FLUKE®
Calibration

96000 Series

RF Reference Source

Remote Programmers Manual

April 2014 Rev.1, 10/14
© 2014 Fluke Corporation. All rights reserved. Specifications are subject to change without notice.
All product names are trademarks of their respective companies.

LIMITED WARRANTY AND LIMITATION OF LIABILITY

Each Fluke product is warranted to be free from defects in material and workmanship under normal use and service. The warranty period is one year and begins on the date of shipment. Parts, product repairs, and services are warranted for 90 days. This warranty extends only to the original buyer or end-user customer of a Fluke authorized reseller, and does not apply to fuses, disposable batteries, or to any product which, in Fluke's opinion, has been misused, altered, neglected, contaminated, or damaged by accident or abnormal conditions of operation or handling. Fluke warrants that software will operate substantially in accordance with its functional specifications for 90 days and that it has been properly recorded on non-defective media. Fluke does not warrant that software will be error free or operate without interruption.

Fluke authorized resellers shall extend this warranty on new and unused products to end-user customers only but have no authority to extend a greater or different warranty on behalf of Fluke. Warranty support is available only if product is purchased through a Fluke authorized sales outlet or Buyer has paid the applicable international price. Fluke reserves the right to invoice Buyer for importation costs of repair/replacement parts when product purchased in one country is submitted for repair in another country.

Fluke's warranty obligation is limited, at Fluke's option, to refund of the purchase price, free of charge repair, or replacement of a defective product which is returned to a Fluke authorized service center within the warranty period.

To obtain warranty service, contact your nearest Fluke authorized service center to obtain return authorization information, then send the product to that service center, with a description of the difficulty, postage and insurance prepaid (FOB Destination). Fluke assumes no risk for damage in transit. Following warranty repair, the product will be returned to Buyer, transportation prepaid (FOB Destination). If Fluke determines that failure was caused by neglect, misuse, contamination, alteration, accident, or abnormal condition of operation or handling, including overvoltage failures caused by use outside the product's specified rating, or normal wear and tear of mechanical components, Fluke will provide an estimate of repair costs and obtain authorization before commencing the work. Following repair, the product will be returned to the Buyer transportation prepaid and the Buyer will be billed for the repair and return transportation charges (FOB Shipping Point).

THIS WARRANTY IS BUYER'S SOLE AND EXCLUSIVE REMEDY AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FLUKE SHALL NOT BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OR LOSSES, INCLUDING LOSS OF DATA, ARISING FROM ANY CAUSE OR THEORY.

Since some countries or states do not allow limitation of the term of an implied warranty, or exclusion or limitation of incidental or consequential damages, the limitations and exclusions of this warranty may not apply to every buyer. If any provision of this Warranty is held invalid or unenforceable by a court or other decision-maker of competent jurisdiction, such holding will not affect the validity or enforceability of any other provision.

Fluke Corporation
P.O. Box 9090
Everett, WA 98206-9090
U.S.A.

Fluke Europe B.V.
P.O. Box 1186
5602 BD Eindhoven
The Netherlands

11/99

Table of Contents

Title	Page
Introduction	1
Prepare the Instrument for Remote Operation	2
Equipment Connections	2
About the Bus Address	2
Set the Bus Address and Other Preferences	3
Switch to Remote Operation	4
Capability Codes	5
SCPI and IEEE Bus Descriptions	6
Reason for SCPI	6
Compatibility	6
Management and Maintenance of Programs	8
How Does SCPI Work in the Instrument?	9
Message Exchange Control Protocol	9
Protocol Requirements	10
Order of Execution - Deferred Commands	11
Sequential and Overlapped Commands	11
Remote Local Protocol	11
Program and Response Messages	12
Syntax and Style	12
Syntax of Program Messages	12
Syntax of Response Messages	15
Command Tree	16
Parameters	17
Numeric Data	17
Boolean Data	17
Other Data Types	17
Initialization and Resetting	18
Reset Strategy	18
Bus Initialization	18
Message Exchange Initialization	18
Device Initialization	19
The *RST Command	19
The *CLS Command	19
Status Reporting System	19
Introduction	19
Error Reporting	21

Read the Error/Event Queue 21
Standardized Error Numbers 21
Command Error 22
Execution Error 22
Device-specific Error 22
Query Error 22
Status Reporting Model 22
The Status Structure 22
Use the Registers 22
Status of the Output Queue (MAV) 23
Using the Status Byte 23
Select Summary Message to Generate SRQ 24
RQS/MSS 24
Set up the Instrument to Report Status 24
Read and Clear the Status 25
Status Byte 25
Status Event Registers 25
Status Condition Registers 25
Summary 26
Standard Event Status Registers 26
SCPI-defined Status Registers 28
Operation Status Group 28
Summary, Operation Status Reporting 28
Questionable Data/Signal Status Group 29
Power-on Status Clear 29
Preset the Status Reporting Structure 29
SCPI Commands 30
Definition of Common Parameter Forms 31
INSTrument Subsystem 32
OUTPut Subsystem 33
INPut Subsystem 34
POWer Subsystem 35
FREQuency Subsystem 36
AM Subsystem 37
FM Subsystem 38
PM Subsystem 39
SWEep Subsystem 40
PMETer Subsystem (96270A) 41
Frequency Counter Subsystem (96270A) 41
Trigger Subsystem 42
Profile Subsystem (96270A) 44
REFerence Subsystem 46
UNIT Subsystem 47
ROSCillator Subsystem 49
SYSTem Subsystem 49
STATus Subsystem 50
CALibration Subsystem 50
Common Commands 51
SCPI Status Registers 52
Operation Status Register 52
Questionable Status Register 53
Coupled Commands 54
What Is Command Coupling? 54
Coupled Command List 56
Programming Examples 58

Remote Programming Examples 58
Leveled Sine Output 58
AM Output 58
FM Output 59
Sweep Output 59
Leveled Sine Output with Offset 60
Frequency Measurement - simple 60
Frequency Measurement - triggered 60
Operation Status Register 60
SRQ Operation and Error Handling 61
HP 3335A Command Emulation 62
Prepare the Instrument for Remote 3335A Emulation 62
Commands that are Emulated 64
HP 8662A/8663A Command Emulation 66
Prepare the Instrument for Remote 8662/8663A Emulation 66
Emulated Commands 68
8662A/8663A Features Not Emulated 71
Error Message Matching 71
Request Service (RQS) Byte 72
HP 836xx Command Emulation (96270A) 73
Prepare the Instrument for Remote 836xx Emulation 73
Emulated Commands 75
HP 8340A Command Emulation (96270A) 91
Prepare the Instrument for Remote 8340A Emulation 91
Emulated Commands 93
Agilent E8257 Command Emulation (96270A) 95
Prepare the Instrument for Remote E8257 Emulation 95
Emulated Commands 97
IEEE Bus Trace Guide 103
The GPIB Trace Softkeys and Menus 104
Buffer Navigation 104
Display Formatting 104
Clearing a Trace 105
Export the Trace Data 105
Configure a Terminal Emulator 106
Construct a Null Modem Cable 106

List of Tables

Table	Title	Page
1.	GPIB Preferences	4
2.	IEEE 488.2 Interface Functions from the SCPI Command Set	5
3.	States for Message Exchange Protocol	10
4.	Actions for Message Exchange Protocol	10
5.	Standardized Errors	21
6.	List of Coupled Commands	56
7.	Emulated Commands	64
8.	Commands not Emulated	65
9.	Emulation Differences	65
10.	Emulated 8662A/8663 Commands	68
11.	Emulation Differences	71
12.	Emulated Bits	72
13.	836xx Command Set	75
14.	8340A Emulated Commands	93
15.	Special HP-IB Only Commands	94
16.	E8257 Emulated Commands	97

List of Figures

Figure	Title	Page
1.	GPIB Personality Screen	3
2.	Vertical Compatibility	6
3.	Horizontal Compatibility	7
4.	Overview of the Firmware in a SCPI Instrument	8
5.	Program and Response Messages	12
6.	Syntax of a Program Message Unit	12
7.	Syntax of a Terminated Program Message	12
8.	Example of a Terminated Program Message	13
9.	The SCPI Command Tree	13
10.	Syntax of a Response Message Unit	15
11.	Syntax of a Terminated Response Message	15
12.	Example of the TRIGger Subsystem Command Tree	16
13.	Instrument Status Register Structure	20
14.	Structural Overview of the Status Event Register	26
15.	Bits in the Standard Event Status Register	27
16.	Bus Command without Coupling	54
17.	Bus Command with Coupling	55
18.	GPIB Personality Screen	62
19.	3335 GPIB Preferences Screen	63
20.	GPIB Personality Screen	66
21.	8662 GPIB Preferences Screen	67
22.	GPIB Personality Screen	73
23.	836xx GPIB Preference Screen	74
24.	GPIB Personality Screen	91
25.	8340A GPIB Preferences Screen	92
26.	GPIB Personality Screen	95
27.	E8257 GPIB Preferences Screen	96
28.	GPIB Trace Screen	103
29.	GPIB Trace Display with both Timestamp and Hexadecimal On	104
30.	Clear Trace Confirmation Screen	105
31.	Exporting the Contents of a GPIB Trace Buffer	105

Introduction

This manual contains descriptions of the IEEE 488 bus and has these main sections:

- Generic SCPI and IEEE 488 Bus Descriptions
- SCPI Commands as they apply to the Instrument
- Instrument Programming Examples
- HP 3335A Command Emulation
- HP 8662A/8663A Command Emulation
- HP 836xx Command Emulation (96270A)
- HP 8340A Command Emulation (96270A)
- HP E8257 Command Emulation (96270A)
- Instrument IEEE Bus Trace Guide

The Instrument conforms to the Standard Specification IEEE 488.1 - 1987: *IEEE Standard Digital Interface for Programmable Instrumentation*, and to IEEE 488.2 - 1988: *Codes, Formats, Protocols and Common Commands*. In IEEE 488.2 terminology the Instrument is a device containing a system interface. It can be connected to a system via its system bus and set into programmed communication with other bus-connected devices under the direction of the system controller.

In a system, devices connected to the IEEE 488 bus are designated as talkers, listeners, talker/listeners, or controllers. The Instrument operates exclusively as a talker/listener on the IEEE 488 bus.

Prepare the Instrument for Remote Operation

The subsequent sections provide instructions to prepare the Instrument for remote operation.

Equipment Connections

The Instrument connects to the IEEE 488 bus using a standard IEEE 488 cable (not supplied with the Instrument). See *Chapter 3, Rear Panel Controls and Connectors* of the Operators Manual for the location of the connector.

⚠⚠Warning

To prevent possible electrical shock, fire, or personal injury, restrict the humidity of the operating environment to the level specified for any equipment used that conforms to IEC60950-1used with the Product.

⚠ Caution

To prevent equipment damage, remove power from both the Product and the IEEE 488 system before you connect or disconnect the Product to or from the IEEE 488 bus.

To connect the Instrument to an IEEE 488 system/controller:

1. Power down both the Instrument and the system/controller.
2. Connect one end of a standard IEEE 488 cable to the IEEE 488 connector on the rear of the Instrument.
3. Connect the other end of the IEEE 488 cable to the system/controller.
4. Power up both the Instrument and the system/controller.

After completing the equipment connections, set the bus address for the Instrument as described in the subsequent paragraphs.

About the Bus Address

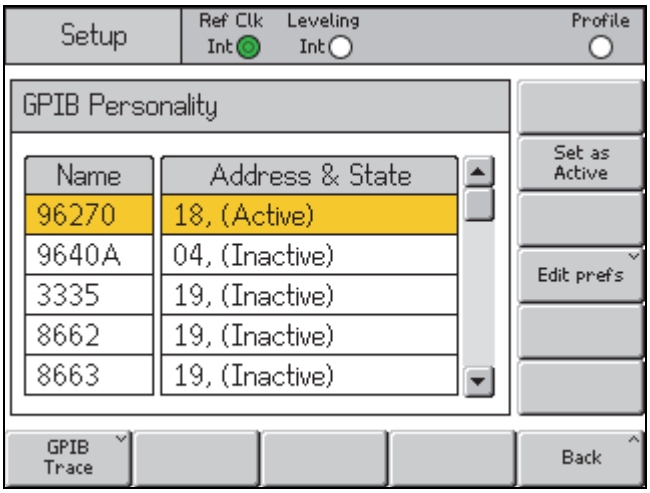
Each instrument in an IEEE 488 system requires a separate and unique address so the controller can call and communicate with each instrument individually. These bus addresses are numeric and are within the range of 0 to 30, inclusive. They are considered primary addresses, and the user can assign any one of them to the Instrument.

Secondary addressing is not available on the Instrument. In other words, the source cannot respond to any address outside the range of 0 to 30. When a controller addresses the Instrument, it must also send data to define and instruct the Instrument as a talker or listener.

Set the Bus Address and Other Preferences

To set the IEEE 488 bus address and other GPIB preferences:

1. From the front panel, push SETUP to show the Instrument Setup screen.
2. Push the GPIB Preferences softkey to show the GPIB Personality screen, see Figure .



ead32f.bmp

Figure 1. GPIB Personality Screen

Note

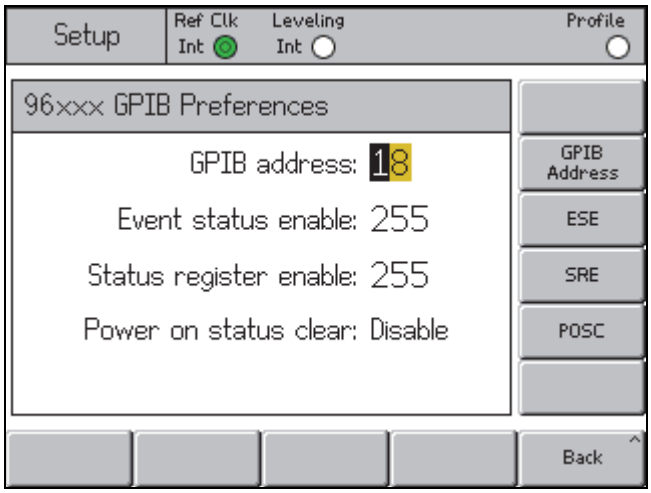
For instructions to set the parameters for a given GPIB personality, refer to the relevant part of this manual.

3. Use the cursor keys or spin wheel to select the 96270A. Then, select the Edit Pref's softkey to show the 96270A GPIB Preferences screen.
4. On the 96270A GPIB Preferences screen (see Table), select the GPIB Address field.
5. Enter the address (0 to 30) assigned to the Instrument. The default factory setting is 18.
6. Select the Event Status Enable field.
7. Enter the number of the GPIB Event Status Register.
8. Select the Status Register Enable field.

9. Enter the number of the GPIB Status Register.
10. Select the Power On Status Clear field and set a preference.

    The POSC setting determines whether or not the Instrument powers up with the PON bit of the Event Status Register set.
11. Push the Back softkey to save the Global Preferences settings and return to the Instrument Setup screen.

Table 1. GPIB Preferences



ead98f.bmp

Field	Preference
GPIB Address	0 to 30
Event Status Enable	0 to 255 (May also be set using SCPI *ESE command.)
Status Register Enable	0 to 255 (May also be set using SCPI *SRE command.)
Power On Status Clear	Enable, Disable

Switch to Remote Operation

When the Instrument is in local operation, the instrument is fully programmable both from the front panel and from the IEEE 488 bus. There is no physical switch for selecting remote operation. Rather, when the Instrument receives a bus message it switches to remote operation. If the message arrives while a change is being entered from the front panel, the front panel entry is interrupted and then the bus message is executed. Once the Instrument is set to remote operation all of the front panel (local) controls are locked out (inoperable) with the exception of the STBY key and the Go to Local softkey at the bottom of the display. Pushing the GO to Local softkey causes the Instrument to return to local operation.

Capability Codes

Table shows the IEEE 488.2 interface functions from the SCPI command set. These commands define the interface capabilities of the Instrument.

Table 2. IEEE 488.2 Interface Functions from the SCPI Command Set

Description	Code	Description
Instrument Handshake	SH1	The Instrument can exchange data with other instruments or a controller using the bus handshake lines: DAV, NRFD, and NADC.
Acceptor Handshake	AH1	The Instrument can exchange data with other instruments or a controller using the bus handshake lines: DAV, NRFD, and NADC.
Control Function	C0	The Instrument does not function as a controller.
Talker Function	T6	The Instrument can send responses and the results of its settings to other devices or to the controller. T6 means that it has the following functions: • Basic talker. • No talker only. • It can send out a status byte as response to a serial poll from the controller. Automatic un-addressing as a talker when it is addressed as a listener.
Listener Function	L4	The Instrument can receive programming instructions from the controller. L4 means that it has the following functions: • Basic listener. • No listen only. • Automatic un-addressing as listener when addressed as a talker.
Service Request	SR1	The Instrument can call for attention from the controller, e.g., when a response is available or an error has occurred.
Remote/Local Function	RL1	You can control the Instrument manually (locally) from the front panel or remotely from the controller. The LLO, local-lock-out function, can disable the LOCAL button on the front panel.
Parallel Poll	PP0	The Instrument does not have any parallel poll facility.
Device Clear Function	DC1	The controller can reset the Instrument via interface message DCL (Device clear) or SDC (Selective Device Clear).
Device Trigger Function	DT0	The Instrument supports GET (Group Execute Trigger) for the frequency counter and Power Meter.
Bus Drivers	E2	The GPIB interface has tri-state bus drivers.

SCPI and IEEE Bus Descriptions

SCPI (Standard Commands for Programmable Instruments) is a standardized set of commands used to remotely control programmable test and measurement instruments. The Instrument firmware contains the SCPI. It defines the syntax and semantics that the controller must use to communicate with the Instrument.

This section is an overview of SCPI and shows how SCPI is used in the Instrument.

SCPI is based on IEEE-488.2 to which it owes much of its structure and syntax. SCPI can, however, be used with any of the standard interfaces, such as GPIB (IEC625/IEEE 488), VXI and RS-232.

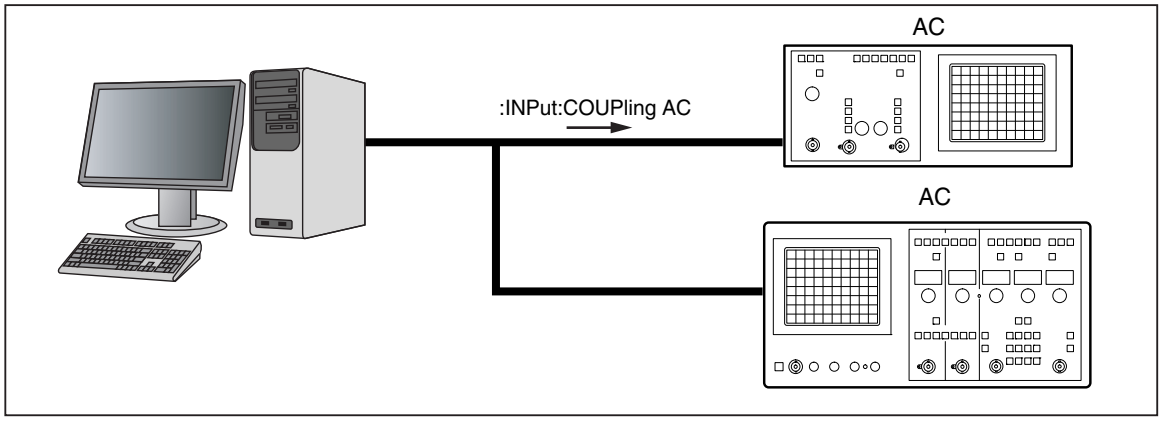
Reason for SCPI

For each Instrument function, SCPI defines a specific command set. The advantage of SCPI is that programming an instrument is only function dependent and no longer instrument dependent. Several different types of instruments, for example an oscilloscope, an Instrument and a multimeter, can carry out the same function, such as frequency measurement. If these instruments are SCPI compatible, you can use the same commands to measure the frequency on all three instruments, although there may be differences in accuracy, resolution, speed, etc.

Compatibility

SCPI provides two types of compatibility, vertical and horizontal.

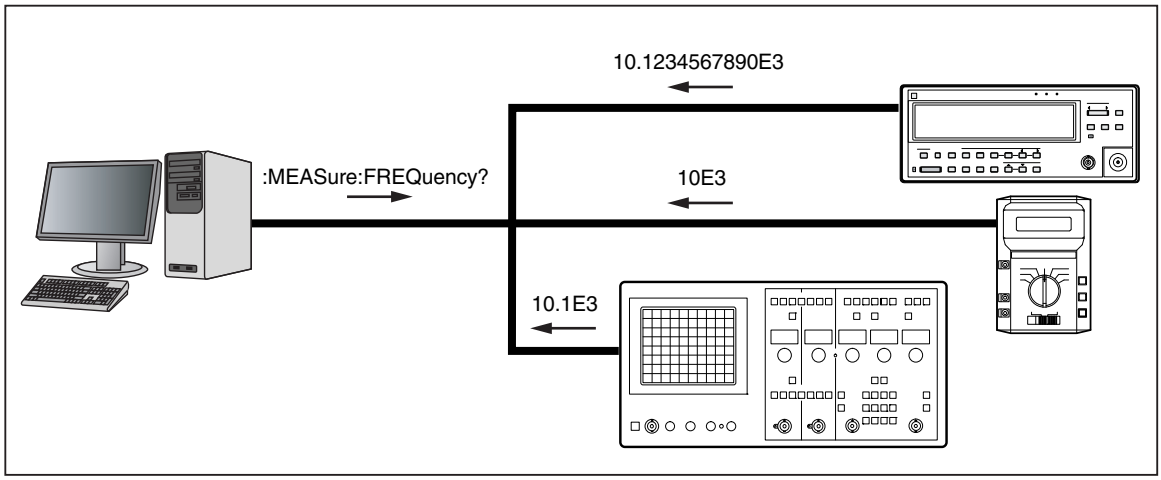
Vertical compatibility means that all instruments of the same type have identical controls. For example, oscilloscopes will have the same controls for timebase, triggers and voltage settings. See Figure 2.



ead100f.eps

Figure 2. Vertical Compatibility

Horizontal compatibility means that instruments of different types that perform the same functions have the same commands. For example, a DMM, an oscilloscope, and a source can all measure frequency with the same commands. See Figure 3.



ead101f.eps

Figure 3. Horizontal Compatibility

Management and Maintenance of Programs

SCPI simplifies maintenance and management of the programs. Today changes and additions in a good working program are hardly possible because of the great diversity in program messages and instruments. Programs are difficult to understand for anyone other than the original programmer. After some time even the programmer may be unable to understand them.

