

PROGRAMMING REFERENCE MANUAL

HP 53310A

Modulation Domain Analyzer

MANUAL APPLICABILITY

This manual applies directly to an HP 53310A having a serial number prefix up to and including the one listed below. If the serial number prefix on your instrument is higher than the one shown below, refer to the "Manual Updating Changes" included with this manual.

For additional important information about serial numbers, see "Instruments Covered by This Manual" in the Introduction of the HP 53310A Operating Reference Manual.

SERIAL NUMBER

Serial Number Prefix: 3121

Edition 1

© Copyright HEWLETT-PACKARD COMPANY 1992
5301 STEVENS CREEK BOULEVARD, SANTA CLARA, CA 95052-8059

All rights reserved.

Printed NOVEMBER 1992

MANUAL PART NO. 53310-90038



Certification and Warranty

CERTIFICATION

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology (formerly National Bureau of Standards), to the extent allowed by that organization's calibration facility, and to the calibration facilities of other International Standards Organization members.

WARRANTY

This Hewlett-Packard instrument product is warranted against defects in material and workmanship for a period of one year from date of shipment. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products which prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by HP. Buyer shall prepay shipping charges to HP and HP shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to HP from another country.

HP warrants that its software and firmware designated by HP for use with an instrument will execute its programming instructions when properly installed on that instrument. HP does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

LIMITATION OF WARRANTY

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HP SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

EXCLUSIVE REMEDIES

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HP SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

ASSISTANCE

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Support Office. Addresses are provided at the back of this manual.

Safety Considerations

GENERAL

This product and related documentation must be reviewed for familiarization with safety markings and instructions before operation.

This product is a Safety Class I instrument (provided with a protective earth terminal).

BEFORE APPLYING POWER

Verify that the product is set to match the available line voltage and the correct fuse is installed. Refer to instructions in appendix A.

SAFETY EARTH GROUND

An uninterruptible safety earth ground must be provided from the mains power source to the product input wiring terminals or supplied power cable.

SAFETY SYMBOLS



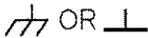
Instruction manual symbol; the product will be marked with this symbol when it is necessary for the user to refer to the instruction manual.



Indicates hazardous voltages.



Indicates earth (ground) terminal.



Indicates terminal is connected to chassis when such connection is not apparent.



Alternating current.



Direct current.

WARNING

THIS DENOTES A HAZARD. IT CALLS ATTENTION TO A PROCEDURE, PRACTICE, OR THE LIKE, WHICH, IF NOT CORRECTLY PERFORMED OR ADHERED TO, COULD RESULT IN PERSONAL INJURY. DO NOT PROCEED BEYOND A WARNING SIGN UNTIL THE INDICATED CONDITIONS ARE FULLY UNDERSTOOD AND MET.

CAUTION

This denotes a hazard. It calls attention to an operating procedure, practice, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product. Do not proceed beyond a **CAUTION** sign until the indicated conditions are fully understood and met.

**SAFETY
INFORMATION****Warning**

Any interruption of the protective grounding conductor (inside or outside the instrument) or disconnecting the protective earth terminal will cause a potential shock hazard that could result in personal injury. (Grounding one conductor of a two conductor outlet is not sufficient protection.)

Whenever it is likely that the protection has been impaired, the instrument must be made inoperative and be secured against any unintended operation.

If this instrument is to be energized via an autotransformer (for voltage reduction) make sure the common terminal is connected to the earthed pole terminal (neutral) of the power source.

Instructions for adjustments while covers are removed and for servicing are for use by service-trained personnel only. To avoid dangerous electric shock, do not perform such adjustments or servicing unless qualified to do so.

For continued protection against fire, replace the line fuse(s) only with 250V fuse(s) of the same current rating and type (for example, normal blow, time delay). Do not use repaired fuses or short circuited fuseholders.

