

clear marker, 2-13

*CLS, 4-9

Cole-Cole plot, 2-18

command abbreviation, 1-8

COMMAND ENTRY, 10-3

common command, 1-7

compensation coefficient array, 11-28, C-33

compensation coefficient array, 3-4

complex plane, 2-10

condition register, 4-3

controller, 1-5

convention, 11-1

conversion, 11-23, E-1

conversion mode, 2-9

⌨, 10-14

cursor

moving, 10-4

D

data array, 11-28, C-33

data array, 3-2

data arrays, 3-1

data header, 3-9

data math, 11-25, C-35

data processing flow, 3-1

data trace, 11-117, C-33

data trace, 3-2

data transfer format, 11-52, C-34

data transfer methods, 3-7

DATE, 10-18

DATE\$, 10-18

dc bias, 11-101

dc bias , 2-11
 dc current bias, 11-101
 dc current sweep , 2-5
 dc voltage bias, 11-101
 dc voltage sweep , 2-5
 debugging program , 8-11
Delete, 10-15
 device selector , 1-3, **1-6**
 disk to variable , 8-2
DISPLAY ALLOCATION, 10-2
 display allocation , 1-13
 display format, 11-17
 display format , 2-10
 DOS format , 10-6
 DOS Format , 9-4
 dual channel, 11-62, 11-128, C-5, C-9

E

EDIT mode, 10-3
 electrical length, 11-111
 electrical length , 2-7
 enable register , 4-2
End, 10-15
Enter, 10-14
 equivalent circuit, 11-29
 error, 11-111, C-34
 "ERROR -257", 10-7
 error corrected data , 3-2
 error message, Messages-1
 error queue , 8-11
 event register , 4-2
 external controller , 1-14
 external reference, 11-32

F

Facilitating Program Execution , 9-1
 file name
 listing, 10-7
 "File name error", 10-7
 file transfer, 8-15
 File transfer, C-36
 firmware revision, 11-32
 fixture compensation , 2-7, 3-4
 fixture compensation load standard array,
 11-28, C-33
 fixture compensation open standard array,
 11-28, C-33
 fixture compensation short standard array,
 11-28, C-33
 floppy disk drive, 9-4
 floppy disk drive , 8-1
 format option, 9-4
 frequency sweep , 2-4

G

general status register model , 4-1
 GET, 10-8
 getting data from analyzer , 3-10
 getting measured data to controller , 2-13
 GPIB , v, 1-5
 GPIB address, 10-2
 setting , 1-13, 1-14
 GPIB address , **1-6**
 GPIB cable , 1-14
 GPIB command , 2-1
 GPIB control , 2-1
 GPIB error , 8-11
 graphics, 10-12

H

HALF INSTR HALF BASIC, 10-2
 hard copy, 10-6, 10-13
 HCOP, 8-5
Home, 10-15
 HP BASIC , v
 HP CITIfile , 8-3
 HP instrument BASIC, 10-1
 HP Instrument BASIC, 11-69, C-34
 HP Instrument BASIC , 1-6
 HP Instrument BASIC graphic function,
 11-38

I

IBASIC, 10-1
 idle state , 5-2
 IEEE 32-bit floating point format , 3-8
 IEEE 488 bus , 1-5
 IEEE 64-bit floating point format , 3-8
 impedance , 2-9
 INIT:CONT , 5-2
 INITIALIZE, 9-4
 initializing module , 1-3
insert, 10-15
 Instrument BASIC
 controlling , 8-6
 preparing , 1-13
 Instrument BASIC , 1-13
 Instrument BASIC editor , 1-15
 Instrument BASIC Execution Status
 Controlling , 8-8
 Determining , 8-8
 instrument control command, 1-7
 Instrument Event Status Enable register,
 11-105
 instrument event status register , 4-5
 Instrument Status Register, 11-105
 ":INTERNAL", 9-4, 10-6
 internal trigger , 5-4

Interrupt , 4-9
 I/O interfaces, 10-11
 I/O port. *See* 8 bit I/O
 accessing from external controller , 6-2
 pin assignment , 6-1
 reading data , 6-2
 writing data , 6-2
 I/O port , 10-18, 10-20

K

keyboard, 10-14
 key code, 11-113

L

label function, 10-8
 level monitor, 11-8, C-24, C-33
 level monitor , 3-5
 level monitor array, 11-28, C-33
 LIF format , 10-6
 LIF Format , 9-4
 limit test, 11-29, C-34
 line
 clearing, 10-5
 deleting, 10-5
 inserting, 10-5
 jumping, 10-5
 recalling, 10-5
 renumber, 10-5
 scrolling, 10-4
 linear format , 2-10
 linear sweep
 frequency , 2-4, 2-5
 list, 10-6
 on the screen, 10-6
 to printer, 10-6
 log format , 2-10
 LOGGING , 8-13
 log sweep
 dc bias , 2-5
 frequency , 2-4
 OSC level , 2-5