A programmer with SCPI experience, however, will understand the meaning and reasons of a SCPI program, because of his knowledge of the standard. Changes, extensions, and additions are much easier to make in an existing application program. SCPI is a step towards portability of instrument programming software and, as a consequence, it allows the exchange of instruments. Figure provides an overview of the firmware in a SCPI instrument.

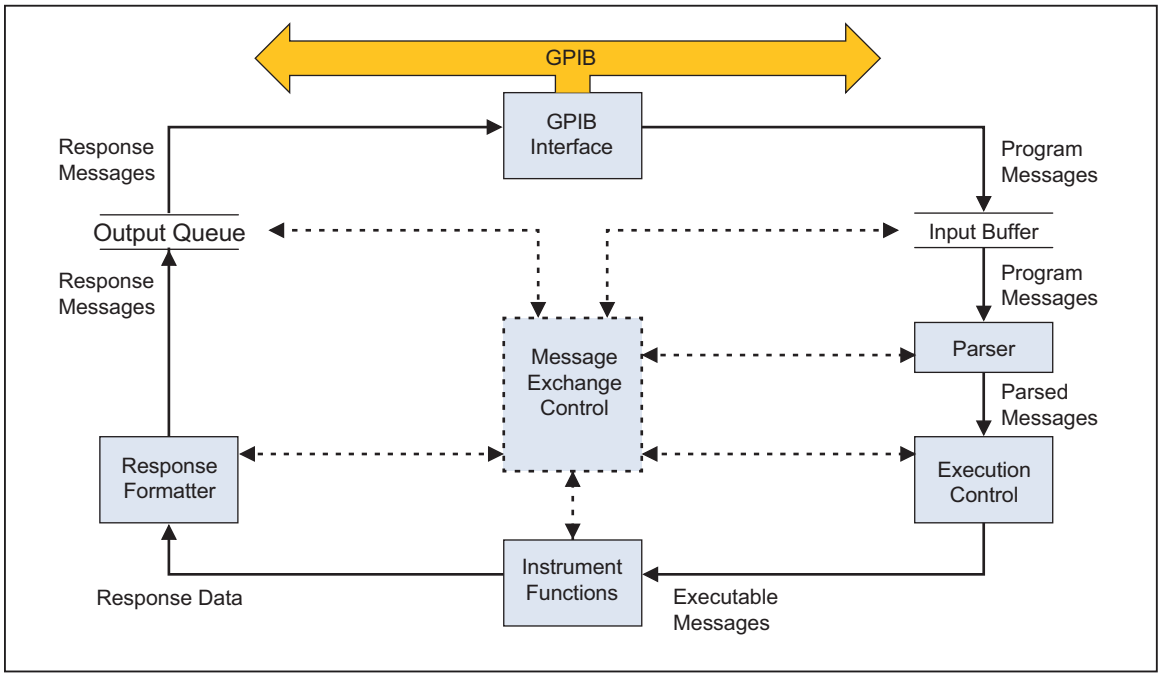


Figure 4. Overview of the Firmware in a SCPI Instrument

How Does SCPI Work in the Instrument?

The functions inside an Instrument that control the operation provide SCPI compatibility. Figure shows a simplified logical model of the message flow inside a SCPI instrument.

When the controller sends a message to a SCPI instrument, this is basically what occurs:

- The GPIB controller addresses the Instrument as listener.
- The GPIB interface function places the message in the Input Buffer.
- The Parser fetches the message from the Input Buffer, parses (decodes) the message, and checks for the correct syntax. The Instrument reports incorrect syntax by sending command errors via the status system to the controller. Moreover, the parser will detect if the controller requires a response. This is the case when the input message is a query (command with a “?” appended).

The Parser will transfer the executable messages to the Execution Control block in token form (internal codes). The Execution Control block will gather the information required for a device action and will initiate the requested task at the appropriate time. The instrument reports execution errors via the status system over the GPIB and places them in the Error Queue.

When the controller addresses the Instrument as talker, the Instrument takes data from the Output Queue and sends it over the GPIB to the controller.

Message Exchange Control Protocol

Another important function is the Message Exchange Control, defined by IEEE 488.2. The Message Exchange Control protocol specifies the interactions between the several functional elements that exist between the GPIB functions and the device-specific functions.

The Message Exchange Control protocol specifies how the Instrument and controller should exchange messages. For example, it specifies exactly how an instrument shall handle program and response messages that it receives from and returns to a controller.

This protocol introduces the idea of commands and queries; queries are program messages that require the device to send a response. When the controller does not read this response, the device will generate a Query Error. On the other hand, commands will not cause the device to generate a response. When the controller tries to read a response anyway, the device then generates a Query Error.

The Message Exchange Control protocol also deals with the order of execution of program messages. It defines how to respond if Command Errors, Query Errors, Execution Errors, and Device-Specific errors occur. The protocol demands that the Instrument report any violation of the IEEE-488.2 rules to the controller, even when it is the controller that violates these rules.

The IEEE 488.2 standard defines a set of operational states and actions to implement the message exchange protocol. See Table and Table .

Table 3. States for Message Exchange Protocol

State	Purpose
IDLE	Wait for messages
READ	Read and execute messages
QUERY	Store responses to be sent
SEND	Send responses
RESPONSE	Complete sending responses
DONE	Finished sending responses
DEADLOCK	The device cannot buffer more data

Table 4. Actions for Message Exchange Protocol

Action	Reason
Unterminated	The controller attempts to read the device without first having sent a complete query message.
Interrupted	The device is interrupted by a new program message before it finishes sending a response message.

Protocol Requirements

In addition to the above functional elements, which process the data, the message exchange protocol has the following characteristics:

- The controller must end a program message containing a query with a message terminator before reading the response from the device (address the device as talker). If the controller breaks this rule, the device will report a query error (unterminated action).
- The controller must read the response to a query in a previously (terminated) program message before sending a new program message. When the controller violates this rule, the device will report a query error (interrupted action).
- The Instrument sends only one response message for each query message. If the query message resulted in more than one answer, all answers will be sent in one response message.

Order of Execution - Deferred Commands

Execution control collects commands until the end of the message, or until it finds a query or other special command that forces execution. It then checks that the setting resulting from the commands is a valid one: No range limits are exceeded, no coupled parameters are in conflict, etc. If this is the case, the commands are executed in the sequence they have been received; otherwise, an execution error is generated, and the commands are discarded.

This deferred execution guarantees:

- All valid commands received before a query are executed before the query is executed.
- All queries are executed in the order they are received.
- The order of execution of commands is never reversed.

Sequential and Overlapped Commands

SCPI defines two classes of commands: sequential and overlapped commands. All commands in the Instrument are sequential, that is one command finishes before the next command executes.

Remote Local Protocol

Definitions Remote Operation

When an Instrument operates in remote, all local controls, except the Go To Local softkey and STBY are disabled.

Local Operation

An Instrument operates in local when it is not in remote mode as defined above.

Local Lockout

In addition to the remote state, an Instrument can be set to remote with ‘local lockout’. This disables the return-to-local button. In theory, the state local with local lockout is also possible; then, all local controls except the return-to-local key are active.

The Instrument in Remote Operation

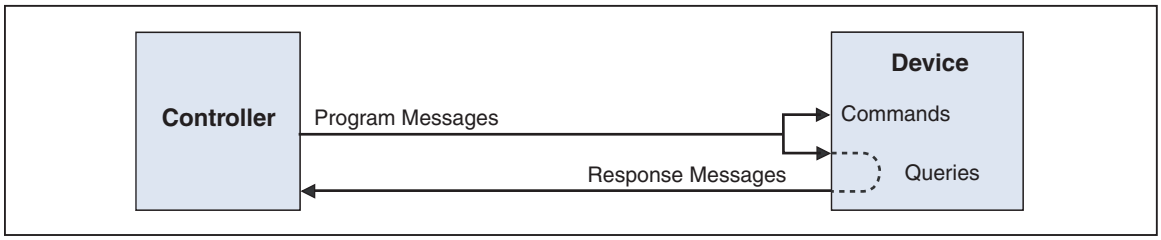
When the Instrument is in remote operation, it disables all its local controls except the Go To Local softkey.

The Instrument in Local Operation

When the Instrument is in local operation, the Instrument is fully programmable both from the front panel and from the bus. If a bus message arrives while a change is being entered from the front panel, the front panel entry is interrupted and the bus message is executed.

Program and Response Messages

The communication between the system controller and the SCPI instruments connected to the GPIB takes place through Program and Response Messages. See Figure . A Program Message is a sequence of one or more commands sent from the controller to an instrument. Conversely, a Response Message is the data from the Instrument to the controller.



ead103.eps

Figure 5. Program and Response Messages

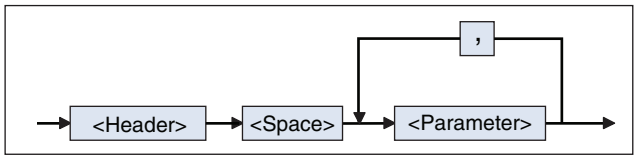
The GPIB controller instructs the device through program messages. The device will only send responses when explicitly requested to do so; that is, when the controller sends a query. Queries are recognized by the question mark at the end of the header, for example: *IDN? (requests the instrument to send identity data).

Syntax and Style

The subsequent sections describe the syntax of program and response messages.

Syntax of Program Messages

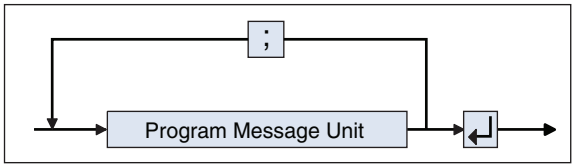
A command or query is called a program message unit. A program message unit consists of a header followed by one or more parameters, as shown in Figure .



ead104f.eps

Figure 6. Syntax of a Program Message Unit

One or more program message units (commands) may be sent within a simple program message, see Figure .



ead105f.eps

Figure 7. Syntax of a Terminated Program Message

The ↵ is the pmt (program message terminator) and it must be one of the following codes:

Note

- *NL is the same as the ASCII LF*
- *LF (<line feed> = ASCII 10 decimal)*
- *The END message is sent via the EOI-line of the GPIB.*
- *The ^ character stands for "at the same time".*

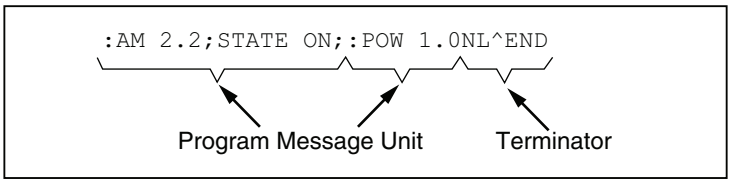
NL^END This is <new line> code sent concurrently with the END message on the GPIB.

NL This is the <new line> code.

<dab>^END This is the END message sent concurrently with the last data byte <dab>.

Most controller programming languages send these terminators automatically, but allow changing it. Make sure that the terminator is as above.

Figure is an example of a terminated program message:



ead106f.eps

Figure 8. Example of a Terminated Program Message

This program message consists of two message units. The unit separator (semi-colon) separates message units.

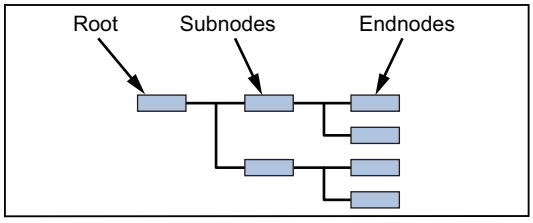
Basically there are two types of commands: common commands and SCPI commands.

Common Commands

The common command header starts with the asterisk character (*), for example *RST.

SCPI Commands

SCPI command headers may consist of several keywords (mnemonics), separated by the colon character (:). An sample of the SCPI command tree structure is shown in Figure .



ead107f.eps

Figure 9. The SCPI Command Tree

Each keyword in a SCPI command header represents a node in the SCPI command tree. The leftmost keyword (AM in the previous example) is the root level keyword, representing the highest hierarchical level in the command tree.

The keywords following represent subnodes under the root node. See the *Command Tree* section of this manual for more details of this subject.

Forgiving Listening

The syntax specification of a command is as follows:

POWer:OFFSet <numeric value>

Where: POW and OFFS specify the shortform, and POWer and OFFSet specify the longform. However, POWE or OFF are not allowed and cause a command error.

In program messages either the long or the shortform may be used in upper or lower case letters. You may even mix upper and lower case. There is no semantic difference between upper and lower case in program messages. This instrument behavior is called forgiving listening.

For example, an application program may send the following characters over the bus:

SEND=> pOwEr:OFfSeT 1.23

The example shows the shortform used in a mix of upper and lower case

SEND=> Power:Offs 1.23

The example shows a mix of longform and shortform and a mix of upper and lower case.

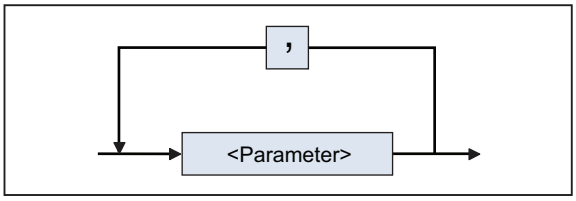
Notation Habit in Command Syntax

To clarify the difference between the forms, the shortform in a syntax specification is shown in upper case letters and the remaining part of the longform in lower case letters.

Notice however, that this does not specify the use of upper and lower case characters in the message that you actually sent. Upper and lower case letters, as used in syntax specifications, are only a notation convention to ease the distinction between longform and shortform.

Syntax of Response Messages

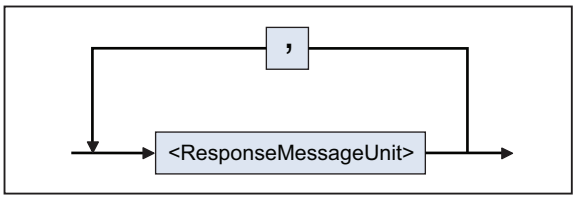
The response of a SCPI instrument to a query (response message unit) consists of one or more parameters (data elements) as shown in Figure . There is no header returned.



ead108f.eps

Figure 10. Syntax of a Response Message Unit

If there are multiple queries in a program message, the instrument groups the multiple response message units together in one response message according to the syntax shown in Figure .



ead109f.eps

Figure 11. Syntax of a Terminated Response Message

The response message terminator (rmt) is always NL^END, where:

NL^END is <new line> code (equal to <line feed> code = ASCII 10 decimal) sent concurrently with the END message. The END message is sent by asserting the EOI line of the GPIB bus.

Responses:

A SCPI instrument always sends its response data in shortform and in capitals.

Example:

You program an instrument with the following command:

SEND=> :ROSCillator:SOURce EXTernal

Then you send the following query to the instrument:

SEND=> :ROSCillator:SOURce?

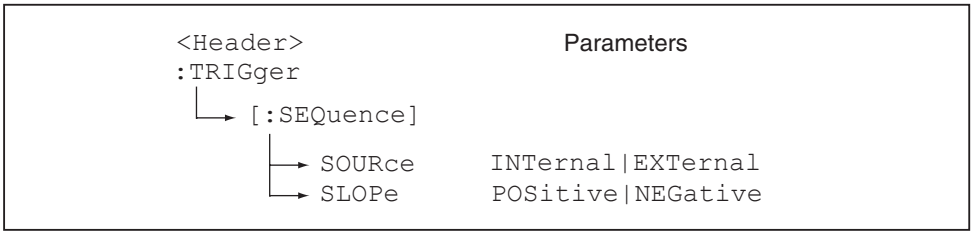
The instrument will return:

READ<= EXT

response in shortform and in capitals.

Command Tree

Command Trees like the one shown in Figure are used to document the SCPI command set in this manual. The keyword (mnemonic) on the root level of the command tree is the name of the subsystem. The following example illustrates the Command Tree of the TRIGger subsystem.



ead110f.eps

Figure 12. Example of the TRIGger Subsystem Command Tree

The keywords placed in square brackets are optional nodes. This means that you may omit them from the program message.

Example:

SEND=> TRIGger:SEQuence:SOURce INTernal

is the same as

SEND=> TRIGger:SOURce INTernal

The command tree shows the paths you should use for the command syntax. A single command header begins from the root level downward to the 'leaf nodes' of the command tree. (Leaf nodes are the last keywords in the command header, before the parameters.)

Example:

SEND=> TRIGger:SEQuence:SOURce INTernal

Where: TRIGger is the root node and SEQuence is the leaf node.

Each colon in the command header moves the current path down one level from the root in the command tree. Once you reach the leaf node level in the tree, you can add several leaf nodes without having to repeat the path from the root level.

Just follow the rules below:

You can only do this if the header path of the new leaf-node is the same as that of the previous one. If not, the full header path must be given starting with a colon.

Command header = Header path + leaf node

Once you send the pmt (program message terminator), the first command in a new program message must start from the root.

Always give the full header path, from the root, for the first command in a new program message.

For the following commands within the same program message, omit the header path and send only the leaf node (without colon).

Example:

SEND=> TRIGger:SEQuence:SOURce INTernal;SLOPe POSitive

This is the command where:

TRIGger:SEQuence is the header path and :SOURce is the first leaf-node and SLOPe is the second leaf-node because SLOPe is also a leaf-node under the header path TRIGger:SEQuence.

The important point to note here is that there is no colon before SLOPe.

Parameters

Numeric Data

Decimal data are printed as numerical values throughout this manual. Numeric values may contain both a decimal point and an exponent (base 10).

These numerals are often represented as NRf (NR = NumeRic, f = flexible) format.

Boolean Data

A Boolean parameter specifies a single binary condition which is either true or false.

Boolean parameters can be one of the following:

- ON or 1 means condition true.
- OFF or 0 means condition false.

Example

SEND=> :OUTP:STATe ON

SEND=> OUTP:STATe 1

This switches signal source output on.

A query, for instance OUTP:STATe?, will return 1 or 0; never ON or OFF.

Other Data Types

Other data types that can be used for parameters are the following:

String data	Always enclosed between single or double quotes, for example “This is a string” or ‘This is a string.’
Character data	For this data type, the same rules apply as for the command header mnemonics. For example: POSitive, NEGative.

Initialization and Resetting

Reset Strategy

The levels of initialization are:

- Bus initialization
- Message exchange initialization
- Device initialization

Bus Initialization

This is the first level of initialization. The controller program should start with this, which initializes the IEEE-interfaces of all connected instruments. It puts the complete system into remote enable (REN-line active) and the controller sends the interface clear (IFC) command. The command or the command sequence for this initialization is controller and language dependent. Refer to the user manual of the system controller in use.

Message Exchange Initialization

Device clear is the second level of initialization. It initializes the bus message exchange, but does not affect the device functions.

Device clear can be signaled either with DCL to all instruments or SDC (Selective device-clear) only to the addressed instruments. The instrument action on receiving DCL and SDC is identical, they will do the following:

- Clear the input buffer.
- Clear the output queue.
- Reset the parser.
- Clear any pending commands

The device-clear commands will not do the following:

- Change the Instrument settings or stored data in the Instrument.
- Interrupt or affect any device operation in progress.
- Change the status byte register other than clearing the MAV bit as a result of clearing the output queue.

Many older IEEE-instruments that are not IEEE-488.2 compatible returned to the power-on default settings when receiving a device-clear command. IEEE-488.2 does not allow this.

When to use a Device-clear Command

The command is useful to escape from erroneous conditions without having to alter the current settings of the Instrument. The Instrument will then discard pending commands and will clear responses from the output queue. For example, suppose you are using the Instrument in an automated test equipment system where the controller program returns to its main loop on any error condition in the system or the tested unit. To ensure that no unread query response remains in the output queue and that no unparsed message is in the input buffer, it is wise to use device-clear. (Such remaining responses and commands could influence later commands and queries.)

Device Initialization

The third level of initialization is on the device level. This means that it concerns only the addressed instruments.

*The *RST Command*

Use this command to reset a device. It initializes the device-specific functions in the Instrument.

The following happens when using the *RST command:

- The instrument-specific functions are set to a known default state. The *RST condition for each command is given in the command reference section.
- The Instrument is set to an idle state (outputs are disabled), so that it can start new operations.

*The *CLS Command*

Use this command to clear the status data structures. See *Status Reporting System*.

The following happens when you use the *CLS command:

- The Instrument clears all event registers summarized in the status byte register.
- It empties all queues, which are summarized in the status byte register, except the output queue, which is summarized in the MAV bit.