When measuring power line signals, be extremely careful and always use a step-down isolation transformer whose output voltage is compatible with the input measurement capabilities of this product. This product's front and rear panels are typically at earth ground, so **NEVER TRY TO MEASURE AC POWER LINE SIGNALS WITHOUT AN ISOLATION TRANSFORMER.**

v

Regulatory Notice

**HP 53310A
Modulation Domain Analyzer**

ACCOUSTIC NOISE EMISSION:

LpA 47 dB at operator position, at normal operation, tested per ISO 7779. All data are the results from type test.

(German)

GERAeUSCHEMISSION:

LpA 47 dB am Arbeits platz, normaler Betrieb, geprueft nach DIN 45635 Teil 19. Die Angaben beruhen auf Ergebnissen von Typpruefungen.

Table of Contents

Before You Start ...

| | |
|------------------------------|-----------|
| Welcome | Preface-1 |
| Getting Started | Preface-1 |
| How to Use This Manual | Preface-2 |
| Manual Contents | Preface-4 |
| Assumptions | Preface-6 |

1 PROGRAMMING OVERVIEW

| | |
|--|------|
| In This Chapter | 1-1 |
| SCPI Commands | 1-2 |
| SCPI Conformance Information | 1-5 |
| Subsystem Commands Summary Table | 1-5 |
| Autoscale Group Commands Summary Table | 1-18 |
| Common Commands Summary Table | 1-22 |
| Front Panel To SCPI Command Mapping | 1-23 |
| Introduction To SCPI Programming | 1-54 |
| Interfacing to an External Computer | 1-54 |
| Programming Languages | 1-55 |
| Initialization | 1-57 |
| Remote Mode | 1-58 |
| SCPI Summary | 1-59 |
| How to Write SCPI Programs | 1-63 |

2 SUBSYSTEM COMMANDS REFERENCE DICTIONARY

| | |
|---|------|
| In This Chapter | 2-1 |
| Subsystems | 2-2 |
| :ABORt Subsystem | 2-5 |
| Autoscale Group | 2-7 |
| :CONFigure Subsystem | 2-15 |
| :CONFigure:HISTogram[:VOLTage]:FREQuency[1 2 3] | 2-16 |
| :CONFigure:HISTogram[:VOLTage]:TINTerval | 2-17 |
| :CONFigure:XTIME[:VOLTage]:FREQuency[1 2 3] | 2-18 |
| :CONFigure:XTIME[:VOLTage]:TINTerval | 2-19 |
| :FETCh Subsystem | 2-20 |

Table of Contents

2 SUBSYSTEM COMMANDS REFERENCE DICTIONARY (Continued)

| | |
|---|------|
| :MEASure Subsystem | 2-23 |
| :MEASure:HISTogram[:VOLTage]:FREQuency[1 2 3]? | 2-24 |
| :MEASure:HISTogram[:VOLTage]:TINTerval? | 2-25 |
| :MEASure:XTIME[:VOLTage]:FREQuency[1 2 3]? | 2-26 |
| :MEASure:XTIME[:VOLTage]:TINTerval? | 2-27 |
| :READ Subsystem | 2-28 |
| | |
| :CALCulate Subsystem | 2-31 |
| :CALCulate:ANALysis:AOFF | 2-36 |
| :CALCulate:ANALysis:DELimit[1 2] | 2-37 |
| :CALCulate:ANALysis:<analy_fcn>[:STATe] | 2-38 |
| :CALCulate:ANALysis:<analy_fcn>:RESult? | 2-39 |
| :CALCulate:DATA? | 2-41 |
| :CALCulate:DIMension[1 2] Subtree | 2-42 |
| :CALCulate:DIMension[1 2]:ENCode:NOValue? | 2-43 |
| :CALCulate:DIMension[1 2]:FORMat? | 2-44 |
| :CALCulate:DIMension[1 2]:LABel? | 2-45 |
| :CALCulate:DIMension[1 2]:OFFSet? | 2-46 |
| :CALCulate:DIMension[1 2]:SCALE? | 2-47 |
| :CALCulate:DIMension[1 2]:SIZE? | 2-48 |
| :CALCulate:DIMension[1 2]:TYPE? | 2-49 |
| :CALCulate:HISTogram[:STATe] | 2-50 |
| :CALCulate:HISTogram:ACCumulate | 2-51 |
| :CALCulate:HISTogram:SAMPles:ABOVE? | 2-52 |
| :CALCulate:HISTogram:SAMPles:BELOW? | 2-53 |
| :CALCulate:HISTogram:SAMPles:IN? | 2-54 |
| :CALCulate:IMMEDIATE | 2-55 |
| :CALCulate:PATH? | 2-56 |
| | |
| :CLEar Subsystem | 2-57 |
| | |
| :DIAGnostic Subsystem | 2-59 |
| :DIAGnostic:CALibration:INPut[1 2]:GAIN:AUTO | 2-60 |
| :DIAGnostic:CALibration:INPut[1 2]:OFFSet:AUTO | 2-61 |
| :DIAGnostic:CALibration:INTerp:AUTO | 2-62 |
| :DIAGnostic:CALibration:STATus? | 2-63 |
| :DISPlay Subsystem | 2-65 |
| :DISPlay[:WINDow]:TRACe:CONNect | 2-68 |
| :DISPlay[:WINDow]:TRACe:GRATicule:AXIS[:STATe] | 2-69 |
| :DISPlay[:WINDow]:TRACe:GRATicule:FRAMe[:STATe] | 2-70 |