M

manual changes , A-1
 manual overview , 1-1
 manual trigger , 5-2, 5-5
 marker, 11-4, C-34
 marker , 2-13
 marker search , 2-13
 marker statistics, 11-7
 MAV , 4-4
 maximum , 2-13
 measurement parameter , 2-9
 measurement state , 5-3
 ":MEMORY", 9-4, 10-6

memory array, 11-30, C-33
 memory disk, 9-4
 memory disk , 8-1
 MEMORY PARTITION , 9-2
 memory trace, 11-117, C-33
 memory trace , 3-6
 minimum , 2-13
 MMEM: COPY , 9-4
 module , 1-3
 monitor array , 3-5
 MS-DOS format , 3-9
 MSS , 4-2, 4-4

N

number of groups , 5-4
 <numeric>, 1-9
 (Num Lock), 10-14

O

offset value, 11-27
 On Key Label function, 10-8
 Operation Status Register, 11-106
 operation status register , 4-7
 OSC level
 frequency sweep , 2-4
 OSC level sweep, 11-98
 OSC level sweep , 2-4
 OUTPUT , 2-1

P

(Page Down), 10-14
 (Page Up), 10-14
 pass control , 10-9
 PASS CONTROL , 1-5
 peak , 2-13
 permeability , 2-20
 point averaging , 2-12, 5-6
 point averaging factor, 11-75
 point data
 retrieving , 5-6
 point delay , 2-12, 5-6
 point measure complete bit , 5-5
 polar chart , 2-10
 port extension, 11-80
 port extension , 2-7
 power-on test, 11-32
 preparation for operation , 1-13
 preset, 11-114, C-35
 printer
 list to, 10-6
 PRINTER IS, 10-6
 printing , 8-5
 processing time measurement , 8-12
 program
 getting, 10-8

- listing, 10-6
- saving, 10-6
- Program Module*
 - adding complex arrays, B-1
 - auto scale for user trace , 7-3
 - checking calibration mode , 2-7
 - checking calibration state , 2-6
 - checking GPIB error , 8-11
 - controlling Instrument BASIC execution status , 8-8
 - detecting sweep end using SRQ and interrupt , 4-9
 - determining Instrument BASIC execution state , 8-8
 - displaying Cole-Cole plot , 2-18
 - dividing complex arrays, B-2
 - downloading , 8-10
 - entering data-trace data from disk into variable , 8-2
 - generating SRQ on PAUSED , 8-8
 - getting data from analyzer using binary transfer , 3-10
 - getting data trace array , 3-3
 - getting level monitor data , 3-5
 - getting marker measurement data , 2-13
 - getting memory array data , 3-6
 - getting raw data array , 3-2
 - getting stimulus array , 3-5
 - initialize for the external controller , 1-3
 - initialize module for Instrument BASIC , 1-3
 - making a single sweep measurement , 2-12
 - measuring processing time , 8-12
 - modifying permittivity compensation kit , 2-16
 - multiplying complex arrays, B-1
 - performing compensation (permeability) , 2-19
 - performing compensation (permittivity) , 2-16
 - performing fixture compensation , 2-8
 - presetting the analyzer , 2-1
 - putting numerical data into array , 8-6
 - putting numerical data into variable , 8-6
 - putting string variable (1) , 8-7
 - putting string variable (2) , 8-7
 - querying active channel setting , 2-2
 - reading an event register , 4-8
 - Reading data from I/O port , 6-2
 - reading data from I/O port to an external controller , 6-2
 - reading numerical array , 8-6
 - reading numerical variable , 8-6
 - reading string variable , 8-7
 - retrieving data from the analyzer using ASCII transfer , 3-8
 - saving the analyzer status , 8-1
 - selecting the connected fixture , 2-7, 2-19
 - selecting the connected fixture :
 - permittivity , 2-16
 - send data to I/O port , 6-2
 - send data to I/O port from an external controller , 6-2
 - sending data to analyzer by ASCII transfer , 3-8
 - sending data to analyze using binary transfer , 3-10
 - sending multiple commands in a line , 2-1
 - setting a log format , 2-10
 - Setting data array data , 3-2
 - setting data train for x- and y-axis for user trace , 7-3
 - setting dc bias , 2-11
 - setting dc current sweep , 2-5
 - setting dc voltage sweep , 2-5
 - setting frequency sweep , 2-4
 - setting grid for x-axis of user trace , 7-3
 - setting grid for y-axis of user trace , 7-3
 - setting measurement parameter , 2-9
 - setting measurement parameter (permeability) , 2-20
 - setting measurement parameter (permittivity) , 2-18
 - setting memory trace array data , 3-6
 - setting MUT size , 2-20
 - setting MUT thickness , 2-17
 - setting OSC level , 2-4
 - setting OSC level sweep , 2-4
 - setting the active channel , 2-3
 - setting the user fixture , 2-7
 - subtracting complex arrays, B-1
 - sweeping a specified number of times , 5-5
 - sweeping once using the GPIB trigger , 5-4
 - triggering on each point using the manual trigger , 5-6
 - turning off and clearing the user trace , 7-4
 - turning on the user trace , 7-4
 - uploading , 8-10
 - using marker on the user trace , 7-4
 - using PROG:WAIT? GPIB query , 8-9
- program module , 1-3
- Program Running bit , 8-8
- PROGram[:SElected]:NUMBer, 10-11
- PROGram subsystem , 8-6
- PURGE, 10-7
- putting numeric variable , 8-6