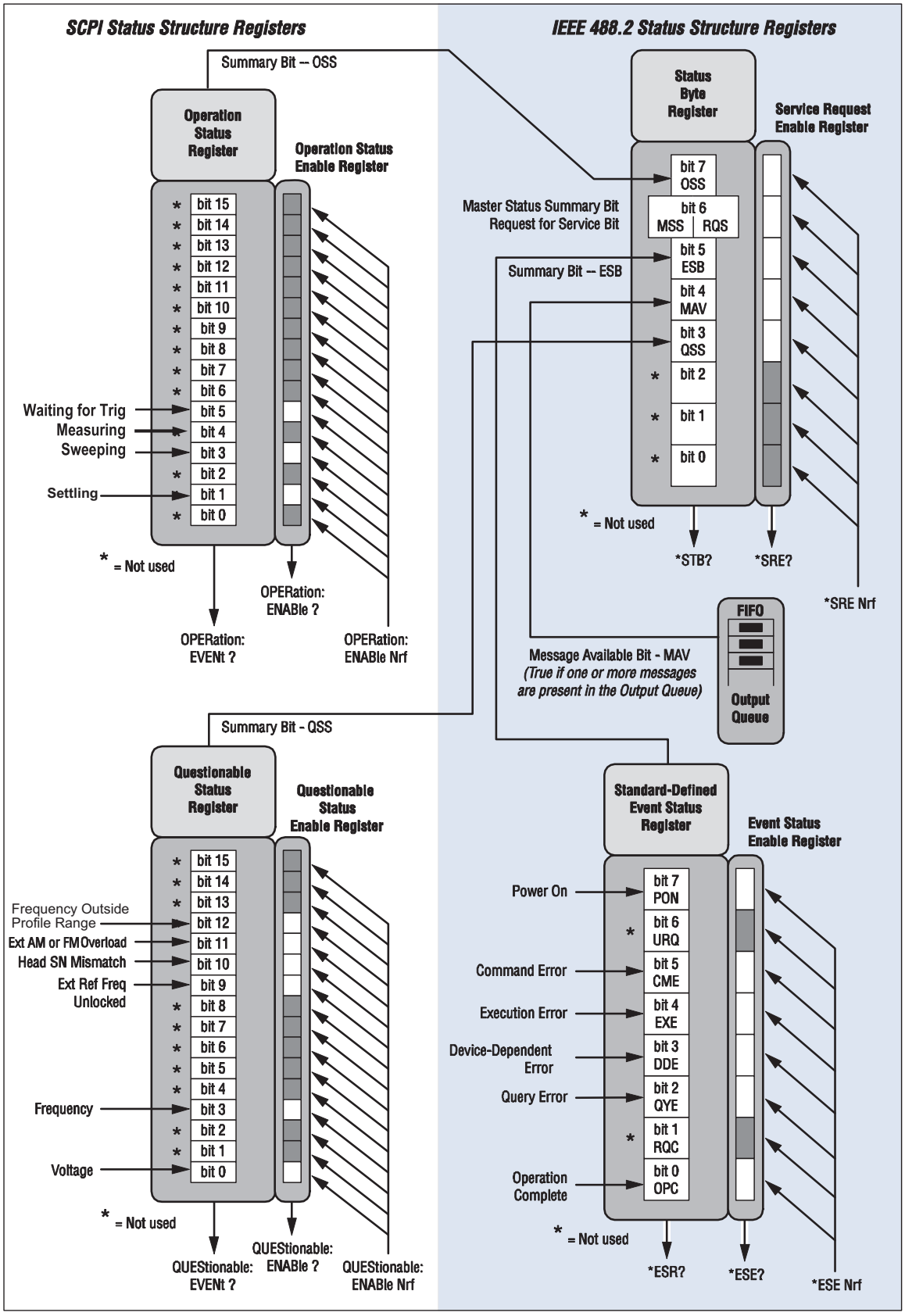
Status Reporting System

Introduction

Status reporting is a method to let the controller know what the Instrument is doing. You can ask the Instrument what status it is in whenever you want to know.

You can select some conditions in the Instrument that should be reported in the Status Byte Register. You can also select if some bits in the Status Byte should generate a Service Request (SRQ). See Figure 13 for an overview of the Status Register Structure.

(An SRQ is the Instrument's way to call the controller for help.)



ead111f.eps

Figure 13. Instrument Status Register Structure

Error Reporting

The Instrument will place a detected error in its Error Queue. This queue is a FIFO (First-In First-Out) buffer. When you read the queue, the first error will come out first, the last error last.

If the queue overflows, an overflow message is placed last in the queue, and further errors are thrown away until there is room in the queue again.

Read the Error/Event Queue

This is done with the :SYSTem:ERRor? query.

Example

SEND=> :SYSTem:ERRor?

READ<= –100,“Command-Error”

The query returns the error number followed by the error description.

If more than one error occurred, the query will return the error that occurred first. When you read an error you will also remove it from the queue. You can read the next error by repeating the query. When you have read all errors the queue is empty, and the :SYSTem:ERRor? query will return:

0,“No error”

When errors occur and you do not read these errors, the Error Queue may overflow. Then the Instrument will overwrite the last error in the queue with the following:

–350,“Queue overflow”

If more errors occur, they will be discarded.

Standardized Error Numbers

The Instrument reports four classes of standardized errors in the Standard Event Status and in the Error/Event Queue as shown in the Table .

Table 5. Standardized Errors

Error Class	Range of Error Numbers	Standard Event Register
Command Error	–100 to –199	bit 5 - CME
Execution Error	–200 to –299	bit 4 - EXE
Device Specific Error	–300 to –399	bit 3 - DDE
	+100 to +32767	
Query Error	–400 to –499	bit 2 -QYE

Command Error

This error shows that the Instrument detected a syntax error.

Execution Error

This error shows that the Instrument has received a valid program message which it cannot execute because of some device specific conditions.

Device-specific Error

This error shows that the Instrument could not properly complete some device specific operations.

Query Error

This error will occur when the Message Exchange Protocol is violated, for example, when you send a query to the Instrument and then send a new command without first reading the response data from the previous query. Also, trying to read data from the Instrument without first sending a query to the Instrument will cause this error.

Status Reporting Model

The Status Structure

The status reporting model used is standardized in IEEE 488.2 and SCPI, so you will find similar status reporting in most modern instruments. Figure shows an overview of the complete status register structure.

- The Standard Event Register reports the standardized IEEE 488.2 errors and conditions.
- The Questionable Data Register reports when the output data from the Instrument may not be trusted.
- The Operational Data Register reports what events are in operation.
- The Status Byte contains eight bits. Each bit shows if there is information to be fetched in the above described registers and queues of the status structure.

Use the Registers

Each status register monitors several conditions at once. If something happens to any one of the monitored conditions, a summary bit is set true in the Status Byte Register.

Enable registers are available so that you can select what conditions should be reported in the status byte, and what bits in the status byte should cause SRQ.

A register bit is TRUE, i.e., something has happened, when it is set to 1. It is FALSE when set to 0.

Note that all event registers and the status byte record positive events. That is when a condition changes from inactive to active, the bit in the event register is set true. When the condition changes from active to inactive, the event register bits are not affected at all.

When reading the contents of a register, the Instrument answers with the decimal sum of the bits in the register.

Example:

The Instrument answers 40 when you ask for the contents of the Standard Event Status Register. Convert this to binary form. It will give you 101000.

- Bit 5 is true showing that a command error has occurred.
- Bit 3 is also true, showing that a device dependent error has occurred.

Use the same technique when you program the enable registers.

1. Select which bits should be true.
2. Convert the binary expression to decimal data.
3. Send the decimal data to the Instrument.

Clearing/Setting all bits

Clear an enable register by programming it to zero. To set all bits true in a 16-bit event enable register program it to 32767 (bit 16 not used).

To set all bits true in an 8-bit registers program it to 255 (Service Request Enable and Standard Event Enable.)

Status of the Output Queue (MAV)

The MAV (message available) queue status message appears in bit 4 of the status byte register. It indicates if there are bytes ready to be read over the GPIB in the GPIB output queue of the instrument. The output queue is where the formatted data appears before it is transferred to the controller.

The controller reads this queue by addressing the instrument as a talker.

Using the Status Byte

The status byte is an eight-bit status message. It is sent to the controller as a response to a serial poll or a *STB? query. Each bit in the status byte contains a summary message from the status structure. You can select what bits in the status byte should generate a service request to alert the controller.

When a service request occurs, the SRQ-line of the GPIB will be activated. Whether or not the controller will react on the service request depends on the controller program. The controller may be interrupted on occurrence of a service request, it may regularly test the SRQ-line, it may regularly make serial poll or *STB?, or the controller may not react at all. The preferred method is to use SRQ because it presents a minimum of disturbance to the measurement process.

Select Summary Message to Generate SRQ

The Instrument does not generate any SRQ by default. You must first select which summary message(s) from the status byte register should give SRQ. You do that with the Service Request Enable command *SRE <bit mask>.

Example

*SRE 32

This sets bit 4 (16=00010000_2) in the service request enable register. This makes the instrument signal SRQ when a message is available in the output queue.

RQS/MSS

The original status byte of IEEE 488.1 is sent as a response to a serial poll, and bit 6 means requested service, RQS.

IEEE 488.2 added the *STB? query and expanded the status byte with a slightly different bit 6, the MSS. This bit is true as long as there is unfetched data in any of the status event registers.

- The Requested Service bit, RQS, is set true when a service request has been signaled. If you read the status byte via a Serial Poll, bit 6 represents RQS. Reading the status byte with a serial poll will set the RQS bit false, showing that the status byte has been read.
- The Master Summary Status bit, MSS, is set true if any of the bits that generates SRQ is true. If you read the status byte using *STB?, bit 6 represents MSS. MSS remains true until all event registers are cleared and all queues are empty.

Set up the Instrument to Report Status

To use the status reporting feature, include the following steps in your program.

*CLS clears all event registers and the error queue.

*ESE <bit mask> selects what conditions in the Standard Event Status register should be reported in bit 5 of the status byte.

:STATus:OPERation:ENABle <bit mask> selects which conditions in the Operation Status register should be reported in bit 7 of the status byte.

:STATus:QUEStionable:ENABle <bit mask> selects which conditions in the Questionable Status register should be reported in bit 3 of the status byte.

*SRE <bit mask> selects which bits in the status byte should cause a Service Request.

Read and Clear the Status

Status Byte

There are two way to read the status byte register: Using the Serial Poll and using the Common Query.

Using the Serial Poll (IEEE-488.1 defined)

Response:

- Bit 6: RQS message shows that the Instrument has requested service via the SRQ signal.
- Other bits show their summary messages
- A serial poll sets the RQS bit FALSE, but does not change other bits.

Using the Common Query *STB?

Response:

- Bit 6: MSS message shows that there is a reason for service request.
- Other bits show their summary messages.
- Reading the response will not alter the status byte.

Status Event Registers

Use the following queries to read the Status Event registers:

*ESR? reads the Standard Event Status register

:STATus:OPERation? reads the Operation Status Event register

:STATus:QUEStionable? reads the Questionable Status Event register

Reading one of these registers will clear the register and the summary message bit in the status byte.

To clear all event registers use the *CLS (Clear Status) command.

Status Condition Registers

Two of the status register structures also have condition registers: The Status Operation and the Status Questionable register.

The condition registers differ from the event registers in that they are not latched. That is, if a condition in the Instrument goes on and then off, the condition register indicates true while the condition is on and false when the condition goes off. The Event register that monitors the same condition continues to indicate true until you read the register.

:STATus:OPERation:CONDition? reads the Operation Status Condition register

:STATus:QUEStionable:CONDition? reads the Questionable Status Condition register

Reading the condition register will not affect the contents of the register.

Summary

The way to work when writing your bus program is as follows:

Set up

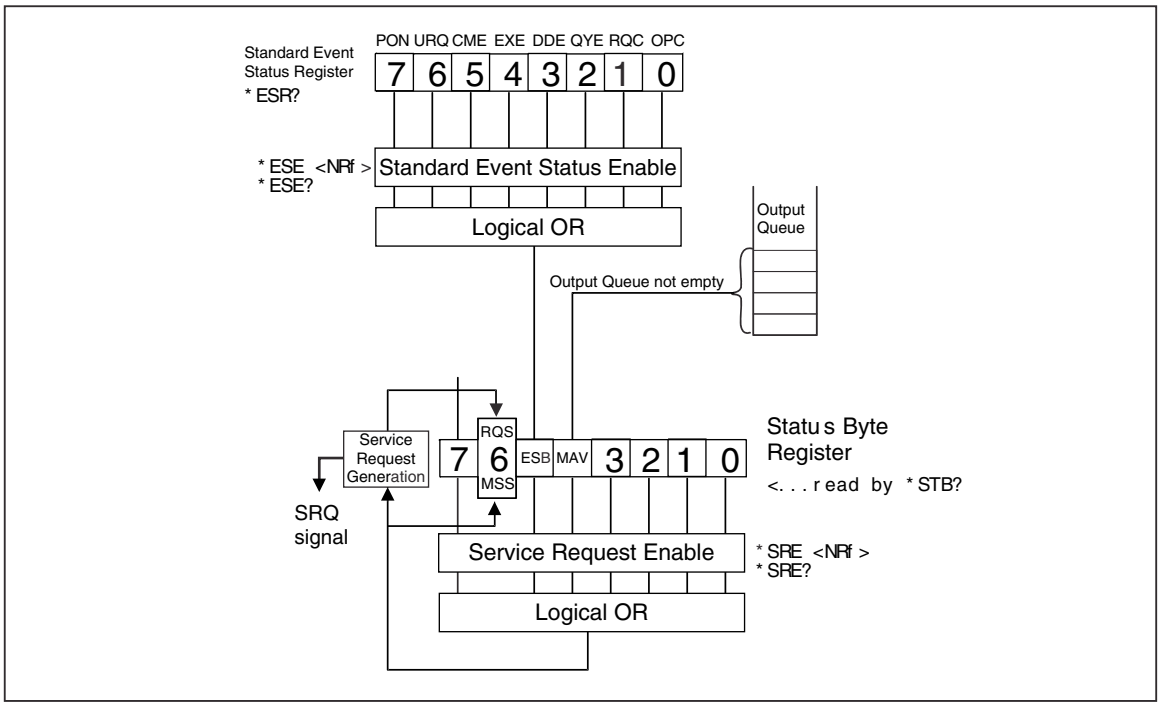
- Set up the enable registers so that the events you are interested in are summarized in the status byte.
- Set up the enable masks so that the conditions you want to be alerted about generate SRQ. It is good practice to generate SRQ on the MAV bit. So, enable the MAV-bit via *SRE.

Check & Action

- Check if an SRQ has been received.
- Make a serial poll of the instruments on the bus until you find the instrument that issued the SRQ (the instrument that has RQS bit true in the Status Byte).
- When you find it, check which bits in the Status Byte Register are true.
- Let's say that bit 7, OSS, is true. Then read the contents of the Operation Status Register. In this register you can see what caused the SRQ.
- Take appropriate actions depending on the reason for the SRQ.

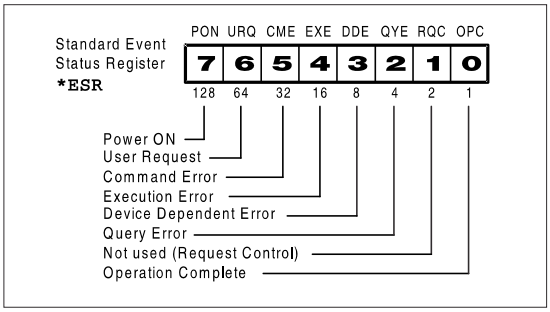
Standard Event Status Registers

The Event Status registers are mandatory in all instruments that fulfill the IEEE 488.2 standard. They are structured as shown in Figure , and an overview of the status bits is shown in Figure .



ead114f.eps

Figure 14. Structural Overview of the Status Event Register



ead115f.eps

Figure 15. Bits in the Standard Event Status Register

Bit 7 (weight 128) — Power-on (PON)

Shows that the Instrument power supply has been turned off and on (since the last time the controller read or cleared this register).

Bit 6 (weight 64)—User Request (URQ)

Shows that the user has pushed a key on the front panel. This is not implemented on the Instrument.

Bit 5 (weight 32) — Command Error (CME)

Shows that the Instrument has detected a command error. This means that it has received data that violates the syntax rules for program messages.

Bit 4 (weight 16) — Execution Error (EXE)

Shows that the Instrument detected an error while trying to execute a command. (See *Error reporting*.) The command is syntactically correct, but the Instrument cannot execute it, for example because a parameter is out of range.

Bit 3 (weight 8) — Device-dependent Error (DDE)

A device-dependent error is any device operation that did not execute properly because of some internal condition, for instance error queue overflow. This bit shows that the error was not a command, query or execution error.

Bit 2 (weight 4) — Query Error (QYE)

The output queue control detects query errors. For example the QYE bit shows the unterminated, interrupted, and deadlock conditions. For more details, see *Error Reporting*.

Bit 1 (weight 2)—Request Control (RQC)

Shows the controller that the device wants to become the active controller-in-charge. Not used in the Instrument.

Bit 0 (weight 1) — Operation Complete (OPC)

The Instrument only sets this bit TRUE in response to the operation complete command (*OPC). It shows that the Instrument has completed all previously started actions.

SCPI-defined Status Registers

The Instrument has two 16-bit SCPI-defined status structures, the operation status register and the questionable data register. These are 16 bits wide, while the status byte and the standard status groups are 8 bits wide.

Operation Status Group

Bit 5 (weight 32) — Waiting for Trigger

This bit is set true when:

- Sweep is active and waiting to start a sweep (for example from an :INIT command)
- Frequency Counter is active and in single-short mode and waiting for a trigger
- Power Meter is active and in single-short mode and waiting for a trigger

Note

This bit is only used in the above functions. General Power Meter measurements such as leveling do not set this bit.

Bit 4 (weight 16) — Measurement In Progress

This bit shows that the Instrument is measuring when:

- Frequency Counter is active and measuring
- Power Meter is active and measuring

Note

This bit is only used in the above functions. General Power Meter measurements such as leveling do not set this bit.

Bit 3 (weight 8) — Sweep In Progress

This bit shows that the Instrument is sweeping. It is set when the sweep has been triggered. For internally triggered sweeps, it is set at the same time as the Waiting for trigger bit.

Bit 1 (weight 2) — Settling

This bit shows that the Instrument is settling. It is set when there has been a change to the hardware that will have disturbed the output.

Summary, Operation Status Reporting

:STAT:OPER:ENAB

Enable reporting of Operation Status in the status byte.

*SRE 128

Enable SRQ when operation status has something to report.

:STAT:OPER?

Reading and clearing the event register of the Operation Status Register structure

:STAT:OPER:COND?

Reading the condition register of the Operation Status Register structure.

Questionable Data/Signal Status Group

The Questionable Data Status reports when the output data from the Instrument may not be trusted.

Bit 12 (weight 4096) — Frequency Outside Profile Range

This bit shows that the current frequency is outside the range of frequencies in the profile data file. Thus, the correction that is being calculated from this profile data may not be as the user expects.

Bit 11 (weight 2048) — External AM or FM Overload

This bit shows that the external AM or FM signal that is being applied to the Instrument is too large.

Bit 10 (weight 1024) — Head Serial Number Mis-match

This bit is set when the head that is plugged in (50 Ω or 75 Ω) to the Instrument was not calibrated with this base unit.

Bit 9 (weight 512) — External Reference Frequency Unlocked

The Instrument sets this bit true when it has lost lock to the externally applied reference library.

Bit 3 (weight 8) Frequency

The Instrument sets this bit true when it has lost internal frequency lock.

Bit 0 (weight 1) Voltage

The Instrument sets this bit true when it cannot level the output voltage.

Power-on Status Clear

Power-on clears all event enable registers and the service request enable register if the power-on status clear flag is set TRUE (see the common command *PSC.)

Preset the Status Reporting Structure

You can preset the complete status structure to a known state with a single command, the STATus:PRESet command, which does the following:

- Disables all bits in the Standard Event Register, the Operation Status Register, and the Questionable Data Register
- Enables all bits in Device Register 0

Leaves the Service Request Enable Register unaffected.

SCPI Commands

⚠ Caution

This Product contains relays that have a long, but finite lifespan. When programming the instrument from the IEEE Bus take care to not constantly exercise them.

To maximize the lifespan of the relays, observe the following good-practice points when controlling the Instrument from the bus:

- Minimize the number of output operate and standby transitions (OUTP ON | OFF) that are sent to the instrument.
- Minimize the number of resets (*RST) sent to the instrument when the output is on.
- Avoid repeatedly changing between functions (Sine, AM, FM etc.) with the output on.
- Group similar setup points (level and frequency) together rather than send sequences of disparate setup points.

This section documents the SCPI (Standard Commands for Programmable Instruments) Command Set for the Instrument. The commands are presented in a series of tables that are organized by functional subsystems, power, AM, FM, Sweep, etc. Also included are the common commands and the Status Registers. Each table represents a functional grouping and is preceded by an identifying second-order heading. Formal table headings and introductory paragraphs have been omitted for clarity.

Definition of Common Parameter Forms

Parameter Form	Definition
<bool>	Boolean data, which is ON or OFF, but allows numeric values also (zero is interpreted as OFF, and any non-zero value as ON).
<CPD>	Character Program Data: Select a parameter name from a listed group.
<string>	String program data type (enclosed in double quotes).
<NRf>	Numeric representation format: Number can be expressed as an integer (e.g. 123), real number (e.g. 123.4) or an exponent (e.g. 1.234E6).
n/a	Not applicable

INSTrument Subsystem

Keyword	Parameter Form	*RST Condition	Notes
:INSTrument			Command long form
:INST:CATalog?		n/a	Query only command that returns a comma-separated list of strings which contains the names of all logical instruments: SINE,SWEEP,AM,FM,PM,FCO,PMETer
:INST:CAT:FULL?		n/a	Query only command that returns a list of string - number pairs. The string contains the name of the logical instrument. The immediately following NR1-formatted number is its associated logical instrument number: SINE1, SWEEP2,AM3,FM4,PM5,FCO6,PMET7
:INST:NSELect[?]	<NRf>	1	This command is used in conjunction with the SELect command. It serves the same purpose, except that it uses a numeric value instead of the identifier used in the SELect command. When queried it shall return the logical instrument number. Changing the selected instrument will put the output into standby. Note that the query version of this command can report the selected functions as zero (0) when the instrument is in a state such as calibration or self-test.
:INST[:SELect][?]	<CPD>	SINE	This command selects the instrument as the default. When a logical instrument is selected, all other logical instruments are unavailable for programming until selected. The selections are: SINE, SWEep, AM, FM, PM, FCOunter, PMETer The query returns the string name of the currently selected instrument. Changing the selected instrument will put the output into standby. Note that the query version of this command can report the selected functions as “NONE” when the instrument is in a state such as calibration or self-test.

OUTPut Subsystem

Keyword	Parameter Form	*RST Condition	Notes
:OUTPut			Command long form
:OUTP[:STATe][?]	<bool>	0	The STATe command controls whether the output terminals are open or closed. When the state is OFF (0), the terminals are at maximum isolation from the signal.
:OUTP:MODE[?]	<SOURce|MEASure>	SOURce	[1]Selects the Source/Measure mode to show signal generation settings and power sensor readouts. Only applies to sine.
:OUTP:ROUTe[?]	<CPD>{HEAD|MICRowave}	HEAD	[1]The ROUTe command controls whether the signal is delivered via the leveling head or via the microwave output.
:OUTP:ROSCillator[:STATe][?]	<bool>	Unchanged	The STATe command controls whether the reference frequency is output on the rear panel BNC.
:OUTP:ROSCillator:FREQuency[?]	<NRf>	Unchanged	Selects the output reference frequency on the rear panel BNC, in Hz.
:OUTP:FITTed?	<SPD>,<SPD>	n/a	Query only command that returns two strings, the first is the leveling head model type and the second is the serial number. If no leveling head is fitted then the query will return
:OUTP:LEVeling[?]	<CPD>{NONE, S1, S2}	Unchanged	[1]Sets the leveling sensor to be either NONE, S1, S2, where S1 is the left inserted power sensor. S2 is the right inserted power sensor. NONE disables sensor/splitter leveling.
:OUTP:LEVeling:LIMit[?]	<NRf>	Unchanged	[1]Maximum actual output allowed when using sensor/splitter leveling.