2 SUBSYSTEM COMMANDS REFERENCE DICTIONARY (Continued)

| | |
|---|-------|
| :DISPlay[:WINDow]:TRACe:GRATicule:GRID[:STATe] | 2-71 |
| :DISPlay[:WINDow]:TRACe:PERsistence | 2-72 |
| :DISPlay[:WINDow]:TRACe:PROBability[:SCALe]:AUTO | 2-73 |
| :DISPlay[:WINDow]:TRACe:PROBability[:SCALe]:MAXimum | 2-74 |
| :DISPlay[:WINDow]:TRACe:PROBability[:SCALe]:MINimum | 2-75 |
| :DISPlay[:WINDow]:TRACe:PROBability:SPACing | 2-76 |
| :DISPlay:MENU[:NAME] | 2-77 |
| | |
| :FORMat Subsystem | 2-79 |
| :FORMat[:DATA] | 2-80 |
| | |
| :INITiate Subsystem | 2-83 |
| :INITiate[:IMMediate] | 2-84 |
| :INITiate:CONTinuous | 2-85 |
| | |
| :INPut[1 2] Subsystem | 2-87 |
| :INPut[1 2]:COUPling | 2-88 |
| :INPut[1 2]:IMPedance | 2-89 |
| | |
| :MARKer Subsystem | 2-91 |
| :MARKer:FREQuency[:STATe] | 2-93 |
| :MARKer:FREQuency:POSition[1 2] | 2-94 |
| :MARKer:FREQuency:POSition[1 2]:AUTO | 2-95 |
| :MARKer:PROBability[:STATe] | 2-96 |
| :MARKer:PROBability:POSition[1 2]? | 2-97 |
| :MARKer:TIME[:STATe] | 2-98 |
| :MARKer:TIME:POSition[1 2] | 2-99 |
| :MARKer:TINTerval[:STATe] | 2-100 |
| :MARKer:TINTerval:POSition[1 2] | 2-101 |
| :MARKer:TINTerval:POSition[1 2]:AUTO | 2-102 |
| | |
| [[:SENSe] Subsystem | 2-103 |
| [[:SENSe]:DATA? | 2-104 |
| [[:SENSe]:DIMension[1 2] Subtree | 2-106 |
| [[:SENSe]:DIMension[1 2]:ENCode:NOValue? | 2-107 |
| [[:SENSe]:DIMension[1 2]:FORMat? | 2-108 |
| [[:SENSe]:DIMension[1 2]:LABel? | 2-109 |
| [[:SENSe]:DIMension[1 2]:OFFSet? | 2-110 |
| [[:SENSe]:DIMension[1 2]:SCALe? | 2-111 |
| [[:SENSe]:DIMension[1 2]:SIZE? | 2-112 |