putting string variable , 8-7

Q

query , 2-2

query response , 2-2

Questionable Status Register, 11-108

Q value, 11-12

R

raw data , 3-2

raw data array, 11-28, C-33

reading an event register directly , 4-8

reading numeric variable , 8-6

reading string variable , 8-7

READIO , 6-2, 10-18

real-time clock , 8-12

reference , 2-13

reflection coefficients , 2-9

related documentation , v

remote mode , 1-14

required instrument , 1-1

RE-SAVE, 10-7

RESTMDISK , 9-4

rotary knob, 11-14

RQS , 4-2, 4-4, 4-9

*RST , 5-2

RUN/CONTInue input , 10-11

run light indication, 10-17

running a program through the softkey
interface, 9-2

S

sample program

loading , 1-15

Sample Program

basic impedance measurement program ,
2-14

BIN sorting using the I/O port , 6-3

loading compensation data from disk ,
3-13

performing calibration , 4-11

storing compensation data to disk , 3-11

time characteristic measurement using
user trace , 7-5

using complex operation sub program ,
B-2

sample program disk , 1-15

SAVE, 10-6

save programs , 9-1

scale , 2-13

SCPI, 1-7

SCPI , v, 5-1

SCPI command , 2-2

screen area

allocating, 10-2

select code, 10-2, 10-11

select code , 1-3

serial number , A-2

serial poll , 4-2

service request , 4-2

SET TIME, 10-19

SET TIMEDATE, 10-19

Shift, 10-14

Shift - **Delete**, 10-15

Shift - **End**, 10-15

Shift - **Insert**, 10-15

simple command, 1-7

single sweep , 5-4

sitmulus array, 11-30

Smith chart , 2-10

softkeys, 10-16

span, 11-138

SPOLL , 4-2

SRQ , 2-8, 2-12, 4-2, **4-9**

SRQ generation , 4-10

standard event status register , 4-6

status byte, 11-123, C-35

status byte register , 4-2, 4-4

status register

using , 4-8

status register structure , 4-4

status reporting system , 4-1

*STB , 4-2

stimulus, 11-99

stimulus , 2-4

stimulus array , 3-5

storage device, 9-4

STORMDISK , 9-2

<*string*>, 1-10

subsystem , 2-2

suffix, 1-10

sweep averaging, 11-76

sweep direction

dc bias , 2-5

OSC level , 2-5

sweeping a specified number of times , 5-4

sweeping once using the GPIB trigger , 5-4

sweep source, 11-99

sweep type, 11-47

system controller , 1-5

T

Tab, 10-14

test fixture

selecting , 2-7

TIME, 10-19

TIME\$, 10-19

TIMEDATE , 8-12

transferring a program , 10-10

transferring program source , 8-10

- transition filter , 4-3, 8-8
- trigger, 11-59, 11-119, C-35
- trigger , 2-12
- TRIGGER , 5-3
- triggering on each point using the manual trigger , 5-5
- trigger on point
 - using , 5-5
- trigger on point , 5-3, 5-6
- trigger on sweep
 - using , 5-4
- trigger on sweep , 5-3, 5-6
- trigger signal , 10-11
- trigger source , 5-2
- trigger system , 5-2

U

- ΔL value, 11-12
- ΔR value, 11-12
- uploading a program , 8-10

- user trace, 11-46, 11-118, C-35
 - clearing , 7-4
 - GPIB command , 7-2
 - log scale , 7-3
 - setting a grid , 7-3
 - setting data train , 7-3
 - turning on , 7-4
 - using marker , 7-4
- user trace , **7-1**
- using disks , 8-1

W

- WAIT , 6-2
- wait for trigger state , 5-2
- width. *See* bandwidth
- WIDTH, 10-7
- WRITEIO , 6-2, 10-20

Z

- zooming aperture, 11-27