:OUTP:LEVeling:CAPTure[?]	<NRf>	Unchanged	[1]Maximum feedback error allowed when using sensor/splitter leveling.
[1] Command(s) not available on the 96040A.			

INPut Subsystem

Keyword [1]	Parameter Form	*RST Condition	Notes
:INPut			Command long form
:INP:REAR[?]	<CPD>{DISable| LEVel | PULL}	DISable	Selects the input mode of the rear BNC connector. Automatically set as counter input when frequency counter mode selected.
:INP:LEVel:FSV[?]	<NRf>	Unchanged	Selects the external leveling Full Scale Voltage.
:INP:LEVel:FSP[?]	<NRf>	Unchanged	Selects the external leveling Full Scale Power.
:INP:LEVel:RTIMe[?]	<CPD>{SLOW| FAST}	SLOW	Selects the external leveling filter speed.
:INP:LEVel:CLAMp[?]	<NRf>	Unchanged	
:INP:FREQuency :POLarity[?]	<CPD>{POSitive |NEGative }<NRf>	POSitive	Selects the direction of pull a positive voltage change has on the frequency.
:INP:FREQuency:GAIN	<NRf>	Unchanged	Selects the frequency pull gain value.
[1] These command nodes are only valid when the SINE instrument is selected.			

POWer Subsystem

Keyword	Parameter Form	*RST Condition	Notes
[:SOURce]:POWer			Command long form
[:SOUR]:POW[:LEVel][:IMMediate][:AMPLitude][?]	<NRf>	-10.0 dBm	This selects the power level of the output for the current instrument that is selected.
[:SOUR]:POW:OFFSet[?]	<NRf>	0.0 dBm	Command not available in the sweep instrument. This value is an offset that is added to the output value.
[:SOUR]:POW:OFFS:STATe[?]	<bool>	0	Command not available in the sweep instrument. This selects whether the offset mode is present.
[:SOUR]:POW:OFFS:APPLy[?]	<bool>	0	Command not available in the sweep instrument. This selects whether the offset value is added to the output power. Note: The offset state must be on for this command to operate.
[:SOUR]:POW:OFFS:ERRor[?]	<NRf>	0.0 %	Command not available in the sweep instrument. Adjusts the output by setting the offset as an error rather than an absolute.

FREQuency Subsystem

Keyword	Parameter Form	*RST Condition	Notes
[:SOURce]:FREQuency			Command long form
[:SOUR]:FREQ:[CW|FIXed][?]	<NRf>	1.0MHz	Command not available in the sweep instrument. This selects the frequency of the output for the current instrument that is selected.
[:SOUR]:FREQ:RESolution[?]	<CPD> {STANdard| ENHanced}	STANdard	This selects the frequency setting resolution in Sine mode
[:SOUR]:FREQ:OFFSet[?]	<NRf>	0.0 Hz	Command not available in the sweep instrument. This value is added to the output value.
[:SOUR]:FREQ:OFFS:STATe[?]	<bool>	0	Command not available in the sweep instrument. This selects whether the offset mode is present
[:SOUR]:FREQ:OFFS:APPLy[?]	<bool>	0	Command not available in the sweep instrument. This selects whether the offset value is added to the output frequency. Note: The offset state must be on for this command to operate
[:SOUR]:FREQ:OFFS:ERRor[?]	<NRf>	0.0 %	Command not available in the sweep instrument. Adjusts the output by setting the offset as an error rather than an absolute.
[:SOUR]:FREQ:CENTer[?]	<NRf>	1.0MHz	Command only available in the sweep instrument. Sets the center frequency for a frequency sweep.
[:SOUR]:FREQ:SPAN[?]	<NRf>	9.0 MHz	Command only available in the sweep instrument. Sets the span for a frequency sweep.
[:SOUR]:FREQ:STARt[?]	<NRf>	1.0MHz	Command only available in the sweep instrument. Sets the start center frequency for a frequency sweep.
[:SOUR]:FREQ:STOP[?]	<NRf>	10.0MHz	Command only available in the sweep instrument. Sets the stop center frequency for a frequency sweep.

AM Subsystem

Keyword	Parameter Form	*RST Condition	Notes
[:SOURce]:AM			Command long form This command node is only available when the AM instrument is selected.
[:SOUR]:AM:STATe[?]	<bool>	0	This selects whether the output signal has an AM component
[:SOUR]:AM[:DEPTh][?]	<NRf>	30.0 %	This selects the depth of the AM for the AM instrument
[:SOUR]:AM:DEPTh:OFFSet[?]	<NRf>	0.0 %	This value is added to the depth of the output value. Changes to this value will be reflected in the AM:OFFSet:ERRor value
[:SOUR]:AM:DEPTh:OFFS:STATe [?]	<bool>	0	This selects whether the offset mode is present
[:SOUR]:AM:DEPTh:OFFS:APPLy [?]	<bool>	0	This selects whether the offset value is added to the output depth. Note: The offset state must be on for this command to operate
[:SOUR]:AM:DEPTh:OFFS:ERRor[?]	<NRf>	0.0 %	Adjusts the output depth by setting the offset as an error rather than an absolute.
[:SOUR]:AM:INTernal:FREQuency[?]	<NRf>	1000 Hz	This selects the modulation rate frequency of the AM
[:SOUR]:AM:INT:FREQ:OFFSet[?]	<NRf>	0.0 Hz	This value is added to the modulation frequency of the output value. Changes to this value will be reflected in the AM:INT:FREQ:OFFSet:ERRor value
[:SOUR]:AM:INT:FREQ:OFFS:STATe[?]	<bool>	0	This selects whether the offset mode is present
[:SOUR]:AM:INT:FREQ:OFFS:APPLy[?]	<bool>	0	This selects whether the offset value is added to the output modulation frequency. Note: The offset state must be on for this command to operate
[:SOUR]:AM:INT:FREQ:OFFS:ERRor[?]	<NRf>	0.0 %	Adjusts the output modulation frequency by setting the offset as an error rather than an absolute.
[:SOUR]:AM:SHAPe[?]	<CPD> {SINE | TRIangle | EXTernal}	SINE	This selects the shape of the modulation of the AM.
[:SOUR]:AM:COUPling[?]	<CPD> AC| DC}	AC	This selects the type of coupling for the AM.
[:SOUR]:AM:EXTernal :TRIGger[?]	<CPD> {DISable | RISing | FALLing}	DISable	Selects the type of external trigger for AM

FM Subsystem

Keyword	Parameter Form	*RST Condition	Notes
[:SOURce]:FM			Command long form This command node is only available when the FM instrument is selected.
[:SOUR]:FM:STATe[?]	<bool>	0	This selects whether the output signal has an FM component
[:SOUR]:FM[:DEViation][?]	<NRf>	10.0 kHz	This selects the deviation of the FM for the FM instrument.
[:SOUR]:FM:DEViation :OFFSet[?]	<NRf>	0.0 Hz	This value is added to the deviation of the output value. Changes to this value will be reflected in the FM:OFFSet:ERRor value
[:SOUR]:FM:DEViation:OFFS :STATe [?]	<bool>	0	This selects whether the offset mode is present
[:SOUR]:FM:DEViation:OFFS :APPLy [?]	<bool>	0	This selects whether the offset value is added to the output deviation. Note: The offset state must be on for this command to operate
[:SOUR]:FM:DEViation:OFFS :ERRor[?]	<NRf>	0.0 %	Adjusts the output depth by setting the offset as an error rather than an absolute.
[:SOUR]:FM:INTernal :FREQuency[?]	<NRf>	1.0 kHz	This selects the modulation rate frequency of the FM
[:SOUR]:FM:INT:FREQ :OFFSet[?]	<NRf>	0.0 Hz	This value is added to the modulation frequency of the output value. Changes to this value will be reflected in the FM:INT:FREQ:OFFSet:ERRor value
[:SOUR]:FM:INT:FREQ:OFFS :STATe[?]	<bool>	0	This selects whether the offset mode is present
[:SOUR]:FM:INT:FREQ:OFFS :APPLy[?]	<bool>	0	This selects whether the offset value is added to the output modulation frequency. Note: The offset state must be on for this command to operate
[:SOUR]:FM:INT:FREQ:OFFS :ERRor[?]	<NRf>	0.0 %	Adjusts the output depth by setting the offset as an error rather than an absolute.
[:SOUR]:FM:SHAPe[?]	<CPD> SINE | EXTernal}	SINE	This selects the shape of the modulation of the FM.
[:SOUR]:FM:COUPling[?]	<CPD> {AC| DC}	AC	This selects the type of coupling for the FM.
[:SOUR]:FM:EXTernal :TRIGger[?]	<CPD> {DISable |RISing |FALLing}	DISable	Selects the type of external trigger for FM

PM Subsystem

Keyword	Parameter Form	*RST Condition	Notes
[:SOURce]:PM			Command long form This command node is only available when the PM instrument is selected.
[:SOUR]:PM:STATe[?]	<bool>	0	This selects whether the output signal has an PM component
[:SOUR]:PM[:DEViation][?]	<NRf>	0.0001 Rad	This selects the deviation in Radians of the PM for the PM instrument.
[:SOUR]:PM:DEViation:OFFSet[?]	<NRf>	0.0 Rad	This value is added to the deviation of the output value. Changes to this value will be reflected in the PM:OFFSet:ERRor value
[:SOUR]:PM:DEViation:OFFS :STATe[?]	<bool>	0	This selects whether the offset mode is present
[:SOUR]:PM:DEViation:OFFS :APPLy[?]	<bool>	0	This selects whether the offset value is added to the output deviation. Note: The offset state must be on for this command to operate
[:SOUR]:PM:DEViation:OFFS :ERRor[?]	<NRf>	0.0 %	Adjusts the output depth by setting the offset as an error rather than an absolute.
[:SOUR]:PM:INTernal :FREQuency[?]	<NRf>	1.0 kHz	This selects the modulation rate frequency of the PM
[:SOUR]:PM:INT:FREQ :OFFSet[?]	<NRf>	0.0 Hz	This value is added to the modulation frequency of the output value. Changes to this value will be reflected in the PM:INT:FREQ:OFFSet:ERRor value
[:SOUR]:PM:INT:FREQ:OFFS :STATe[?]	<bool>	0	This selects whether the offset mode is present
[:SOUR]:PM:INT:FREQ:OFFS :APPLy[?]	<bool>	0	This selects whether the offset value is added to the output modulation frequency. Note: The offset state must be on for this command to operate
[:SOUR]:PM:INT:FREQ:OFFS :ERRor[?]	<NRf>	0.0 %	Adjusts the output depth by setting the offset as an error rather than an absolute.
[:SOUR]:PM:SHAPe[?]	<CPD> {SINE | EXTernal}	SINE	This selects the shape of the modulation of the PM.
[:SOUR]:PM:COUPling[?]	<CPD> {AC| DC}	AC	This selects the type of coupling for the PM.
[:SOUR]:PM:EXTernal :TRIGger[?]	<CPD> {DISable |RISing |FALLing}	DISable	Selects the type of external trigger for PM

SWEep Subsystem

Keyword	Parameter Form	*RST Condition	Notes
:SWEep			Command long form This command node is only available when the SWEEP instrument is selected.
:SWE:STATe?		n/a	Query only, this returns the current state of the sweep: STOP, ARM, RUN or PAUS
:SWE:TIME?		n/a	Query only, this returns the duration of the sweep
:SWE:DWELl[?]	<NRf>	100 ms	Controls the amount of time spent at each point during a sweep.
:SWE:SPACing[?]	<CPD> {LINear | LOGarithmic}	LIN	Selects which type of sweep is performed.
:SWE:STEP[?]	<NRf>	1.0 kHz	Selects the frequency for each step of the sweep.
:SWE:SQUelch[?]	<bool>	0	Select / deselect Squelch during transitions.
:SWE:SHAPe[?]	<CPD> {SAWTooth| TRIangle}	SAWTooth	Selects the method of how the sweep is performed. In the case of SAWTooth, when the sweep reaches the end, it will return directly to the beginning. In the case of TRIangle, it will traverse sweep back to the start.
:SWE:PROGress?		n/a	Query only to return how far through the sweep is. This will report 0.0 if the sweep has never started, and 100% if it has completed.
:SWE:ACTion	<CPD> {PAUSe | CONTinue}	n/a	This will pause and continue a sweep that is in progress. A settings conflict will be reported if the sweep is not in progress. There is no query form.

PMETer Subsystem (96270A)

Keyword	Parameter Form	*RST	Notes
:PMETer[<n>][1]			Where <n> is either 1 or 2 to select power sensor 1 or 2. If there is no <n>, then power sensor 1 is used by default.
:READ?		n/a	Returns the next available reading for sensor <n>. If sensor <n> is in single-trigger mode, this command triggers the reading and returns its value when complete.
:FETCh?		n/a	Returns the last reading taken of sensor <n>. If no reading has been taken, then it returns 9.91E+37
:IDN?		Unchanged	Query to return the make, model, Serial No, FW version. If the sensor is not fitted, the return will be NONE, NONE, NONE, NONE
:FREQuency[?]	<nrf>	1E6	Frequency at which to measure power
:REFerence[?]	<nrf>	Unchanged	Reference level to use for relative measurements
:STATe[?]	<bool>	OFF	Enables/disables relative measurements
:AVERage[?]	<nrf>	AUTO	Sets the number of readings taken in the averaging calculation. AUTO,1,2,4,8,16,32,64,128,256,512,1024,2048, 4096,8192,16384,32768
:TRIGger[?]		CONT	<SINGle | CONTinous>
:ZERO?		n/a	Perform input zero on sensor. Return '0' for successful zero operation; '1' to indicate a fail, with an error message put in the error queue.
[1] Command(s) not available on the 96040A			

Frequency Counter Subsystem (96270A)

Keyword	Parameter Form	*RST	Notes
:FCOunter:RANGe[?]	<nrf>	1	Selects the frequency counter range. 1 = 50 MHz @ 10 kΩ 2 = 50 MHz @ 50 Ω 3= 300 MHz @ 50 Ω

Trigger Subsystem

Keyword [1]	Parameter Form	*RST Condition	Notes
ABORt			This command is provided for aborting triggered action. On this instrument it is specifically used to stop a sweep or frequency counter measurement.
:INITiate[:IMMediate]		n/a	This is used in Sweep Function to initiate a sweep. A setting conflict will be reported if the TRIGger[:SOURce] is set to external. For power meter or frequency counter function, if the Single Event mode (INIT:CONT 0) is set, then this command triggers one reading to be taken. If in repetitive mode (INIT:CONT 1), then the command is ignored and measurements are free running
:INIT:CONTinuous[?]	<bool>	0	For either sweep, power meter or frequency counter this command determines whether there is a single event or repetitive sweep or measurement.
:TRIG[:SEQuence] :TYPE[?]	<CPD> {DISable | INPut | OUTput}	DISable	Selects the action of the rear panel Sweep Trigger BNC. Not applicable in the frequency counter or power meter functions.
:TRIG[:SEQuence] :SLOPe[?]	<CPD> {POSitive | NEGative}	POSitive	Determines if the sweep will be started with a positive (rising) or negative (falling) edge signal. Not applicable in the frequency counter or power meter functions.
:TRIG:GATE[?]	<NRf> {0.2 | 2 | 20 | 80}	2.0	Determines the gate time for the frequency counter. Note that the <NRf> will be rounded to the closest parameter allowed . Not applicable in the power meter function.

Keyword [1]	Parameter Form	*RST Condition	Notes
MEASure?	None	n/a	This command returns a measurement from the frequency counter or power meter function, If the mode is set to CONTinous, then the current measurement will be completed before its value is returned. If it is in SINGle, then a measurement will be initiated and completed before its value is returned. If a measurement function is not selected, then this command will return 9.91E37.
FETCh?		n/a	In Frequency Counter mode, immediately returns the last frequency measurement. If none have been taken (for example no trigger received), the response is the value 9.91E37 If not in frequency counter, will immediately return the last power meter reading taken. If no power sensors are fitted, command will return 9.91E+37.
*TRG	n/a	n/a	In Single Event mode (INIT:CONT 0) this command triggers the frequency counter or power meter. If the counter is in repetitive mode (INIT:CONT 1), the command is ignored.
IEEE 488.1 command GET (Group Execute Trigger)	n/a	n/a	If the instrument is in single shot frequency counter or power meter mode, this command triggers a measurement. Otherwise it is silently ignored. Multiple GETs in the frequency counter cause the measurement to be restarted. If GETs are received at a rate faster than the read rate, the reading will never complete. Multiple GETs in the power meter do no cause the measurement to be restarted. GETs received at a rate faster than the read rate are discarded.
[1] These command nodes are for use when the SWEEP, or Frequency Counter instrument is selected.			

Profile Subsystem (96270A)

Keyword	Parameter Form	*RST	Notes
:PROFile[2]			
:MEASure			
:SENSor[?]	<CPD> S1D |S2D | S1L | S2L}	S1D	Selects the reference plane and power sensor configuration for the self-characterization process. S1D is reference plane at sensor 1, output direct. S2D is reference plane at sensor 2, output direct. S1L is reference plane at sensor 1, levelled by sensor 2. S2L is reference plane at sensor 1, levelled by sensor 1.
:STARt[?]	<NRf>	1.0E6	The start frequency
:STOP[?]	<NRf>	10.0E6	The stop frequency
:LEVel[?]	<NRf>	1.0	Amplitude to measure at
:POINts[?]	<NRf>	n/a	Selects the number of points
:BEGin?	<String1>,<string2>	n/a	This starts the automatic characterization and saves the results in the filename <string1> with the comment <string2>
:SELect[?]	<string>	None	[2] Selects the active correction file.
:STATe[?]	<bool>	OFF	Enables/disables the selected correction file
:INTerpolate[?]	<CPD> {LINear | SMOoth}	Unchanged	Determines the interpolation method between points.
:VALue?			Returns the current correction value that is being applied
:CATalog?			Returns a <string> format response of all of the names in the stored profile filenames, comma separated
:EXISts?	<string >		Query to see if the filename <string> is present in the catalogue. Returns a 0 if it is not present, and a 1 if it is
:DELete?	<string>		[1] Deletes the file named as <string>
:IMPort			
:CLEar?		n/a	Empties the import buffer. This must be done before starting a

			new import. The command always returns a 0.
:ADD	<NRf>,<NRf>		Adds the data pair <frequency>, <level> into the import buffer.
:SAVE?	<String1>,<string2>		Saves the import buffer into the filename <string1> with the comment <string2>. An error is reported if this file already exists and the file not written.
:EXPort			
:NAME?			Queries the currently selected profile name and associated comment. Returns two comma separated strings. The first is the filename, the second is the comment from that file. Filename is a maximum of 8 characters, comment is a maximum of 200 characters.
:POINts?			Queries the number of data pairs in the selected file
:GET?	<NRf>		Get point number <nrf> data pair from selected data file.

[1] Returns 0 for success, 1 for failure, with a further error message put in the queue.

[2] Command(s) not available on the 96040A.

REFerence Subsystem

Keyword [1]	Parameter Form	*RST Condition	Notes
[:SOURce]:REFerence			Command long form This command node is only valid when the SINE instrument is selected.
[:SOUR]:REF[:STATe][?]	<bool>	0	This selects the Reference output mode. The values of the references will available at the output in place of the existing frequency and/or power
:[:SOUR]:REF:FREQuency?		1.0 MHz	Query only that will return the value of the frequency reference.
:[:SOUR]:REF:POWer?		-10.0 dBm	Query only that will return the value of the power reference.
[:SOUR]:REF:TRACk[?]	<CPD> {FREQuency | POWer | NONE}	NONE	This selects whether the reference signal parameters track the frequency or power on the main sine instrument
[:SOUR]:REF:COPY			This copies the current values from the main sine parameter to the reference values. There is no query form
[:SOUR]:REF:CONFirm[?]	<CPD> {DISable | ALWays | ABSolute | INCrease}	DISable	This command is used to determine if an additional OUTP ON command is needed before the reference values are transferred to the output terminal. This is to ensure that the reference parameters, which may be very different to the main sine instrument parameters is not mistakenly output, potentially damaging the unit under test.
[:SOUR]:REF:CONFirm:ABSolute[?]	NRf	-10 dBm	Sets the threshold at which the additional OUTP:ON is needed when switching to the reference output.
[:SOUR]:REF:CONFirm:INCrease[?]	NRf	-10 dBm	Sets the increase of signal at which the additional OUTP:ON is needed when switching to the reference output.

[1] These command nodes are only valid when the SINE instrument is selected.