Table of Contents

2 SUBSYSTEM COMMANDS REFERENCE DICTIONARY (Continued)

| | |
|--|--------|
| [.SENSe]:DIMension[1 2]:TYPE? | 2-113 |
| [.SENSe]:EVENT[1 2] Subtree | 2-114 |
| [.SENSe]:EVENT[1 2]:HYSTeresis:RELative | 2-115 |
| [.SENSe]:EVENT[1 2]:LEVel | 2-116 |
| [.SENSe]:EVENT[1 2]:LEVel:AUTO | 2-117 |
| [.SENSe]:FREQUency Subtree | 2-118 |
| [.SENSe]:FREQUency:ARM:CFINterval[1 2 3] | 2-120 |
| [.SENSe]:FREQUency:ARM:CFINterval[1 2 3]:AUTO | 2-121 |
| [.SENSe]:FREQUency:ARM:ECOUNT[1 2]? | 2-122 |
| [.SENSe]:FREQUency:ARM:ECOUNT[1 2]:MULTiplier | 2-123 |
| [.SENSe]:FREQUency:ARM:FCFINterval[1 2 3] | 2-123a |
| [.SENSe]:FREQUency:ARM:FCFINterval[1 2 3]:AUTO | 2-123b |
| [.SENSe]:FREQUency:ARM:LEVel | 2-124 |
| [.SENSe]:FREQUency:ARM:LINK? | 2-125 |
| [.SENSe]:FREQUency:ARM:SLOPe | 2-126 |
| [.SENSe]:FREQUency:ARM:SOURce | 2-127 |
| [.SENSe]:FREQUency:ARM:TImer | 2-129 |
| [.SENSe]:FREQUency:ARM:TImer:STATUs? | 2-130 |
| [.SENSe]:FREQUency:RANGe[1 2 3]:[UPPer] | 2-131 |
| [.SENSe]:FREQUency:RANGe[1 2 3]:AUTO | 2-132 |
| [.SENSe]:FREQUency:RANGe[1 2 3]:CENTer:AUTO | 2-133 |
| [.SENSe]:FREQUency:RANGe[1 2 3]:LOWer | 2-134 |
| [.SENSe]:FUNCTion | 2-135 |
| [.SENSe]:HISTogram Subtree | 2-137 |
| [.SENSe]:HISTogram:ACCumulate | 2-138 |
| [.SENSe]:HISTogram:ARM:LEVel | 2-139 |
| [.SENSe]:HISTogram:ARM:SLOPe | 2-140 |
| [.SENSe]:HISTogram:ARM:SOURce | 2-141 |
| [.SENSe]:HISTogram:SAMPles:ABOVe?" | 2-142 |
| [.SENSe]:HISTogram:SAMPles:BELOW? | 2-143 |
| [.SENSe]:HISTogram:SAMPles:IN? | 2-144 |
| [.SENSe]:HISTogram:TRIGger:COUNT | 2-145 |
| [.SENSe]:ROSCillator:SOURce? | 2-146 |
| [.SENSe]:SWEep Subtree | 2-147 |
| [.SENSe]:SWEep:TIME:DELay | 2-149 |
| [.SENSe]:SWEep:TIME:DELay:LINK | 2-150 |
| [.SENSe]:SWEep:TIME:PANorama:[STATe] | 2-151 |
| [.SENSe]:SWEep:TIME:PANorama:DELay | 2-152 |
| [.SENSe]:SWEep:TIME:PANorama:DELay:LINK | 2-153 |
| [.SENSe]:SWEep:TIME:PANorama:SPAN | 2-154 |

2 **SUBSYSTEM COMMANDS REFERENCE DICTIONARY (Continued)**

| | |
|--|-------|
| [:SENSE]:SWEep:TIME:REPetitive[:STATe] | 2-155 |
| [:SENSE]:SWEep:TIME:REPetitive:AVERAge | 2-156 |
| [:SENSE]:SWEep:TIME:SPAN | 2-157 |
| [:SENSE]:TINTerval Subtree | 2-158 |
| [:SENSE]:TINTerval:ARM:ATIMer? | 2-160 |
| [:SENSE]:TINTerval:ARM:ECOUNT | 2-161 |
| [:SENSE]:TINTerval:ARM:LEVel | 2-162 |
| [:SENSE]:TINTerval:ARM:SLOPe[1 2 3] | 2-163 |
| [:SENSE]:TINTerval:ARM:SOURce | 2-164 |
| [:SENSE]:TINTerval:ARM:TIMer | 2-165 |
| [:SENSE]:TINTerval:RANGe[:UPPer] | 2-166 |
| [:SENSE]:TINTerval:RANGe:AUTO | 2-167 |
| [:SENSE]:TINTerval:RANGe:CENTer:AUTO | 2-168 |
| [:SENSE]:TINTerval:RANGe:LOWer | 2-169 |
| [:SENSE]:TINTerval:ROUTe | 2-170 |
| [:SENSE]:TINTerval:SLOPe[1 2] | 2-171 |
| [:SENSE]:TINTerval:TYPE | 2-172 |
| | |
| :STATus Subsystem | 2-173 |
| :STATus:OPERation[:EVENT]? | 2-178 |
| :STATus:QUESTionable[:EVENT]? | 2-178 |
| :STATus:QUESTionable:FREQUency[:EVENT]? | 2-178 |
| :STATus:OPERation:CONDition? | 2-179 |
| :STATus:QUESTionable:CONDition? | 2-179 |
| :STATus:QUESTionable:FREQUency:CONDition? | 2-179 |
| :STATus:OPERation:ENABle | 2-180 |
| :STATus:QUESTionable:ENABle | 2-180 |
| :STATus:QUESTionable:FREQUency:ENABle | 2-180 |
| :STATus:OPERation:NTRansition | 2-182 |
| :STATus:QUESTionable:NTRansition | 2-182 |
| :STATus:QUESTionable:FREQUency:NTRansition | 2-182 |
| :STATus:OPERation:PTRansition | 2-183 |
| :STATus:QUESTionable:PTRansition | 2-183 |
| :STATus:QUESTionable:FREQUency:PTRansition | 2-183 |
| :STATus:PRESet | 2-184 |
| | |
| :SYSTem Subsystem | 2-187 |
| :SYSTem:CLICker | 2-188 |
| :SYSTem:ERRor? | 2-189 |
| :SYSTem:KEY | 2-190 |

Table of Contents

2 SUBSYSTEM COMMANDS REFERENCE DICTIONARY (Continued)

| | |
|--|-------|
| :SYSTem:KEY:LOG? | 2-192 |
| :SYSTem:PRESet | 2-193 |
| :SYSTem:PRINt? | 2-194 |
| :SYSTem:SSAVer | 2-195 |
| :SYSTem:VERSion? | 2-196 |
| :TRIGger Subsystem | 2-197 |
| :TRIGger:AUTO | 2-199 |
| :TRIGger:FREQuency:EEMode[1 2 3] | 2-199 |
| :TRIGger:FREQuency:LEVel[1 2 3] | 2-200 |
| :TRIGger:FREQuency:LEVel[1 2 3]:AUTO | 2-201 |
| :TRIGger:FREQuency:SLOPe | 2-202 |
| :TRIGger:HFReject | 2-203 |
| :TRIGger:LEVel | 2-204 |
| :TRIGger:LINK? | 2-205 |
| :TRIGger:SLOPe | 2-206 |
| :TRIGger:SOURce | 2-207 |
| :TRIGger:TINTerval:EEMode | 2-208 |
| :TRIGger:TINTerval:LEVel | 2-209 |
| :TRIGger:TINTerval:LEVel:AUTO | 2-210 |
| :TRIGger:TINTerval:SLOPe | 2-210 |