UNIT Subsystem

Keyword	Parameter Form	*RST Condition	Notes
:UNIT			Command long form
:UNIT:POWer[?]	<CPD> {DBM | W | dBuV | VRMS | VPP}	DBM	This command sets the units of all power commands of the currently selected instrument.
:UNIT:POW:OFFSet[?]	<CPD> {DB | W | VRMS | VPP | PCT | PPM}	DB	This command sets the units of all power offset commands of the currently selected instrument.
UNIT:POW:OFFS:ERRor[?]	<CPD>{DB | PCT, PPM}	PCT	This command sets the units of all offset error commands of the currently selected instrument.
UNIT:FREQuency:OFFSet[?]	<CPD>{HZ, PCT, PPM, PPB}	Hz	This command sets the units of all frequency offset commands of the currently selected instrument. PCT is for percent, PPM is for parts per million, PPB is for parts per billion, only applicable in the Sine instrument.
UNIT:FREQ:OFFS:ERRor[?]	<CPD>{PCT, PPM, PPB}	PCT	This command sets the units of all frequency offset error commands of the currently selected instrument. PCT is for percent, PPM is for parts per million, PPB is for parts per billion, only applicable in the Sine instrument.
UNIT:AM:DEPT:OFFS:ERRor[?]	<CPD>{PCT, PPM}	PCT	This command sets the units of all depth offset error commands of the currently selected instrument. PCT is for percent, PPM is for parts per million
UNIT:AM:INT:FREQ:OFFS[?]	<CPD>{HZ, PCT, PPM}	Hz	This command sets the units of the rate of the currently selected instrument. PCT is for percent, PPM is for parts per million
UNIT:AM:INT:FREQ:OFFS :ERRor[?]	<CPD>{PCT, PPM}	PCT	This command sets the units of all rate offset error commands of the currently selected instrument. PCT is for percent, PPM is for parts per million
UNIT:FM:DEV:OFFS[?]	<CPD>{HZ, PCT, PPM}	Hz	This command sets the units of the deviation commands of the currently selected instrument. PCT is for percent, PPM is for parts per million
UNIT:FM:DEV:OFFS:ERRor[?]	<CPD>{PCT, PPM}	PCT	This command sets the units of all deviation offset error commands of the currently selected instrument. PCT is for percent, PPM is for parts per million

Keyword	Parameter Form	*RST Condition	Notes
UNIT:FM:INT:FREQ:OFFS[?]	<CPD> {HZ, PCT, PPM}	Hz	This command sets the units of the rate of the currently selected instrument. PCT is for percent, PPM is for parts per million
UNIT:FM:INT:FREQ:OFFS :ERRor[?]	<CPD> {PCT, PPM}	PCT	This command sets the units of all rate offset error commands of the currently selected instrument. PCT is for percent, PPM is for parts per million
UNIT:PM:DEV:OFFS[?]	<CPD> {RADian, PCT, PPM}	RAD	This command sets the units of the deviation commands of the currently selected instrument. PCT is for percent, PPM is for parts per million
UNIT:PM:DEV:OFFS:ERRor[?]	<CPD>{PCT, PPM}	PCT	This command sets the units of all deviation offset error commands of the currently selected instrument. PCT is for percent, PPM is for parts per million
UNIT:PM:DEV:OFFS:ERRor[?]	<CPD>{PCT, PPM}	PCT	This command sets the units of all deviation offset error commands of the currently selected instrument. PCT is for percent, PPM is for parts per million
UNIT:PM:INT:FREQ:OFFS:ERRor[?]	<CPD>{PCT, PPM}	PCT	This command sets the units of all rate offset error commands of the currently selected instrument. PCT is for percent, PPM is for parts per million
:UNIT:SWEep:STEP[?]	<CPD> {SPS | SPD | HZ | PPM | PCT}	Hz	Sets the units for the sweep step size. SPS is for Steps Per Sweep SPD if for Steps Per Decade.
:UNIT:SWEep:PROGress[?]	<CPD> {PCT | RANGe}	PCT	Select Sweep Progress Units.
:UNIT:OCLamp[?]	<CPD>{DBM | W | dBuV | VRMS | VPP}	DBM	This command sets the units for the external leveling output clamp.
:UNIT:PMETer<n>:POWer[?]	<CPD> {DBM | W | VRMS | VPP | DBUV}	DMB	Sets the units of the power sensor <n>

ROSCillator Subsystem

Keyword	Parameter Form	*RST Condition	Notes
[:SOURce]:ROSCillator			Command long form
[:SOUR]:ROSC:SOURce[?]	<CPD> {INTernal | EXTernal ENARow}	Unchanged	Selects the source of reference frequency EXTernal requires a reference within ±0.3 ppm The ENARow parameter is equivalence to the EXTernal selection. This has been retained for backward compatibility with the 9640A instrument.
[:SOUR]:ROSC:EXTernal:FREQuency[?]	<NRf>	Unchanged	Sets the external Frequency in Hz.
[:SOUR]:ROSC:LOCKed?		n/a	Query returns 1 if frequency is locked to either the internal or external [:SOURce].

SYSTem Subsystem

Keyword	Parameter Form	*RST Condition	Notes
SYSTem			Command long form
SYSTem:ERRor?		n/a	Query only. Returns instrument error string or 0 if no error.
SYST:VERSion?		n/a	Query only. Returns SCPI version to which instrument complies.
SYST:MSTRike?		n/a	Query only. Returns a string of 20 ASCII ‘1’s or ‘0’s. The characters define the modifications (Mod Strike) that have been applied to this particular instrument. Each character is comma separated.

STATus Subsystem

Keyword	Parameter Form	*RST Condition	Notes
:STATus			Command long form
:STAT:OPER[:EVENt]?		n/a	Query only. Returns the contents of the Operation Event Register.
:STAT:OPER:ENABle[?]	NRf	0	Sets the mask for the Operation Event Register.
:STAT:OPER:CONDition?		n/a	Query only. Returns the contents of the Operation Condition Register.
:STAT:QUES[:EVENt]?		n/a	Query only. Returns the contents of the Questionable Event Register.
:STAT:QUES:ENABle[?]	NRf	0	Sets the mask for the Questionable Event Register.
:STAT:QUES:CONDition?		n/a	Query only. Returns the contents of the Questionable Condition Register.
STAT:PRESent		n/a	Sets Registers to a SCPI defined state.

CALibration Subsystem

Keyword	Parameter Form	*RST Condition	Notes
CALibration		n/a	Command long form
:CAL:SECure:PASSword	<string>		Enables Calibration Mode using a password.
:CAL:SECure:EXIT			Exit Calibration Mode.
:CAL:TARGet	<NRf>,<NRf>,<NRf>		First parameter is level, second is frequency and the third specifies which calibration point the first two parameters apply to.
:CAL:ACTual[?]			Changes the output value for adjustment
:CAL:TRIGger?			Accept the adjustment, return 0=success, 1 =otherwise
:CAL:PRIMary			Sub-command
:CAL:PRIM:FADJust?	<NRf>		Frequency adjust. Return 0 for success, 1 for failure

Common Commands

Keyword	Parameter Form	Notes
*CLS		The *CLS common command clears the status data structures by clearing all event registers and the error queue. It does not clear enable registers and transition filters. It clears any pending *WAI, *OPC, and *OPC?.
*ESE	NR1	Sets the enable bits of the standard event enable register. This enable register contains a mask value for the bits to be enabled in the standard event status register. A bit that is set true in the enable register enables the corresponding bit in the status register. An enabled bit will set the ESB (Event Status Bit) in the Status Byte Register if the enabled event occurs.
*ESR?		Reads out the contents of the standard event status register. Reading the Standard Event Status Register clears the register.
*IDN?		Reads out the manufacturer, model, serial number, Firmware level for main and GPIB program in an ASCII response data element. Response is <Manufacturer> , <Model> , <Serial Number>, <Firmware Level>. Eg: Fluke,96270A,123456,1.00 or Fluke,96040A,54321,1.00
*OPC		The Operation Complete command causes the device to set the operation complete bit in the Standard Event Status Register when all pending selected device operations have been finished. (Currently only AM settings have significant delays.)
*OPC?		Operation Complete query. The Operation Complete query places an ASCii character 1 into the device's Output Queue when all pending selected device operations have been finished. (Currently only AM settings have significant delays.)
*OPT?		Responds with a string of ten comma-separated characters, with '1' representing fitted and '0' representing not fitted. For example: 1,1,1,0,0,0,0,0,0,0 1st - character (always set to 1) represents the 8662/3 emulation.[1] 2nd – character (always set to 1) represents the frequency counter.[1] 3rd - character represents the low level microwave output attenuator. 4th - 10th characters are currently unused.
*PSC	NR1	Enables/disables automatic power-on clearing. The status registers listed below are cleared when the power-on status clear flag is 1. Power-on does not affect the registers when the flag is 0.
*RST		The Reset command resets the instrument. It is the third level of reset in a 3-level reset strategy, and it primarily affects the instruments functions, not the IEEE 488 bus.
*SRE	NR1	The Service Request Enable command sets the service request enable register bits. This enable register contains a mask value for the bits to be enabled in the status byte register. A bit that is set true in the enable register enables the corresponding bit in the status byte register to generate a Service Request.
*STB?		Reads out the value of the Status Byte. Bit 6 reports the Master Summary Status bit (MSS), not the Request Service (RQS). The MSS is set if the instrument has one or more reasons for requesting service.
*TRG		This triggers the instrument when it is in the trigger state (INIT: CONT 0). In this case the command is used to start a frequency counter measurement.
*TST?		The self-test query causes an internal self-test and generates a response indicating whether or not the device completed the self-test without any detected errors.
*WAI		The Wait-to-Continue command prevents the device from executing any further commands or queries until execution of all previous commands or queries have been completed.
[1] Standard feature on 96000, included in OPT? response for 9640A compatibility.		

SCPI Status Registers

Operation Status Register

Bit	2n	Label	Comment
0	1	Calibrating	Not used, Always zero
1	2	Settling	Hardware is settling
2	4	Ranging	Not used, Always zero
3	8	Sweeping	A sweep is in progress
4	16	Measuring	A measurement is in progress
5	32	Waiting for Trig	Waiting for a sweep trigger
6	64	Waiting for Arm	Not used, Always zero
7	128	Correcting	Not used, Always zero
8	256	Unassigned	Not used, Always zero
9	512	Unassigned	Not used, Always zero
10	1024	Unassigned	Not used, Always zero
11	2048	Unassigned	Not used, Always zero
12	4096	Unassigned	Not used, Always zero
13	8192	Instrument Summary	Not used, Always zero
14	16384	Program Summary	Not used, Always zero
15	32768	Not Used	Not used, Always zero

Questionable Status Register

Bit	2n	Label	Comment
0	1	Voltage	The voltage output is no longer levelled
1	2	Current	Not used, Always zero
2	4	Time	Not used, Always zero
3	8	Frequency[1]	The frequency is no longer locked
4	16	Not Used	A measurement is in progress
5	32	Not Used	Not used, Always zero
6	64	Not Used	Not used, Always zero
7	128	Not Used	Not used, Always zero
8	256	Characterization	Factory use only
9	512	External Ref Frequency unlocked	Unable to lock to the externally supplied frequency
10	1024	Head Serial Number mismatch	The currently fitted head serial number was not calibrated by this base unit
11	2048	External AM or FM overload	This external signal is too large
12	4096	Outside Profile[2]	Current frequency is outside the profile data range
13	8192	Unassigned	Not used, Always zero
14	16384	Command warning	Not used, Always zero
15	32768	Not Used	

[1] 1999 SCPI Syntax and Style specification labels this bit as “Power”. The Instrument uses this bit as “Frequency”.

[2] Not used on 96040A.

Coupled Commands

What Is Command Coupling?

Commands from the IEEE interface bus are usually executed serially in the order they are received. However, because commands may come in any order in a command string, it is possible that a combination of commands produce an illegal machine state if executed in isolation, but a valid machine sate if executed collectively.

This problem is overcome by defining a coupling between commands which allows the execution of individual components to be deferred until all contiguous coupled commands in the same group have been parsed and the validity of the combinations checked.

Note

Individual commands may be a member of several coupled command groups.

A good example is power and frequency. Either of these commands could be used individually to configure an instrument (with the other parameter assumed or defaulted). However, there are instances when both commands are required together before the requested configuration is valid.

Suppose the instrument has a profile that allows high frequency at low power and high power at low frequency. Assume the instrument is currently set to a high frequency, low power and we require a change to give high power, low frequency. Manually, we would have to reduce the frequency before we could increase the power.

On the bus, if the power command is sent before the frequency command and the commands were processed as they were received, then an error would be reported as the instrument would think that a high power AND a high frequency were being requested. See Figure .

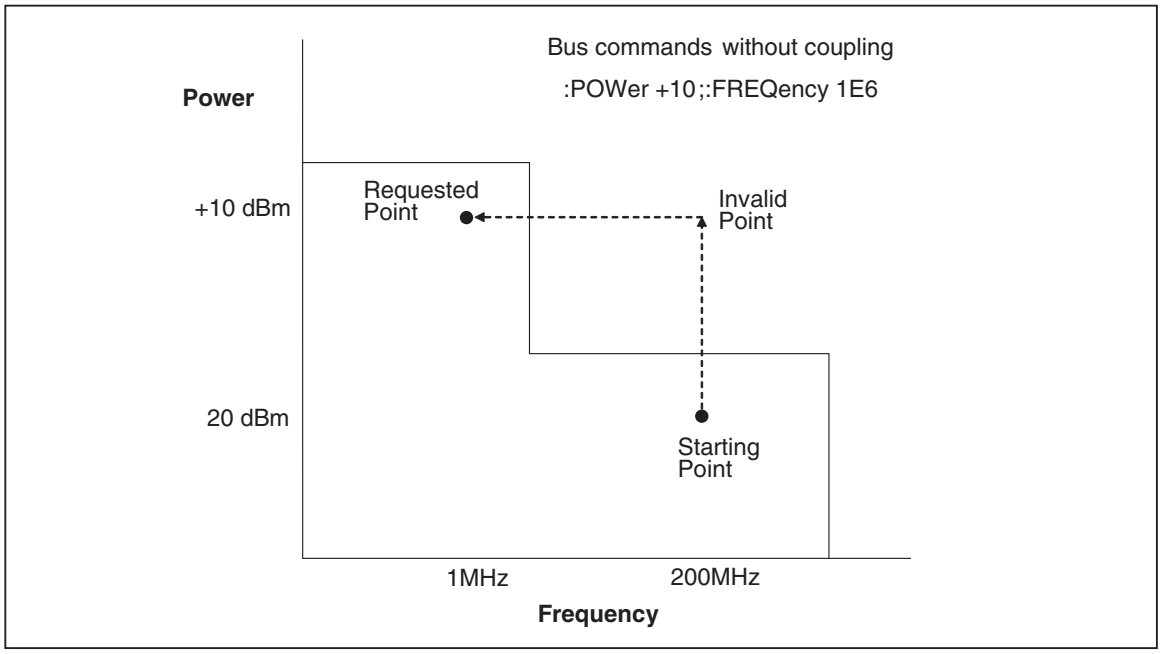


Figure 16. Bus Command without Coupling

Coupling overcomes this by deferring the processing of commands until all related items are gathered together allowing them to be processed at once. In Figure , the Instrument knows that frequency and power are inter-dependent, and that executing power then frequency would be illegal, so it executes the frequency command first, then the power command to successfully get to the point requested.

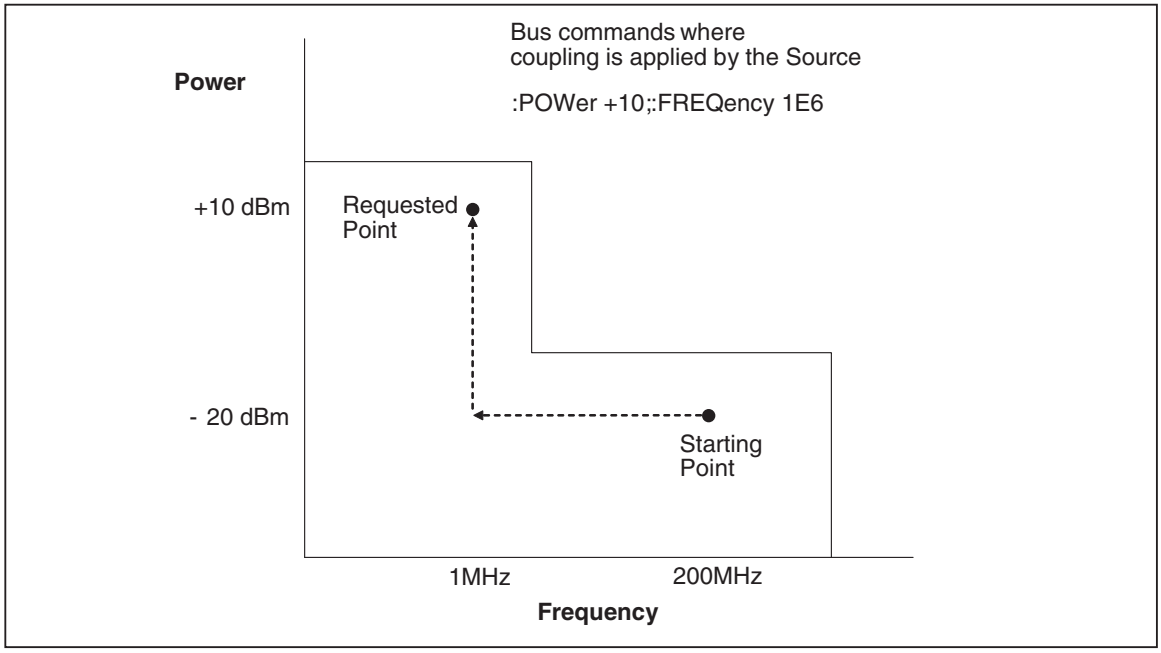


Figure 17. Bus Command with Coupling

Coupled Command List

Table provides a list of Coupled Commands and identifies which commands are coupled. An **x** in a column indicates a coupled row. For example, column 3 has an **x** in the row for :FREQuency:CENTer and :FREQuency:SPAN, indicating these commands are coupled.

Table 6. List of Coupled Commands

COMMAND	"x" in a column indicates a coupled row													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
:FREQuency												x	x	x
:POWer												x	x	x
:FM:DEViation													x	
:FM:INT:FREQ													x	
:FM:SHAPe													x	
:PM:DEViation														x
:PM:INT:FREQ														x
:PM:INT:SHAPe														x
:AM:INT:FREQ												x		
:AM:SHAPe												x		
:AM:DEPTh												x		
:FREQuency:CENTer											x			
:FREQuency:SPAN											x			
:POWer:OFFSet								x						
:POWer:OFFSet:STATe								x						
:POWer:OFFSet:APPLy								x						
:POWer:OFFSet:ERRor								x						
:FREQ:OFFSet							x							
:FREQ:OFFSet:STATe							x							
:FREQ:OFFSet:APPLy							x							
:FREQ:OFFSet:ERRor							x							
:AM:DEPTh:OFFSet						x								
:AM:DEPTh:OFFSet:STATe						x								
:AM:DEPTh:OFFSet:APPLy						x								
:AM:DEPTh:OFFSet:ERRor						x								
:AM:INT:FREQ:OFFSet					x									
:AM:INT:FREQ:OFFSet:STATe					x									
:AM:INT:FREQ:OFFSet:APPLy					x									
:AM:INT:FREQ:OFFSet:ERRor					x									

Table 6. List of Coupled Commands (cont.)

COMMAND	"x" in a column indicates a coupled row													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
:FM:DEV:OFFSet				x										
:FM:DEV:OFFSet:STATe				x										
:FM:DEV:OFFSet:APPLy				x										
:FM:DEV:OFFSet:ERRor				x										
:FM:INT:FREQ:OFFSet			x											
:FM:INT:FREQ:OFFSet:STATe			x											
:FM:INT:FREQ:OFFSet:APPLy			x											
:FM:INT:FREQ:OFFSet:ERRor			x											
:PM:DEV:OFFSet		x												
:PM:DEV:OFFSet:STATe		x												
:PM:DEV:OFFSet:APPLy		x												
:PM:DEV:OFFSet:ERRor		x												
:PM:INT:FREQ:OFFSet	x													
:PM:INT:FREQ:OFFSet:STATe	x													
:PM:INT:FREQ:OFFSet:APPLy	x													
:PM:INT:FREQ:OFFSet:ERRor	x													

Programming Examples

Remote Programming Examples

This section gives some examples of the commands needed to set up various programming scenarios for the Instrument. The examples use a variety of short and long forms of the commands, a variety of upper and lower case, and a variety of ways of representing parameters (e.g. 1E6 or 1000000 are the same).

Leveled Sine Output

Requirement: To output a 1.1 MHz, -14.2 dBm sine wave.

```
*RST
*CLS
INST SINE
UNIT:POWer DBM
POWER -14.2
FREQ 1.1E6
OUTPUT ON
```

To find out what the current output signal is, in V rms.

```
UNIT:POW VRMS
POWER?
```

<- Instrument responds with the value 4.360000000000E-02

AM Output

Requirement: To output a 500 kHz, -5.0 dBm carrier wave with a 2 kHz, 15 % depth modulation. This example uses the full long-form of the power command.

```
INST AM
UNIT:POWer DBM
:SOURCE:POWER:LEVEL:IMMEDIATE:AMPLITUDE -5.0
FREQ 500000.0
AM:INTernal:FREQ 2.0E+3
AM:DEPTh 15
AM:STATe 1
OUTPut ON
```

Requirement: Remove the modulation from the above signal to output just the carrier wave:

```
AM:STATE off
```

FM Output

Requirement:

To output a 430 MHz, 4.556 dBm carrier wave with a 27 kHz, 500 kHz deviation modulation. This example uses tree walking to set up the modulation:

```
INST FM
UNIT:POWer DBM
POW -4.556
FREQ 430e6
FM:STATE 1;DEV 500.0E+3;INTernal:FREQ 27E3
OUTPUT ON
```

Sweep Output

Requirement: To perform a single sweep from 1 MHz to 10 MHz in 1 MHz steps with 133 ms between each step at 1V rms.

```
INST SWEEP
FREQ:START 1E6;:FREQ:STOP 10000000
UNIT:POWER VRMS
:POWER 1.0
:SWE:DWELL 0.133
SWEEP:STEP 1E6
:INIT:CONT OFF
OUTP ON
:INIT
```

Requirement: To perform a repetitive logarithmic sweep of 15 points over 20 MHz, centered around 100 MHz with a dwell of 1 second between each step, started by an external trigger.

```
INST SWEEP
FREQ:CENT 100E6
FREQ:SPAN 20E6
UNIT:POWER VRMS
:POWER 1.0
SWE:SPAC LOG
:SWE:DWELL 1.0
:UNIT:SWEEP:STEP SPD
SWEEP:STEP 15
OUTP ON
TRIG:SOURCE EXT
```

<- The sweep will only begin when the there is a trigger signal on the external rear-panel connector.