3 COMMON COMMANDS REFERENCE DICTIONARY

| | |
|---|------|
| In This Chapter | 3-1 |
| *CAL? (Calibration) | 3-4 |
| *CLS (Clear Status) | 3-5 |
| *ESE (Event Status Enable) | 3-6 |
| *ESR? (Event Status Register Query) | 3-8 |
| *IDN? (Identification Query) | 3-10 |
| *OPC (Operation Complete) | 3-11 |
| *OPC? (Operation Complete) | 3-12 |
| *OPT? (Identification Query) | 3-13 |
| *RCL (Recall) | 3-14 |
| *RST (Reset) | 3-15 |
| *SAV (Save) | 3-16 |
| *SRE (Service Request Enable) | 3-17 |
| *STB? (Status Byte Query) | 3-19 |
| *TRG (Trigger) | 3-21 |
| *TST? (Self-Test Query) | 3-22 |
| *WAI (Wait to Continue) | 3-23 |

Table of Contents

4 STATUS REPORTING
In This Chapter 4-1
Status Reporting Overview 4-1
Standard Status Data Structure 4-9
SCPI Status Reporting 4-18
Has a Monitored Event Occurred? 4-30
Interrupting an External Computer 4-32
What Monitored Event Has Occurred? 4-34

5 DATA TRANSFER
In This Chapter 5-1
Data Transfer Overview 5-1
Commands Used to Transfer Data 5-8
Commands Used to Transfer Measured Data Values 5-9

A HP-IB INTERCONNECTION
In This Appendix A-1
Connecting to a Computer A-1
Connecting to a Printer A-2
Operating Modes A-2
Setting the HP-IB Operating Mode A-3
Setting the HP-IB Address A-3

B HP-IB GENERAL INFORMATION
In This Appendix B-1
HP-IB Description B-2
Analyzer Interface Capabilities B-4
HP-IB Address Selection B-8
Interface Commands B-8
Remote Operation B-15
Data Messages B-19
Receiving the Abort Message B-19

Table of Contents

C SCPI CONCEPTS

| | |
|-------------------------------------|-----|
| In This Appendix | C-1 |
| SCPI Overview | C-1 |
| Program and Response Messages | C-4 |
| Command Types | C-6 |
| Typical Measurements | C-7 |
| Status Reporting | C-7 |
| Data Transfer | C-7 |
| Arming and Triggering | C-7 |

D PROGRAM EXAMPLES

| | |
|------------------------|-----|
| In This Appendix | D-1 |
| Overview | D-2 |

E RELATED DOCUMENTATION

| | |
|-----------------------------|-----|
| In This Appendix | E-1 |
| Related Documentation | E-1 |

F ERROR MESSAGES

| | |
|-------------------------------|-----|
| In This Appendix | F-1 |
| Reading the Error Queue | F-1 |
| Error Types | F-2 |
| Error Queue | F-3 |
| Error Numbers | F-4 |

Before You Start ...

Welcome

Welcome to the *HP 53310A Modulation Domain Analyzer Programming Reference Manual*. This Analyzer may very well be the first stand-alone instrument that you have programmed that conforms to the *Standard Commands for Programmable Instruments (SCPI) Standard Version 1990.0*.

Don't worry, the SCPI standard does not completely redefine how to program instruments over the Hewlett-Packard Interface Bus (HP-IB). However, it does standardize the structure and content of an instrument's command set to reflect the best programming practices developed by people using HP-IB. It also establishes standard command mnemonics for similar functions in all of the instruments that conform to the SCPI standard.

If you have programmed any HP instruments that have been released over the last few years, you will have seen a general trend toward the techniques specified in the SCPI standard. For example, several instruments are already using a hierarchy of commands that is similar to the command structure defined by the SCPI standard.

Getting Started

Before attempting to program the Analyzer, take some time to familiarize yourself with the content of this manual. The remainder of this Preface contains the following information:

- An explanation of how you should use the manual based on your experience programming instruments and your testing requirements.
- A description of the manual contents.
- A statement of the assumptions that are made in the manual.

How to Use This Manual

How to Use This Manual

How you use this manual depends upon how much you already know about programming instruments and how complex your measurement requirements are. Lets start by establishing your programming background and then discuss the type of measurements you want to do.

NOTE

You could always read the entire manual from front to back, but you would probably find that approach both boring and unproductive.