Leveled Sine Output with Offset

Requirement: To output a 1.1 MHz, -14.2 dBm sine wave. Then to offset the output power by +0.1dbm:

```
*RST
*CLS
INST SINE
UNIT:POWer DBM
POWER -14.2
FREQ 1.1E6
OUTPUT ON
POWER:OFFSET:STATE 1
POWER:OFFSET 0.1
```

Requirement: To find out what the UUT error is in the above scenario.

```
POWER:OFFSET:ERROR?
```

<- Instrument responds with the value -2.27600000000E+00

Frequency Measurement - simple

Requirement: To measure the system reference frequency. This requires that the signal to be measured is applied to the rear Frequency Pull ad Counter Input BNC.

```
INST FCO
TRIGGER:GATE 20
MEAS?
```

Frequency Measurement - triggered

Requirement: To measure the system reference frequency at a specific trigger point. This requires that the signal to be measured is applied to the rear Frequency Pull ad Counter Input BNC.

```
INST FCO
TRIGGER:GATE 20
INIT:CONT OFF
   < delay until reading required >
INIT;*WAI
FETCH?
```

Operation Status Register

Requirement: To perform a single sweep from 1 MHz to 10 MHz in 1 MHz steps with 133 ms between each step at 1V rms. Monitor the Operational Status bit that indicates that the sweep is in progress.

```
INST SWEEP
FREQ:START 1E6;:FREQ:STOP 10000000
UNIT:POWER VRMS
:POWER 1.0
:SWE:DWELL 0.133
SWEEP:STEP 1E6
:INIT:CONT OFF
OUTP ON
:INIT
STATUS:OPER:COND?
```

<- Instrument responds with a value that has bit 4 set, i.e., the value 8 wait for 5 seconds

```
STATUS:OPER:COND?
```

<- Instrument responds with a value that has bit 0 clear, for example, the value 0

SRQ Operation and Error Handling

Requirement: To generate a service request from the Instrument when it detects a problem.

```
*RST
*CLS
INST SINE
*SRE 255
*ESE 255
UNIT:POWer DBM
POW 1          <- this command would be executed by the Instrument
POW 1E6        <- The user meant to set up 1 MHz, with FREQ 1E6
                  <- Instrument generates an SRQ

*STB?
```

<- Instrument responds with the value 32 (decimal). This indicates there is Event flag

```
*ESR?
```

<- the Event Register returns 15 (decimal) meaning there is an execution error.

```
SYST:ERR?
```

<- Instrument returns the message from the error queue

<- -222”Data out of range;Value too large”

<- indicating the problem with the last command

HP 3335A Command Emulation

This section describes the 3335A emulation mode. When in this mode, the Instrument responds to 3335A IEEE bus commands instead of the SCPI bus commands.

The 3335A command set has a limited number of functions compared to the Instrument. As a result, there are many features of the Instrument that are not available under emulation mode.

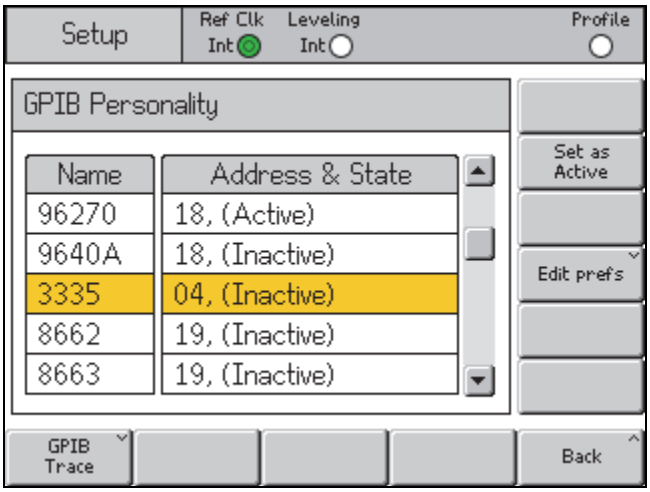
Note

The Instrument does not store the state of the last 3335A bus setting. Therefore, if the user manually switches the Instrument from remote to local, makes a setting change, and then switches back to remote, the Instrument may not be in the state the controlling computer expects when it resumes control.

Prepare the Instrument for Remote 3335A Emulation

To prepare the Instrument for 3335A emulation:

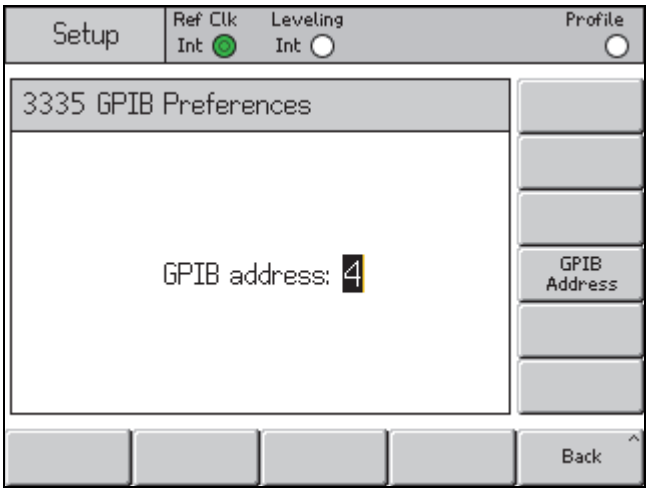
1. Push SETUP on the front panel.
2. Push the GPIB Preferences softkey to show the GPIB Personality screen.
3. Use the ⏶⏷keys to select the 3335 personality. See Figure 18.



ead347f.bmp

Figure 18. GPIB Personality Screen

4. Push the Edit Pref's softkey to bring up the 3335 GPIB Preferences screen. See Figure .



ead348f.bmp

Figure 19. 3335 GPIB Preferences Screen

5. Set the GPIB Address using either the Spin Wheel or ⏶⏷keys.
6. Push the Exit softkey to return to the GPIB Personality screen.

Commands that are Emulated

Table lists the commands to which the Instrument responds.

Table 7. Emulated Commands

Command	3335A Code	Comment
Frequency	F	Sets the frequency
Frequency Increment	I	Sets the frequency increment value
Amplitude	A	Sets the Amplitude
Amplitude Increment	I	Sets the Amplitude Increment value
Sweep Width	W	Sets the sweep width
0-9	0-9	Used for numeric entry
Backspace	B	This backspaces the characters received over the bus.
MHz/-dBm	M	Units for numeric entry
KHz/+dBm	K	Units for numeric entry
Hz/deg	H	Units for numeric entry
Increment up	U	Increment the currently active parameter (F or A)
Increment down	D	Decrement the currently active parameter (F or A)
Go to start freq	G	Go to the start freq of a sweep
Start 10 second single	X	Start a single sweep lasting 10 seconds, **Information:** The 3335A performs 1000 steps during this sweep. The 96270A performs 500 steps during this command. See below about changing sweep modes.
Start 50 second single	Y	Start a single sweep lasting 50 seconds, 1000 steps per sweep. See below about changing sweep modes.
Start Auto	Z	Start a repetitive sweep: 8 sweeps/sec, 100 steps/sweep **Information:** the 96xxx performs 7 steps per sweep during this command.
Stop	Q	Stop the sweep
Negative symbol	-	For entering negative values

Commands Not Emulated

Table lists the commands which are silently ignored by the Instrument.

Table 8. Commands not Emulated

Command	3335A Code	Comment
Store	S	Stores the current setup in one of 0-9 slots
Recall	R	Restores a setup from one of 0-9 slots
Phase increment	P	Sets the phase Increment value
Display Last Entry	L	Used to display the last entry so that it can be edited
Clear	C	Stops the special PAD attenuator mode
PAD selection	T	Selects 1 of 7 attenuators to give a specific level output

Other Differences in Emulation Mode

Table identifies differences between the HP3335A and the Instrument operating in emulation mode.

Table 9. Emulation Differences

HP3335A	96xxx
Sweep timing can be changed between X mode (10 second sweep) and Y mode (50 second sweep), and the sweep continues without restart.	Does not emulate this behaviour because it needs to calculate the sweep parameters before the sweep starts.
Has a Sweep Output Connector to provide a 0 to +2 volts sweep ramp for driving external equipment.	Does not have this feature.
Has a front panel switch to select 50 Ω or 75 Ω output. There is also a bus command to do this. The instrument takes 1.76 dBm off the 50 Ω output.	Requires a different head to be manually inserted to produce correctly levelled 75 Ω signals, 6.4 dBm down from the 50 Ω output.
Provides phase continuous frequency sweep.	Provides phase continuous sweep for output frequencies below 15 MHz, but at frequencies above 15 MHz hardware ranging will cause phase discontinuities in the output waveform.
Balanced 124 Ω / 135 Ω / 150 Ω output	The only outputs available are at 50 Ω and 75 Ω, from a precision N-series male connector.
Isolated from the GPIB bus by opto couplers, effectively isolating the instrument from the bus.	The GPIB ground is connected to earth ground and the RF signal common is floating.

HP 8662A/8663A Command Emulation

This section describes the 8662A and 8663A emulation mode. When in this mode, the Instrument responds to 8662/8663A IEEE bus commands instead of the SCPI bus commands. Also, only functions available on the Instrument are emulated. For example, the Instrument does not provide simultaneous AM and FM modulation like the 8662/8663A, so this feature is not emulated.

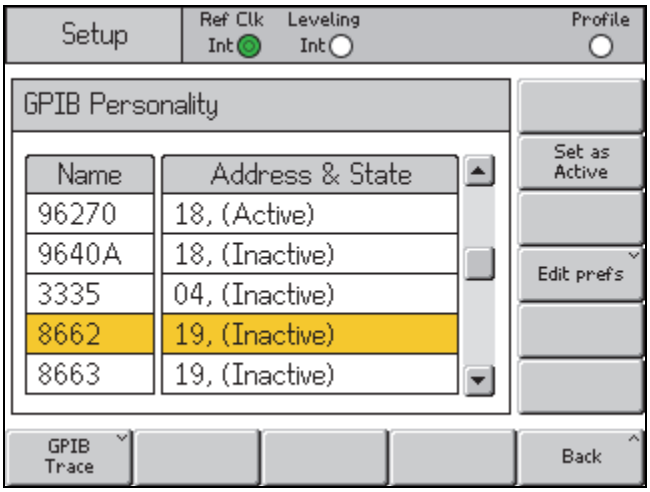
Note

The Instrument does not store the state of the last 8662A/8663A bus setting. Therefore, if the user manually switches the Instrument from remote to local, makes a setting change, and then switches back to remote, the Instrument may not be in the state the controlling computer expects when it resumes control.

Prepare the Instrument for Remote 8662/8663A Emulation

To prepare the Instrument for 8662A emulation:

1. Push SETUP on the front panel.
2. Push the GPIB Preferences softkey to show the GPIB Personality screen.
3. Use the ⏶⏷keys to select the 8662 personality. See Figure .



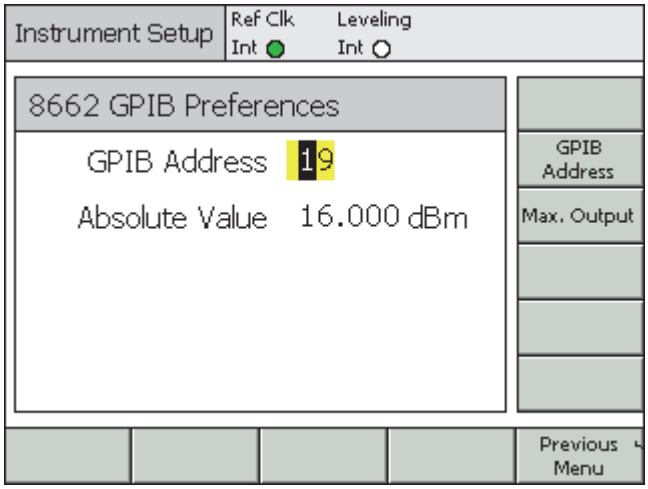
ead349f.bmp

Figure 20. GPIB Personality Screen

4. Push the Edit Pref's softkey to show the 8862 GPIB Preferences screen

5. Set the GPIB Address and the Max. Output amplitude.

   The output amplitude adjustment provides the opportunity to reduce (limit) the output power to match that of the 8662A/8663A, and avoid applying excessive power to the unit under test. See Figure .



ead350f.bmp

Figure 21. 8662 GPIB Preferences Screen

Note

The output threshold set in GPIB Preference has no effect in the local mode. Exceeding the threshold in local and then switching to remote can cause the GPIB Max. Output to be exceeded when control is returned to the controller.

Emulated Commands

Table lists the 8662A/8663A command set. In the Emulate column, a check mark (✓) indicates the Instrument has an equivalent operation and can emulate the command. An X indicates the Instrument has no equivalent operation and emulation is not available. The Instrument silently accepts and ignores commands that have no equivalent operation.

Table 10. Emulated 8662A/8663 Commands

Feature	8662/8663 Code	Emulate	Comment
Amplitude	AP	✓ ✓ ✓ ✓ ✓	Uses units of +D (for +dBm) -D (for -dBm) DM (for dBm) MV (for mV) UV (for μV)
	AO	✓	Amplitude Off , actually sets the output to -139.9dBm
Frequency	FR	✓ ✓ ✓ ✓	Uses units of GZ MZ KZ HZ
Sweep Frequency Start Stop	FA FB	✓ ✓ ✓ ✓	Uses units of GZ MZ KZ HZ
Sweep Frequency Span	FS	✓ ✓ ✓ ✓	Uses units of GZ MZ KZ HZ
Increment	xxISnnUU	✓	xx: Selects what function is to be incremented (e.g. frequency or amplitude) IS: is the increment command nn is the value UU is the Units
	UP DN	✓	Increment up or down the value set with the xxISnnUU command.

Table 10. Emulated 8662A/8663 Commands (cont.)

Knob	R1 R2 R3	X	Used to select which digit is active for the knob up/down command. *When under bus control, the 96xxx has no concept of cursor.*
	xxR4xx	X	Enables the knob setting when changing functions to xx (for example, frequency or amplitude).
	R5	X	Knob command that is not supported
	RU/RD RU/RD	X ✓	Increment / Decrement the knob for digit edit Single step in sweep mode.
Learn Mode	L1 L2	X X	A binary dump mode of instrument setup. *The 96xxx does not support this feature.*
AM	AMnnPC	✓	Sets up Amplitude Modulation nn is the depth, PC is for percent.
FM	FMffKZ	✓	Sets up Frequency Modulation ff is the frequency, KZ is for units of kHz.
PM	PMddDG	✓	Sets up Phase Modulation dd is the angle, DG is for units of degrees.
Pulse	PL	X	Pulse Modulation *The 96xxx does not have this functionality.*
Modulation	M0 M1 M2 M3 M4 MF	✓ ✓ ✓ ✓ ✓ ✓	Turns Mod off Internal 400 Hz Internal 1 kHz External AC External DC Set a modulation frequency
Sequence	AS SSrrrrrST BLSSrrrrrST	X X X	Recall of front panel storage registers in sequence. *The 9640 does not support this feature.*
Status	MS	✓X	This is the error message response command. See ‘Status Error Matching’ section.
Store/Recall	ST RC	X X	Save/recall. *The 96xxx does not support this feature.*
Sweep Modes	W1 W2 W3 W4	✓ ✓ ✓ ✓	Sweep off Auto Sweep Manual Sweep Single Sweep
Sweep Marker	X1 to X7	X	Signal generated from the rear of the 8662. *The 96xxx cannot generate this signal.*
Sweep Step	N1 N2 N3 N4 N5	✓ ✓ ✓ ✓ ✓	Sets Sweep step size and log or lin sweep.

Table 10. Emulated 8662A/8663 Commands (cont.)

Sweep Time (Dwell)	T1 T2 T3 T4 T5	X X X X ✓	Sweep dwell time. *Although all of these commands are accepted, T1 to T4 are all less than the minimum 20 ms that the 96xxx supports. In this case the minimum of 20 ms will be applied.*
Special commands			*Note:* *Only the following SP or BLAP commands are emulated. All other commands are ignored.*
	SP00	✓	Initialize Instrument
	SP81 BLAP81	✓ ✓	Convert V to dBm
	SP87 BLAP87	✓ ✓	8662A: HPIB operator Response, This sets a bit in the Service Request register, which, if enabled will generate and SRQ.
	SP89 BLAP89	✓	8663A: HPIB operator Response, This sets a bit in the Service Request register, which, if enabled will generate and SRQ.
Trigger Mode	CT	X	Configure Trigger. *The 96xxx does not support this feature.*
Trigger	TR GET	X X	*The 96xxx does not support this feature.*
Remote Sweep Step	Y0 Y1 Y2 Y3	✓ ✓ ✓ ✓	RSS Clear – deselects sweep mode RSS with display – this selects sweep mode RSS no display – this selects sweep mode Immediate Execution Mode – does a single sweep step
	@1	✓	Write Request Service Mask. Note that this is a byte value. i.e. single byte in the range 0-255.
	RM	✓	Responds with a single byte containing the value of the Request Service Mask
	@2 @3	✓ ✓	Deferred execution mode Immediate execution mode *Both of these commands are accepted and ignored.*

8662A/8663A Features Not Emulated

Table identifies differences between the 8662A/8663A and the Instrument operating in emulation mode.

Table 11. Emulation Differences

8662A/8663A	96xxx
The number ‘0’ and the characters ‘O’ and ‘o’ are interchangeable. As is the single quote ‘’’ and ‘@’.	No equivalent
There is a deferred mode (default) selected with the command @2 and an immediate mode, selected with @3 to change how commands are handled.	No equivalent
AM Depth can go to 0.0 %.	AM Depth can go to 0.1 %
There is an INT button on the rear of the instrument to select the internal frequency reference.	The equivalent control can be found under the general preferences setup.
There are two buttons and one BNC connector to select the externally supplied ref freq.	The equivalent of these can be found under the general preferences setup.
GET (Group Execute Trigger) is supported.	GET (Group Execute Trigger) is not supported.
Turns on with the same state that it was turned off.	The 96xxx always powers up in the same state.

Error Message Matching.

The status reporting command ‘MS’ format for the 8662A and 8663A are slightly different.

8662: EE,00,00,00,00,00,00,00,00,00,00,00,00,X0

8663: EE,000,X

Where EE is the error code

00 No Error
11 Fm Overmodulated
15 Am Overmodulated
32 Freq Out Of Range
33 Amplitude Over 16dbm
34 Amplitude Under 139x96dbm
35 Amplitude Am Over 10dbm
36 Amplitude Over 999mv
37 Am Over 95pc
40 Fm Deviation Error
43 Wrong Entry Protocol
45 Start Stop Freq Equal
49 Sweep Step Size Error
59 Sweep Span Out Of Limit
99 HW Malfunction

The X is set to 1 when the external modulation (AM or FM) is selected.

Request Service (RQS) Byte

Emulation mode attempts to recreate similar behavior of the SRQ and RQS feature of the 8662A/8663A. However, is not possible that the response timings will be the same, nor will the timing for clearing bits within the SRQ.

Table shows what bits are emulated by the Instrument.

Table 12. Emulated Bits

8	7	6	5	4	3	2	1
128	64	32	16	8	4	2	1
SP87	RQS	Sweep End	Param Changed	Power Fail	HW error	Entry Error	Ready

SP87	Sending this command causes an SRQ
Sweep End	When the last step of a sweep occurs
Param Change	When any parameter of the output changes
Power Fail	When the generator is returned to a power ON condition (from standby or off)
HW Error	When a hardware condition arises
Entry Error	When any invalid keystroke or program command occurs
	Ready When the generator is finished processing a Data message.

HP 836xx Command Emulation (96270A)

This section describes the 836xx emulation mode. When in this mode, the Instrument responds to 836xx IEEE SCPI commands. Also, only commands available on the Instrument are emulated. For example, the Instrument does not provide markers like the 836xx so this feature is not emulated.

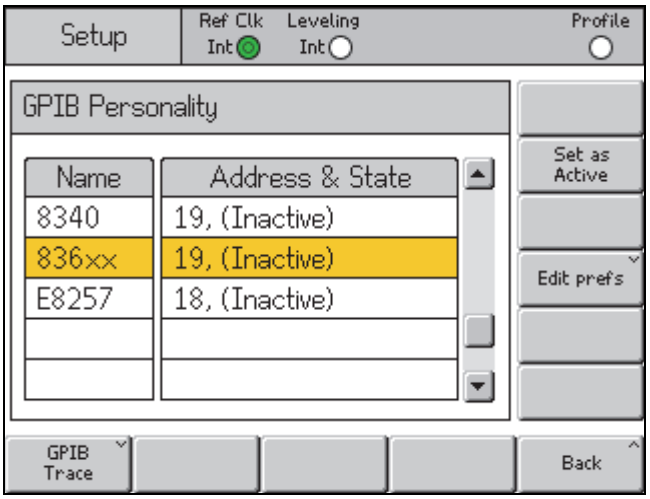
Note

The Instrument does not store the state of the last 836xx bus setting. If the Instrument is manual switched from remote to local, makes a setting change, and then switches back to remote, the Instrument may not be in the state the controlling computer expects when it resumes control.

Prepare the Instrument for Remote 836xx Emulation

To prepare the Instrument for 836xx emulation:

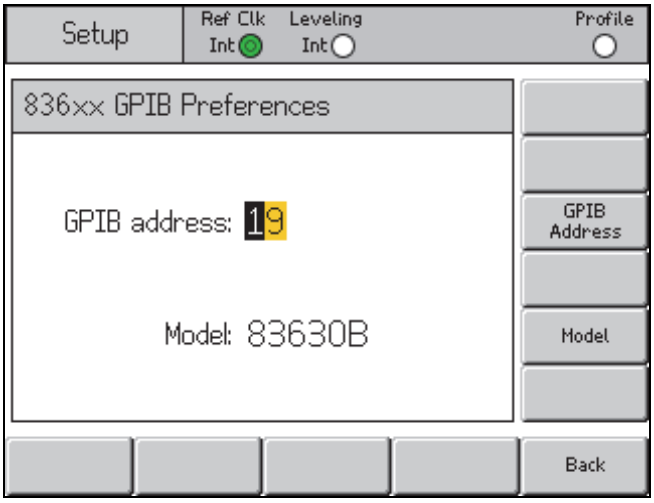
1. Push SETUP on the front panel.
2. Push the GPIB Preferences softkey to show the GPIB Personality screen.
3. Use ▲ and ▼ to select the 836xx personality. See Figure 22.



hpn15.bmp

Figure 22. GPIB Personality Screen

4. Push the Edit Pref's softkey to show the 836xx GPIB Preferences screen.
5. Set the GPIB Address and Model. See Figure .
6. Push the Model softkey and use the keypad to set the <model> part of the *IDN? response to the required 83630 series model number.
7. Push ENTER to save the edited model number.



hpn16.bmp

Figure 23. 836xx GPIB Preference Screen

Emulated Commands

Table lists the 836xx command set. In the Emulate column, a check mark (✓) indicates the Instrument has an equivalent operation and can emulate the command. An X indicates the Instrument has no equivalent operation and emulation is not available. The Instrument silently accepts and ignores commands that have no equivalent operation. See Table .

Table 13. 836xx Command Set

Command	Parameters	Parameter Type1	Allowed Values	Emulate
:ABORt				✓
:AM				✓
:INTernal				✓
[:DEPth]	AM depth %	numeric	0 to 40 dB	✓
:FREQuency	AM frequency	extended numeric	<num>[freq suffix] or MAXimum | Minimum	✓
:FUNCtion	waveform	discrete	SINusoid | SQUare TRIangle | RAMP | NOISe	✓ X ✓ X
:MODE	AM depth	discrete	DEEP | NORMal	X
:SOURce	AM source	discrete	INTernal | EXTernal	✓
:STATe	state	Boolean	ON | OFF | 1 | 0	✓
:TYPE	AM type	discrete	LINear | EXPonential	X
:CALibration				
:AM				X
:AUTO	auto calibrate	Boolean	ON | OFF | 1 | 0	X
[:EXECute]				X
:PEAKing				X
:AUTO	auto RF peak	Boolean	ON | OFF | 1 | 0	X

			[:EXECute]	
:PMETer				X
:DETector				X
:INITiate?	type of det cal	discrete	IDETector | DIODe	X
:NEXT?	power correction value	extended numeric	<num> [lvl suffix]	X
:CALibration				X
:PMETer				X
:FLATness				X
:INITiate?		discrete	USER | DIODE | PMETer | MMHead	X
:NEXT?	measured power	extended numeric	<num> [lvl suffix]	X
:SPAN				X
:AUTO	auto calibrate state	Boolean	ON | OFF | 1 | 0	X
[:EXECute]				X
:TRACk				X

:CORRection				X
:ARRay[0 | 1]	1601 pts of correction	extended numeric	f<num>[DB]g1601*1601	X
:FLATness	801 freq-correction pairs	extended numeric	f<num>[freq su x],DBg2*801	X

:POINts?	num of freq-correction pairs	extended numeric	[MAXimum | MINimum]	X
:SOURce[0 | 1]	correction source	discrete	ARRay | FLATness	X
[:STATe]	state	Boolean	ON | OFF | 1 | 0	X

:DIAGnostics				X
:ABUS				X
:AVERage	ADC averages	extended numeric	<num>	X
:STATus?				X
:INSTrument				X
:PMETer				X
:ADDRess	power meter address	extended numeric	1 to 31	X
:PRINter				X
:ADDRess	printer address	extended numeric	1 to 31	X
:IORW	I/O device number and value	extended numeric	<num>, <num>	X
:OUTPut				X
:FAULts?				X
:RESult?				X

:DIAGnostics				X

:TEST				X
:CONTinue				X
:DATA				X
:DESC?				X
:MAXimum?				X
:MINimum?				X
:VALue?				X
:DISable	disable listed self-tests	extended numeric or discrete	f<num>g1*? | ALL	X
:ENABle	enable listed self-tests	extended numeric or discrete	f<num>g1*? | ALL	X
[:EXECute]		extended numeric	0 to 288	X
:LOG				X
:SOURce	log when	discrete	ALL | FAIL	X
[:STATe]	state	Boolean	ON | OFF | 1 | 0	X
:LOOP	state	Boolean	ON | OFF | 1 | 0	X
:NAME?	Self-test number	extended numeric	0 to 288	X
:POINts?	number of self-tests			X
:RESult?	condition of			X

	self-tests			
:TINT?				X

:DISPlay				X
[:STATe]	state	Boolean	ON | OFF | 1 | 0	X

:FM				✓
:COUPling	coupling type	discrete	AC | DC	✓
[:DEViation]	peak FM deviation	extended numeric	<num>[freq su x] or MAXimum | MINimum	✓
:FILTer				X
:HPASs	FM AC Bandwidth	extended numeric	<num>[freq su x] or MAXimum | MINimum	X
:INTernal				✓
:FREQuency	FM frequency	extended numeric	<num>[freq su x] or MAXimum | MINimum	✓
:FUNCtion	FM waveform	discrete	SINusoid | SQUare TRIangle | RAMP | NOISe	SINE
:SENSitivity		extended numeric	100KHZ/V | 1MHZ/V | 10MHZ/V or MAXimum | MINimum	X

:FM				✓
:SOURce	FM source	discrete	INTernal | EXTernal	✓
:STATe	state		ON | OFF | 1 | 0	✓
:FREQuency				✓

:CENTer	center freq	extended numeric	speci ed freq range or MAXimum | MINimum | UP | DOWN	✓
[:CW]	CW freq	extended numeric	speci ed freq range or MAXimum | MINimum | UP | DOWN	✓
:AUTO	coupled to center freq Boolean	ON | OFF | 1 | 0		X
:MANual	manual freq	extended numeric	start/stop limits or MAXimum | MINimum | UP | DOWN	X
:MODE	free mode	discrete	CW | SWEep | LIST	X
:MULTiplier	freq mult	extended numeric	+36 to 36 or MAXimum | MINimum	X
:STATe	state	Boolean	ON | OFF | 1 | 0	X
:OFFSet	freq o set	extended numeric	+99.999 to 99.999 GHz or MAXimum | MINimum	✓
:STATe	state	Boolean	ON | OFF | 1 | 0	✓
:SPAN	freq span	extended numeric	0 to MAX-MIN or MAXimum | MINimum | UP | DOWN	✓
:STARt	start freq	extended numeric	speci ed freq rangeor MAXimum | MINimum | UP | DOWN	✓
:STEP				✓
:AUTO	auto freq step	Boolean	ON | OFF | 1 | 0	X
[:INCRement] freq step	extended numeric	range or MAXimum | MINimum		✓
:STOP	stop freq	extended numeric	speci ed freq range or MAXimum | MINimum | UP | DOWN	✓

:INITiate				✓

				✓
:CONTinuous	cont sweep	Boolean	ON | OFF | 1 | 0	✓
[:IMMediate]	sweep immediately			✓

:LIST				X
:DWELl	dwell time	extended numeric	f0.1 to 3200 msg*801 or fMAXimum | MINimumg1*801	X
:POINts?				X
:FREQuency	list freq	extended numeric	fspeci ed freq rangeg*801 or fMAXimum | MINimumg1*801	X
:POINts	num of freq points	extended numeric	[MAXimum | MINimum]	X
:LIST				X
:MANual	num of points to lock on	numeric	1 to maximum dened	X
:MODE	list sweep mode	discrete	AUTO | MANual [:POWer]	X
:CORRection	correction level	extended numeric	f+40 to 40 DBg1*801 or fMAXimum | MINimumg1*801	X
:POINts? num of corr levels	numeric	[MAXimum | MINimum]		X
:TRIGger				X
:SOURce	list trig source	discrete	IMMediate | BUS | EXTernal	

:MARKer[n]	[n] is 1 to 5 (1 is the default)			X
:AMPLitude				X
[:STATe]	state	Boolean	ON | OFF | 1 | 0	X

:VALue	amp marker depth	extended numeric	+10 to 10DB | MAXimum | MINimum	X
:AOFF				X
:DELTa?	dierence between two markers	numeric		X
:FREQuency	marker frequency	extended numeric	speci ed freq rangeor MAXimum | MINimum	X
:MODE	marker mode	discrete	FREQuency | DELTa	X
:REFerence	delta marker ref	numeric	1 to 5	X
[:STATe]	state	Boolean	ON | OFF | 1 | 0	X

:MODulation				X
:OUTPut				X
:SOURce	output mod source	discrete	AM | FM	X
:STATe	output mod state	Boolean	ON | OFF | 1 | 0	X
:STATe?	state	Boolean	ON | OFF | 1 | 0	X

:POWer				✓
:ALC				
:BANDwidth	ALC bwidth	extended numeric	<num>[freq su x] or MAXimum | MINimum	X
:AUTO	bwidth selection	Boolean	ON | OFF | 1 | 0	X
:CFACtor	coupling factor	extended	0 to 90[DB] or MAXimum |	X

		numeric	MINimum | UP | DOWN	
:SOURce	leveling point	discrete	INTernal | DIODe | PMETer | MMHead	X
[:STATe]	state	Boolean	ON | OFF | 1 | 0	X

:POWer				
:AMPLi er				X
:STATE	Boolean	ON | OFF | 1 | 0		X
:AUTO	Boolean	ON | OFF | 1 | 0		X
:ATTenuation	atten setting	extended numeric	0 to 90 [DB] or MAXimum | MINimum | UP | DOWN	X
:AUTO	coupled atten	Boolean	ON | OFF | 1 | 0	X
:CENTer	power sweep center	extended numeric	speci ed power range or MAXimum | MINimum | UP | DOWN	X
[:LEVel]	output level	extended numeric	speci ed power range or MAXimum | MINimum | UP | DOWN	✓
:MODE	power mode	discrete	FIXed | SWEep	X
:OFFSet	power equation o set	extended numeric	<num>[level su x] or MAXimum | MINimum | UP | DOWN	X
:STATe	state	Boolean	ON | OFF | 1 | 0	X
:RANGe	power meter range	extended numeric	30 to 90DB or MAXimum | MINimum | UP | DOWN	X
:SEARch	search mode	Boolean	ON | OFF | 1 | 0 | ONCE	X

:SLOPe	power slope	extended numeric	2.5 to 2.5DB/GHZ or MAXimum | MINimum | UP | DOWN	X
:STATe	state	Boolean	ON | OFF | 1 | 0	X
:SPAN	power sweep span	extended numeric	+45 to 45DB or MAXimum | MINimum | UP | DOWN	X
:STARt	power sweep start value	extended numeric	speci ed power range or MAXimum | MINimum | UP | DOWN	X
:STATe	RF on/o	Boolean	ON | OFF | 1 | 0	X
:STEP				X
:AUTO	step size determined Boolean	ON | OFF | 1 | 0		X
[:INCRement]	step size	extended numeric	20 to 0.01DB or MAXimum | MINimum	X
:STOP	power sweep stop value	extended numeric	speci ed power range or MAXimum | MINimum | UP | DOWN	X
:PULSe				X
:FREQuency	pulse freq	extended numeric	<num>[freq su x] or MAXimum | MINimum	X
:PERiod	pulse period	extended numeric	<num>[time su x] or MAXimum | MINimum	X
:WIDTh	pulse width	extended numeric	<num>[time su x] or MAXimum | MINimum	X

:PULM				X
:EXTernal				X
:POLarity	extnl pulse polarity	discrete	NORMal | INVerted	X
:DELay	extnl pulse delay	extended numeric	<num>[time su x] or MAXimum | MINimum	X

:INTernal				X
:FREQuency intnl pulse frequency extended numeric	<num>[freq su x] or MAXimum | MINimum			X
:GATE	intnl pulse gating	Boolean	ON | OFF | 1 | 0	X
:PERiod	intnl pulse period	extended numeric	<num>[time su x] or MAXimum | MINimum	X
:TRIGger				X
:SOURce	pulse trigger source	discrete	INTernal | EXTernal	X
:WIDTh	intnl pulse width	extended numeric	<num>[time su x] or MAXimum | MINimum	X
:SLEW	pulse modulation	extended numeric	<num>[time su x]	X
or MAXimum | MINimum				
:AUTO	pulse mod rise time	Boolean	ON | OFF | 1 | 0	X
:SOURce	pulse mod source	discrete	INTernal | EXTernal | SCALar	X
:STATe	state	Boolean	ON | OFF | 1 | 0	X

:ROSCillator				✓
:SOURce	ref osc source	discrete	INTernal | EXTernal | NONE	✓
:AUTO	state	Boolean	ON | OFF | 1 | 0	X

:STATus				✓

:OPERation				✓
:CONDition?				✓
:ENABle	numeric	0 to 2047 [:EVENt]?		✓
:NTRansition neg transition lter	numeric	0 to 2047		X
:PTRansition pos transition lter	numeric	0 to 2047		X
:PRESet				✓
:STATus				
:QUEStionable				✓
:CONDition?				✓
:ENABle	SRQ enable register	numeric	0 to 2047	✓
[:EVENt]?				✓
:NTRansition neg transition lter	numeric	0 to 2047		X
:PTRansition pos transition lter	numeric	0 to 2047		X

:SWEep				✓
:CONTrol				X
:STATe	dual source mode	Boolean	ON | OFF | 1 | 0	X
:TYPE	type ofsweep control	discrete	MASTer | SLAVe	X
:DWELl	settling time plus dwell time	extended numeric	0.1 to 3200 ms or MAXimum | MINimum	✓
:AUTO	dwell calculation state	Boolean	ON | OFF | 1 | 0	X

:GENeration	type of sweep	discrete	STEPped | ANALog	X
:MANual				X
:POINt	step point number	numeric	1 to the number of step points	✓
[:RELative]	percent of sweep	extended numeric	0 to 100%	X
:MARKer				X
:STATe	state	Boolean	ON | OFF | 1 | 0	X
:XFER	M1=start, M2=stop			X
:MODE	manual sweep mode switch	discrete	AUTO | MANual	X
:POINts	points in step sweep numeric	<num> | MAXimum | MINimum		✓
:STEP	step size	extended numeric	function of current span MAXimum | MINimum	X
:TIME	sweep time	extended numeric	200s to 133 ms or MAXimum | MINimum	✓
:AUTO	auto sweep time switch	Boolean	ON | OFF | 1 | 0	X
:LLIMit	fastest sweep time	extended numeric	<num>[time su x] or MAXimum | MINimum	X
:TRIGger				X
:SOURce	stepped trig source	discrete	IMMediate | BUS | EXTernal	X

:SYSTem				
:ALTernate	save/recall register	numeric	1 to 8 | MAXimum | MINimum	X

:STATe	state	Boolean	ON | OFF | 1 | 0	X
:COMMunicate				X
:GPIB				X
:ADDRess instrument address	numeric	1 to	30	X
:DUMP				X
:PRINter?				X
:ERRor?				✓
:KEY				X
:ASSign	key code assign	numeric	0 to 511, 1 to 14 excluding 5 and 10	X
:CLEar	clears user menu	numeric	1 to 14 | ALL	X
:DISable	save lock	discrete	SAVE	X
:ENABle	save lock	discrete	SAVE	X
:LANGuage	language selection	discrete	SCPI | CIIL | COMPatible	SCPI
:MMHead				X
SELect	discrete	FRONt | REAR | NONE		X
AUTO	Boolean	ON | OFF | 0 | 1		X
:PRESet				X
[:EXEC]				X

:SAVE				X
:TYPE	preset mode	discrete	FACTory | USER	X
:SECurity				X
:COUnt	memory clear	numeric	0 to 32767 | MAXimum | MINimum	X
[:STATe]	state	Boolean	ON | OFF | 1 | 0	X
:VERSion?				✓

:TRIGger				✓
[:IMMediate]				✓
:ODELay	output delay	extended numeric	0 to 3.2s	X
:SOURce	trig source	discrete	IMMediate | BUS | EXTernal	✓
:TSWeep equivalent of :ABORt;INITiate[:IMMediate]				
:UNIT				✓
:AM	AM depth units	discrete	DB | PCT	X
:POWer	default power units	string	DBM	✓

Common Commands

Keyword	Emulated
*CLS	✓
*ESE	✓
*ESE?	✓
*ESR?	✓
*IDN?	✓
*OPC	✓
*OPC?	✓
*OPT?	✓
*PSC	✓
*RST	✓
*SRE	✓
*STB?	✓
*TRG	✓
*TST?	✓
*WAI	✓
*LRN?	X
*RCL	X
*SAV	X

HP 8340A Command Emulation (96270A)

This section describes the 8340A emulation mode. When in this mode, the Instrument responds to 8340A IEEE bus commands instead of the SCPI bus commands. Also, only functions available on the Instrument are emulated. For example, the Instrument does not provide markers like the 8340A, so this feature is not emulated.

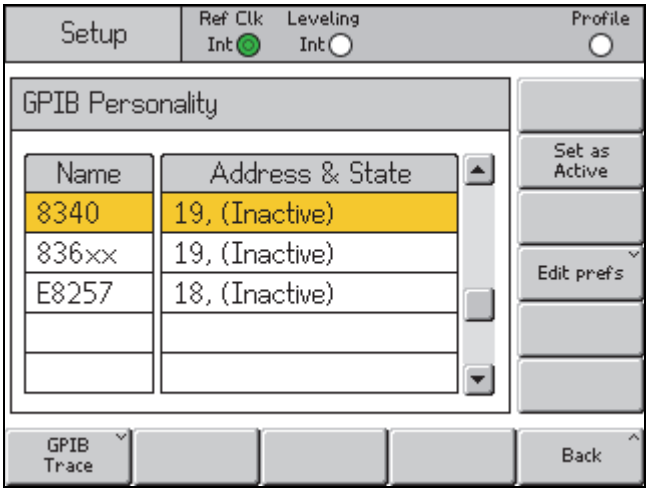
Note

The Instrument does not store the state of the last 8340A bus setting. Therefore, if you manually switches the Instrument from remote to local, makes a setting change, and then switches back to remote, the Instrument may not be in the state the controlling computer expects when it resumes control.

Prepare the Instrument for Remote 8340A Emulation

To prepare the Instrument for 8340A emulation:

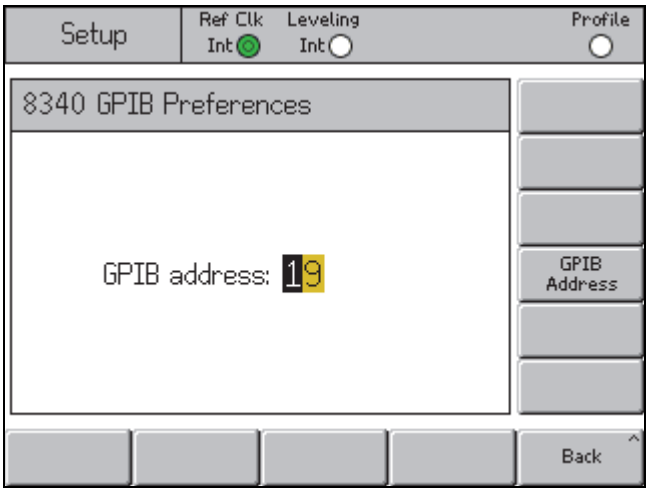
1. Push SETUP on the front panel.
2. Push the GPIB Preferences softkey to show the GPIB Personality screen.
3. Use the ⌃ and ⌄ to select the 8340 personality. See Figure .



hpn17.bmp

Figure 24. GPIB Personality Screen

4. Push the Edit Pref's softkey to show the 8340A GPIB Preferences screen.
5. Set the GPIB Address as necessary. See Figure .



hpn18.bmp

Figure 25. 8340A GPIB Preferences Screen

Emulated Commands

Tables and list the 8340A command set that indicates the Instrument has an equivalent operation and can emulate the command. A command that is not listed indicates the Instrument has no equivalent operation and emulation is not available. The Instrument silently accepts and ignores commands that have no equivalent operation.

Table 14. 8340A Emulated Commands

Command	8340A Code	Comment
CW	CW	Sets CW (10 MHz to 26.5 GHz)
CF	CF	Sets Center Frequency (10.00005 MHz to 26.49999995 GHz)
ΔF	DF	Sets Sweep Width Frequency (100 Hz to 26.49 GHz)
START	FA	Sets Sweep Start Frequency (10 MHz to 26 499.0000 MHz, Default: 10 MHz)
STOP	FB	Sets Sweep Stop Frequency (10.0001 MHz to 26.5 GHz, Default: 26.5 GHz)
Frequency Units	GZ MZ KZ HZ	Frequency in GHz Frequency in MHz Frequency in KHz Frequency in Hz
Power units	DB	Power units of dBm
Time Units	SC MS	Seconds Milli Seconds
CONTINUOUS	S1	Sets sweep mode to be continues
SINGLE	S2 or SG	Sets sweep mode to be single
FREE RUN	T1	Sets Sweep Trigger to be free run
EXTERNAL	T3	Sets Sweep Trigger to be external
Sweep Time	ST	Sets Sweep Time (10 milliseconds to 200 seconds, max is limited to 10 second by 96270)
AM ON	AM1	Sets amplitude modulation on
AM OFF	AM0	Sets amplitude modulation off
POWER LEVEL	PL	Sets output power level
RF ON	RF1	Sets RF power on
RF OFF	RF0	Sets RF power off
INTERNAL	A1	Sets internal power level
EXTERNAL CRYSTAL	A2	
EXTERNAL POWER METER	A3	Sets external power meter

Table 15. Special HP-IB Only Commands

Command	8340A Code	Comment
CLEAR BOTH STATUS BYTES	OS	Gets output status bytes (both bytes): 00 is reported.
STATUS BYTE MASK	RM	Sets output status (BYTE 1)
EXTENDED STATUS BYTE MASK	RE	Sets output extended status (BYTE 2)
CLEAR BOTH STATUS BYTES	CS	Clears both status bytes
OUTPUT FIRMWARE ID	OI	Gets firmware ID
OUTPUT INTERROGATED VALUE	OP	Gets interrogated parameter value (Implemented as OPPL, OPCW, OPCF, OPDF, OPFA, OPFB, OPST)
OUTPUT ACTIVE VALUE	OA	Gets active parameter value. If last active value is not available (Entry Display is blank), 0 is reported.
OUPUT POWER LEVEL	OR	Gets power level value
OUTPUT FAULT	OF	Gets output fault value: 0 is reported

All other 8340A GPIB commands will not be processed by the emulation mode.

Agilent E8257 Command Emulation (96270A)

This section describes the E8257 emulation mode. When in this mode, the Instrument responds to E8257 IEEE SCPI commands. Also, only commands available on the Instrument are emulated. For example, the Instrument does not provide markers like the E8257, so this feature is not emulated.