New Users

As a new user, you should understand that you must have some understanding of a high-level language such as Pascal, BASIC, C, or FORTRAN before you can use the command set defined in this manual to control the Analyzer. Knowing BASIC is helpful because most of the examples are written in HP BASIC (there are also four Microsoft QuickBASIC programming examples in appendix D, (Programming Examples). However, whatever language you use, the command strings that control the Analyzer remain the same.

Learning to Program the Analyzer

To learn how to program the Analyzer, do the following:

- Scan the summary tables in chapter 1 "Programming Overview" and the syntax drawings in chapter 2, "Subsystem Commands" to get a feeling for the number and structure of commands available to you.
- Go to the *Quick Start Guide* and perform enough of the examples to become familiar with the Analyzer. For additional examples and descriptions of front-panel operation, see the *Operating Reference Manual*.
- Study the "Front Panel to SCPI Command Mapping" drawings and read the "Introduction to SCPI Programming" information in chapter 1, "Programming Overview." There is a programming

How to Use This Manual

flowchart at the end of the chapter that illustrates some of the decisions that you must make when programming the Analyzer.

- Read appendix C “SCPI Concepts,” if you want additional information on how SCPI concepts are applied in the Analyzer.
- Write some simple programs to perform measurements with the Analyzer. Use the Autoscale Group of commands and observe the automatically selected parameters to get a feel for the range of parameters used with each type of signal. You can observe these parameters directly on the Analyzer front-panel display or you can return individual parameter values using the query form of each command.
- Modify some programs to select specific aspects of the measured signal. If the programs work, consider yourself an experienced programmer and read the instructions for that category.

Experienced Programmers

If you have programmed other HP-IB instruments, you will probably be familiar with many of the concepts and techniques discussed in this manual. Also, you will find that using the SCPI commands is very similar to using older HP-IB commands. The main difference is the hierarchy of the subsystem commands (and this type of structure has been used on other instruments previously).

However, because the SCPI command set and some of the status reporting techniques are new, you may want to use the following sequence to learn the Analyzer programming requirements:

- Look over the steps for a new user and perform any that you think are applicable to your current level of knowledge. In particular, look at the measurement techniques and examples described in the *Operating Reference Manual*.
- Review the summary tables in chapter 1 or the *Programming Quick Reference Guide* (provided

Manual Contents

with the Analyzer). If they contain sufficient information to get you started, write some programs to explore the Analyzer's capabilities. If you need additional information on any command, refer to the applicable command description in chapters 2 and 3.

- Review the remaining information in this manual to determine what is applicable to your programming requirements.

If you need more information than is contained in this manual, see appendix E "Related Documentation."

Applications

After you have read the general information and written some measurement programs, you may want to expand the scope of your applications. The following two techniques are explained in detail:

- If you are going to write interrupt-driven programs (or if you just want to determine the status of the Analyzer), read chapter 4 "Status Reporting."
- If you are going to write programs to transfer data between the Analyzer and an external computer, read chapter 5 "Data Transfer."

Manual Contents

In addition to this Preface, the following information is contained in this manual:

- Chapter 1 "Programming Overview" introduces the Standard Commands for Programmable Instruments (SCPI) structure and commands. It also includes general information required to program the Analyzer and a task-oriented programming flowchart. The flowchart illustrates a recommended approach to organizing the SCPI commands and tasks required to implement efficient programs. This is not a mandatory sequence, but it does assure that you will consider all of your options.