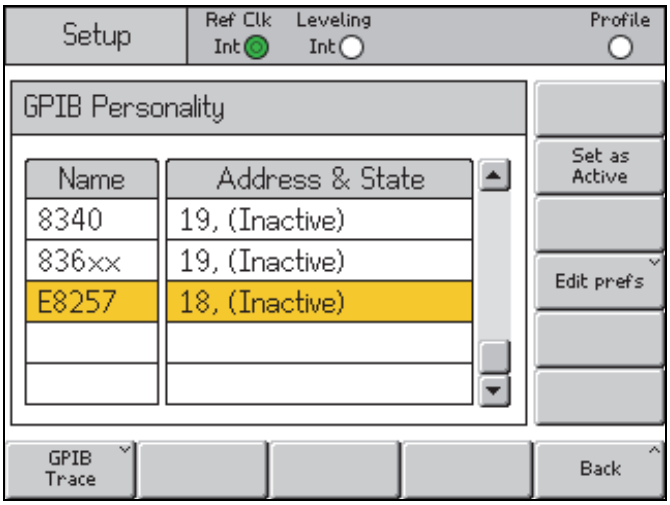
Note

The Instrument does not store the state of the last E8257 bus setting. Therefore, if the user manually switches the Instrument from remote to local, makes a setting change, and then switches back to remote, the Instrument may not be in the state the controlling computer expects when it resumes control.

Prepare the Instrument for Remote E8257 Emulation

To prepare the Instrument for E8257 emulation:

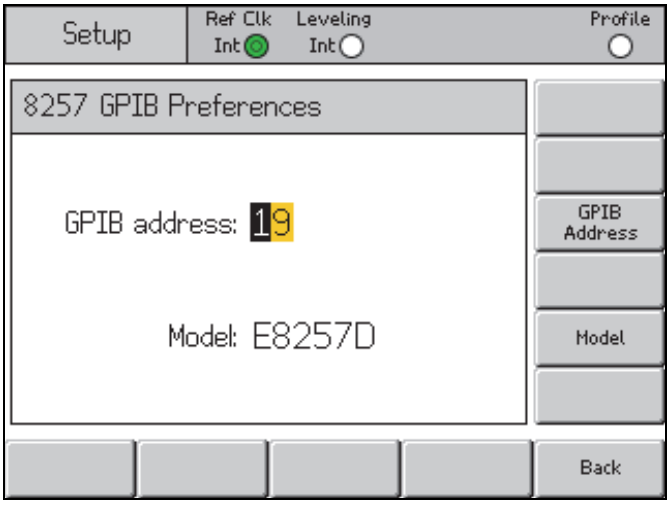
1. Push SETUP on the front panel.
2. Push the GPIB Preferences softkey to show the GPIB Personality screen.
3. Use ▲ and ▼ to select the E8257 personality. See Figure .



hpn19.bmp

Figure 26. GPIB Personality Screen

4. Push the Edit Pref's softkey to show the E8257 GPIB Preferences screen
5. Set the GPIB Address and the Model. See Figure .
6. Push the Model softkey and use the keypad to set the <model> part of the *IDN? response to the required E8257 series model number.
7. Push ENTER to save the edited model number.



hpn20.bmp

Figure 27. E8257 GPIB Preferences Screen

Emulated Commands

Table lists the E8257 command set. In the Emulate column, a check mark (✓) indicates the Instrument has an equivalent operation and can emulate the command. An X indicates the Instrument has no equivalent operation and emulation is not available. The instrument reports -113, “undefined header” for commands that have no implement.

Table 16. E8257 Emulated Commands

Command	Emulate
:CALibration	x
:SYSTem:COMMunicate	x
:DIAGnostic[:CPU]:INFOrmation	x
:DISPlay	x
IEEE 488.2 Common Commands	
*CLS	✓
*ESE	✓
*ESE?	✓
*ESR?	✓
*IDN?	✓
*OPC	✓
*OPC?	✓
*PSC	✓
*PSC?	x
*RCL	✓
*RST	x
*SAV	✓
*SRE	✓
*SRE?	✓
*STB?	✓
*TRG	✓
*TST?	✓
*WAI	
:LBFilter	x
:MEMory	x
:MMEMory	x
:OUTPut	
:BLANking:AUTO	x
:BLANking:[STATe]	x
:MODulation[:STATe]	x
:SETTled?	x
:SETTled:POLarity	x
:SETTled:RETRace	x
:SETTled:RFOFf	x

[:STATe]	✓
:ROUTe:HARDware:DGENerator	x
:STATus	
:OPERation:BASeband:CONDition	x
:OPERation:BASeband:ENABle	x
:OPERation:BASeband:NTRansition . . .	x
:OPERation:BASeband:PTRansition . . .	x
:OPERation:BASeband[:EVENt]	x
:OPERation:CONDition	✓
:OPERation:ENABle	✓
:OPERation:NTRansition	x
:OPERation:PTRansition	x
:OPERation[:EVENt]	✓
:PRESet	✓
:QUEStionable:CALibration:CONDition. .	x
:QUEStionable:CALibration:ENABle. . .	x
:QUEStionable:CALibration:NTRansition.	x
:QUEStionable:CALibration:PTRansition	x
:QUEStionable:CALibration[:EVENt] . .	x
:QUEStionable:CONDition	✓
:QUEStionable:ENABle	✓
:QUEStionable:FREQuency:CONDition . .	x
:QUEStionable:FREQuency:ENABle	x
:QUEStionable:FREQuency:NTRansition .	x
:QUEStionable:FREQuency:PTRansition .	x
:QUEStionable:FREQuency[:EVENt] . . .	x
:QUEStionable:MODulation:CONDition. .	x
:QUEStionable:MODulation:ENABle. . . .	x
:QUEStionable:MODulation:NTRansition.	x
:QUEStionable:MODulation:PTRansition .	x
:QUEStionable:MODulation[:EVENt] . . .	x
:QUEStionable:NTRansition.	x
:QUEStionable:POWer:CONDition	x
:QUEStionable:POWer:ENABle	x
:QUEStionable:POWer:NTRansition . . .	x
:QUEStionable:POWer:PTRansition . . .	x
:QUEStionable:POWer[:EVENt]	x
:QUEStionable:PTRansition	x
:QUEStionable[:EVENt]	✓

:SYSTem	
:ALTernate	x
:ALTernate:STAte	x
:CAPability	✓
:DATE	x
:ERRor[:NEXT]	✓not [:NEXT]
:ERRor:SCPI[:SYNTax] . .	x
:FILEsystem:SAFEmode . .	x
:HELP:MODE	x
:IDN	x
:LANGuage	✓
:OEMHead:FREQuency:STARt	x
:OEMHead:FREQuency:STOP	x
:OEMHead:SELect	x
:OEMHead:FREQuency:BAND WR15|WR12|WR10|WR8|WR6|WR5|WR3	x
:OEMHead:FREQuency:MULTiplier	x
:PON:TYPE	x
:PRESet	x
:PRESet:ALL	x
:PRESet:LANGuage . .	x
:PRESet:PERSistent.	x
:PRESet:PN9	x
:PRESet:TYPE	x
:PRESet[:USER]:SAVE	x
:SECurity:DISPlay .	x
:SECurity:ERASeall .	x
:SECurity:LEVel . .	x
:SECurity:LEVel:STATe	x
:SECurity:OVERwrite	x
:SECurity:SANitize .	x
:SSAVer:DELay . . .	x
:SSAVer:MODE	x
:SSAVer:STATe . . .	x
:TIME.	x
:VERSion.	✓
Trigger Subsystem:	
:ABORt	✓
:INITiate:CONTinuous[:ALL] . . .	✓ not [:ALL]
:INITiate[:IMMediate][:ALL]. . .	✓ not [:ALL]
:TRIGger:OUTPut:POLarity	x
:TRIGger[:SEQuence]:SLOPe . . .	x
:TRIGger[:SEQuence]:SOURce . . .	x
:TRIGger[:SEQuence][:IMMediate]	x

UNIT:POWer	✓
CORRection Subsystem:	x
FREQuency Subsystem:	
:FREQuency:CENTer	✓
:FREQuency:CHANnels:BAND	x
:FREQuency:CHANnels:NUMBer	x
:FREQuency:CHANnels[:STATe] . . .	x
:FREQuency:FIXed	✓not [UP|DOWN]
:FREQuency:MANual	x
:FREQuency:MODE	x
:FREQuency:MULTiplier	x
:FREQuency:OFFSet.	✓
:FREQuency:OFFSet:STATe	✓
:FREQuency:REFerence	x
:FREQuency:REFerence:SET	x
:FREQuency:REFerence:STATe	x
:FREQuency:SPAN.	✓
:FREQuency:STARt	✓
:FREQuency:STOP.	✓
:FREQuency[:CW]	✓
:FREQuency[:CW]:STEP[:INCRement] .	x
:FREQuency:SYNThesis	x
:PHASe:REFerence	x
:PHASe[:ADJust]	x
:ROSCillator:BANDwidth:DEFaults .	x
:ROSCillator:BANDwidth:EXTernal .	x
:ROSCillator:BANDwidth:INTernal .	x
:ROSCillator:SOURce:AUTO	x
:ROSCillator:SOURce	x
LIST	x
SWEep Subsystem:	
:SWEep:CONTrol:STATe	x
:SWEep:CONTrol:TYPE.	x
:SWEep:DWELl.	✓
:SWEep:GENeration. .	x
:SWEep:MODE	x
:SWEep:POINts. . . .	✓ query only
:SWEep:TIME.	✓
:SWEep:TIME:AUTO . .	x
:MARKer	x
Power Subsystem	
[:SOURce]:POWer .	✓

:ALC:BANDwidth|BWIDth.	x
:ALC:BANDwidth|BWIDth:AUTO	x
:ALC:LEVel	x
:ALC:SEARch	x
:ALC:SEARch:REFerence	x
:ALC:SEARch:REFerence:LEVel	x
:ALC:SEARch:SPAN:START	x
:ALC:SEARch:SPAN:STOP	x
:ALC:SEARch:SPAN:TYPE FULL|USER . .	x
:ALC:SEARch:SPAN[:STATe] ON|OFF|1|0	x
:ALC:SOURce	x
:ALC:SOURce:EXTernal:COUPling . . .	x
:ALC[:STATe]	x
:ATTenuation	x
:ATTenuation:AUTO.	x
:MODE	x
:NOISe[:STATe]	x
:POWer:LIMit[:MAX]:ADJust	x
:POWer:LIMit[:MAX]	x
:PROTection:STATe	x
:REFerence	x
:REFerence:STATe	x
:STARt	x
:STOP.	x
[:LEVel][:IMMediate]:OFFSet	x
[:LEVel][:IMMediate][:AMPLitude]. .	x
[:LEVel][:IMMediate][:AMPLitude]:STEP[:INCREment]	x
TSWeep	x
:AM	
:AM[1]|2....	✓
:AM:INTernal:FREQuency:STEP[:INCRement].	x
:AM:MODE	x
:AM:POLarity	x
:AM:WIDeband:SENSitivity	x
:AM:WIDeband:STATe	x
:AM[1]|2:EXTernal[1]|2:COUPling	x
:AM[1]|2:EXTernal[1]|2:IMPedance	x
:AM[1]|2:INTernal[1]|2:FREQuency	✓
:AM[1]|2:INTernal[1]:FREQuency:ALTernate	x
:AM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:PERCent	x
:AM[1]|2:INTernal[1]|2:FUNCtion:NOISe	x
:AM[1]|2:INTernal[1]|2:FUNCtion:RAMP .	x
:AM[1]|2:INTernal[1]|2:FUNCtion:SHAPe	✓ SINE only

:AM[1]|2:INTernal[1]:SWEep:RATE. . . .	x
:AM[1]|2:INTernal[1]:SWEep:TRIGger . .	x
:AM[1]|2:SOURce	x
:AM[1]|2:STATe	✓
:AM[1]|2:TYPE.	x
:AM[1]|2[:DEPTh]:EXPonential	x
:AM[1]|2[:DEPTh][:LINear]	x
:AM[1]|2[:DEPTh][:LINear]:TRACk . . .	x
:AM[:DEPTh]:STEP[:INCRement]	x
:FM	
:FM[1][2]	✓
:FM:INTernal:FREQuency:STEP[:INCRement]	x
:FM[1]|2:EXTernal[1]|2:COUPLing	x
:FM[1]|2:EXTernal[1]|2:IMPedance. . . .	x
:FM[1]|2:INTernal[1]:FREQuency:ALTernate	x
:FM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:PERCent	x
:FM[1]|2:INTernal[1]:SWEep:RATE . . .	x
:FM[1]|2:INTernal[1]:SWEep:TRIGger .	x
:FM[1]|2:INTernal[1]|2:FREQuency . .	✓
:FM[1]|2:INTernal[1]|2:FUNCtion:NOISe	x
:FM[1]|2:INTernal[1]|2:FUNCtion:RAMP	x
:FM[1]|2:INTernal[1]|2:FUNCtion:SHAPe	✓
:FM[1]|2:SOURce	x
:FM[1]|2:STATe	✓
:FM[1]|2[:DEViation]	✓
:FM[1]|2[:DEViation]:TRACk	x
:LFOutput	x
:PM	
:PM[1]|2...	✓
:PM:INTernal:FREQuency:STEP[:INCRement]	x
:PM[1]|2:BANDwidth|BWIDth	x
:PM[1]|2:EXTernal[1]|2:COUPling.	x
:PM[1]|2:EXTernal[1]|2:IMPedance	x
:PM[1]|2:INTernal[1]:FREQuency	✓
:PM[1]|2:INTernal[1]:FREQuency:ALTernate	x
:PM[1]|2:INTernal[1]|2:FUNCtion:NOISe. .	x
:PM[1]|2:INTernal[1]|2:FUNCtion:RAMP . .	x
:PM[1]|2:INTernal[1]:FREQuency:ALTernate:AMPLitude:PERCent	x
:PM[1]|2:INTernal[1]:FUNCtion:SHAPe	x
:PM[1]|2:INTernal2:FUNCtion:SHAPe.	✓ SINE only
:PM[1]|2:INTernal[1]:SWEep:RATE . .	x
:PM[1]|2:INTernal[1]:SWEep:TRIGger	x
:PM[1]|2:SOURce	x

:PM[1]|2:STATe	✓
:PM[1]|2[:DEViation]	✓
:PM[1]|2[:DEViation]:TRACk	x
:PM[:DEViation]:STEP[:INCRement]. .	x
:PULM	x
5 Digital Modulation Commands	x
6 Digital Signal Interface Module Commands	x

IEEE Bus Trace Guide

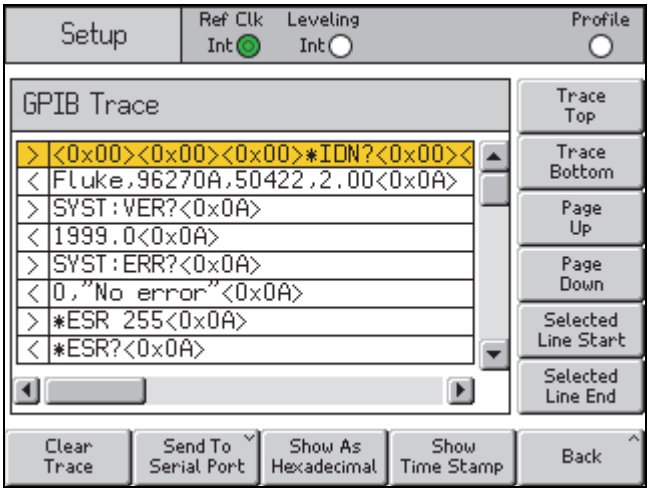
This section documents the Instrument IEEE Bus Trace program. The program is an integral part of the instrument functionality. Its purpose is to record GPIB bus transactions to aid in the diagnosis of GPIB remote control issues. Bus transactions are stored in a FIFO 2 Mb buffer. As the buffer fills, newer data pushes the oldest data out of the buffer, discarding it.

Note

Stored bus transaction data is not retained after the Instrument is powered down.

To use the trace facility:

1. Set up the GPIB preferences and run the remote commands in question.
2. When the command sequence is complete, push the Go to Local softkey. This will place the instrument under local control.
3. Push SETUP .
4. Push the GPIB Preferences softkey.
5. From the GPIB Personality screen, push the GPIB Trace softkey. This will bring up the GPIB Trace screen and display the contents of the trace buffer. See Figure .



ead351f.bmp

Figure 28. GPIB Trace Screen

The field on the left contains either a < or > character to indicate the direction of the message. The > character means a command directed at the instrument, and the < character indicates a response from the instrument.

In ASCII mode, any character that falls outside the range 32 to 127 will be displayed as hexadecimal. For example, the new-line character is shown as <0x0A>.

The GPIB Trace Softkeys and Menus

Buffer Navigation

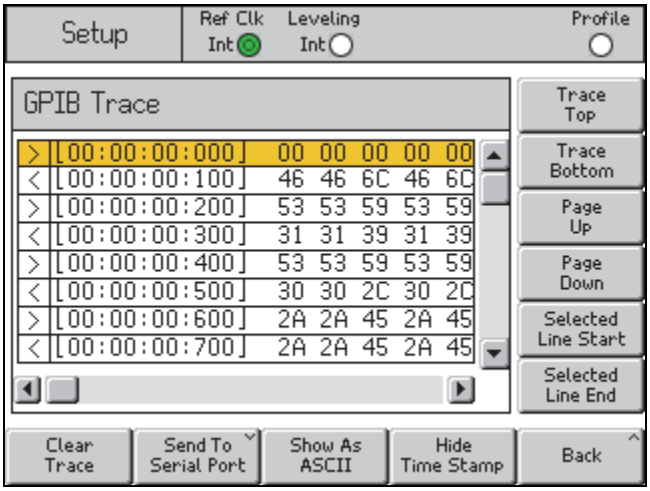
The right-hand softkeys marked Trace Top, Trace Bottom, Page Up, and Page Down allow the user to move up and down the trace buffer. To move one line at a time, use the ⏶⏷ cursor keys (or the navigation wheel).

To view the beginning and end of a long line, use the Selected Line Start and Selected Line End softkeys, (a line is considered selected when the yellow line cursor is over it).

The ⏴⏵ cursor keys allow the user to scroll the trace display left and right one character at a time.

Display Formatting

By default, the sent/received trace strings are shown in ASCII format. They can, however, be modified to display in hexadecimal and to include a prefixed time stamp, as shown in the Figure .



ead352f.bmp

Figure 29. GPIB Trace Display with both Timestamp and Hexadecimal On

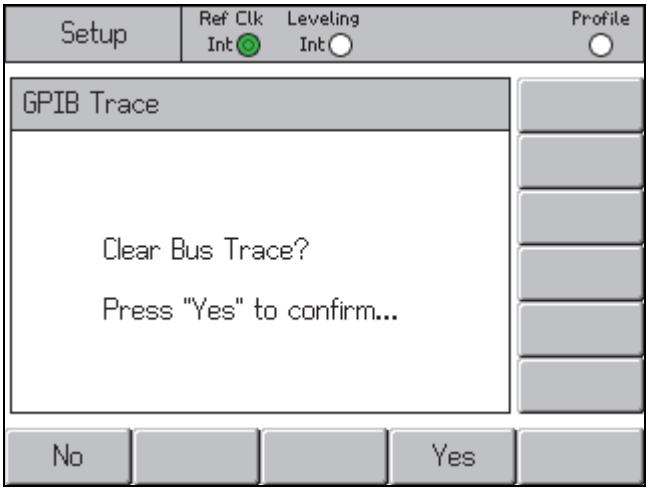
Two softkeys at the bottom of the trace screen operate in a toggle mode to control the display format, that is, Show As ASCII/Hexadecimal and Show/Hide Time Stamp.

Clearing a Trace

There are two ways to clear the trace buffer (and screen):

- Power the Instrument off and then on.
- Push the Clear Trace softkey (present while the GPIB Trace screen is active).

  This method also shows a confirmation screen to prevent accidental erasure. See Figure .



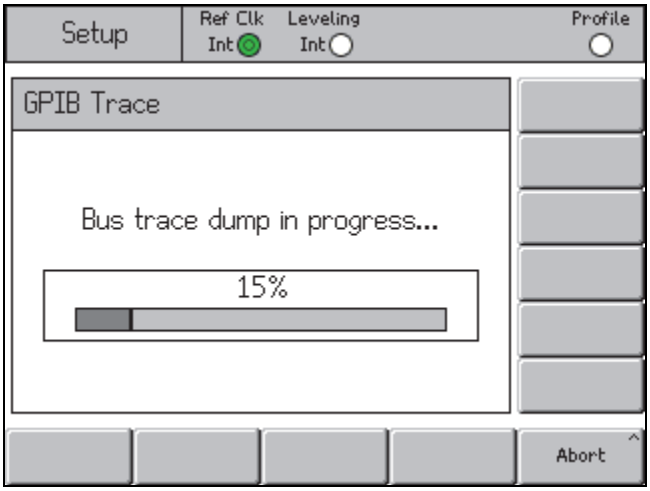
ead353f.bmp

Figure 30. Clear Trace Confirmation Screen

Export the Trace Data

The user can transfer the contents of the trace buffer to a laptop or PC by using the serial port on the rear of the Instrument. The data is exported in ASCII format with a time stamp.

To use the export function, configure a terminal emulator on the PC or laptop with the settings shown below, connect a null- modem cable, and push the Send to Serial Port softkey to start a transfer. Push the Abort softkey to cancel transfers at any time. See Figure .



ead354f.bmp

Figure 31. Exporting the Contents of a GPIB Trace Buffer

Configure a Terminal Emulator

Any terminal emulator can be used to receive the exported trace data. This includes a HyperTerminal, Tera Term Pro, or any of the many VT100 terminal emulators.

Use the subsequent settings to configure the serial port and terminal settings for the computer (or laptop):

Baud Rate: 115200
Parity: None
Word length: 8 Bits
Stop Bits: 1 Bit
Handshaking: None (no hardware handshaking enabled)
Local Echo Off
No need to set *line-feed with carriage return*

Construct a Null Modem Cable

The cable necessary for exporting data is a standard DB-9 female-to-female null-modem cable. To construct a null-modem cable, use the following wire/pin connections. Leave the rest of the pins unconnected.

PC Connector		Instrument Connector
2	⟷	3
3	⟷	2
5	⟷	5