Manual Contents

- Chapter 2 “Subsystem Commands Reference Dictionary” describes the SCPI subsystem commands.
- Chapter 3 “Common Commands Reference Dictionary” describes the Common Commands (defined by IEEE 488.2-1987).
- Chapter 4 “Status Reporting” describes how to perform status reporting using the Common Commands and the Status subsystem commands.
- Chapter 5 “Data Transfer” describes how to transfer data between the Analyzer and an external computer.
- Appendix A “HP-IB Interconnection” describes how to connect an external computer to the Analyzer, how to set the HP-IB Address, and how to use a printer with the Analyzer.
- Appendix B “General HP-IB Information” contains additional information about interface functions and bus commands that are applicable to the Analyzer.
- Appendix C “SCPI Concepts” explains the SCPI concepts and formats that are directly applicable to the Analyzer.
- Appendix D “Programming Examples” contains programming examples that illustrate specific applications for the Analyzer.
- Appendix E “Related Documentation” lists documentation that is related to the use of the Analyzer.
- Appendix F “Error Messages” lists all the error messages the Analyzer can generate and what caused the error.
- Glossary
- Index

Assumptions

Assumptions

This manual assumes the Analyzer is correctly installed and interfaced to an external computer. If it is not, see Appendix A "HP-IB Interconnection" for the information required to connect the Analyzer to a computer.

This manual also assumes you are familiar with the front-panel operation of the Analyzer. See the *Operating Reference Manual* for detailed information about front-panel operation. Knowing how to control the Analyzer from the front panel and understanding the measurements you wish to perform makes the programming task much easier.

PROGRAMMING OVERVIEW

In This Chapter

This chapter provides a programming overview of the Standard Commands for Programmable Instruments (SCPI) Common and Subsystem Commands as they apply to the HP 53310A Modulation Domain Analyzer (Analyzer).

This chapter presents the material in the following order:

- A brief description of the SCPI command structure.
- A summary of the SCPI Conformance Information.
- The Subsystem Commands Summary Table.
- The Autoscale Group Commands Summary Table.
- The Common Commands Summary Table.
- A cross-reference between the front-panel keys and menus and the associated SCPI commands.
- A general introduction on how to program the Analyzer.

The summary tables show and define the structure of the commands used in the Analyzer. Besides defining the command structure, they provide the experienced user with a quick reference for programming the Analyzer. In addition, a separate *Programming Quick Reference Guide* is provided for the Analyzer.

The Autoscale Group Commands are a subset of the Subsystem Commands. They are presented separately because they are the equivalent of the Autoscale key on the front panel and allow you to make measurements with a minimum of commands.

The cross reference between the front-panel operation and the associated SCPI commands allows you to quickly find the appropriate SCPI commands to implement a program that duplicates a measurement you have performed manually.

SCPI Commands

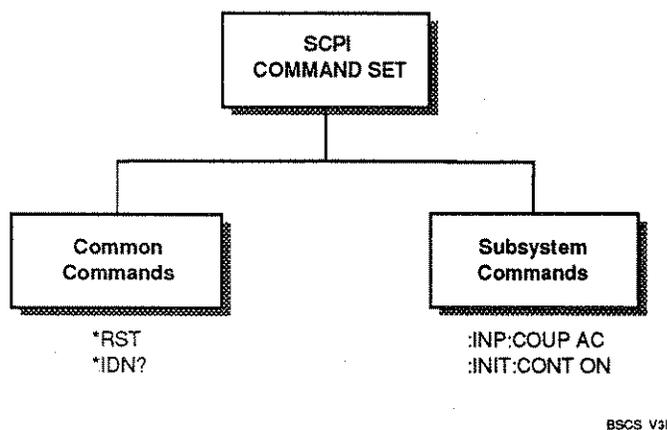
The general introduction on how to program the Analyzer contains a collection of facts you need to know about programming instruments. It is not intended as a tutorial. However, it does give you enough information to get you started programming the Analyzer.

SCPI Commands

The SCPI commands are separated into two types: Common Commands and Subsystem Commands (see *figure 1-1*).

Common Commands

The common commands are general purpose commands that are common to all instruments (as defined in IEEE 488.2). Common commands are easy to recognize because they all begin with an "*" (for example, *RST, *IDN?, *OPC). These commands are generally not related to measurement configuration. They are used for functions like resetting the instrument, identification, or synchronization.



BSCS_V3I

Figure 1-1. SCPI Command Structure

**Subsystem
Commands**

Subsystem commands include all measurement functions and some general purpose functions. The Autoscale Group Commands are a subset of the Subsystem Commands. Subsystem commands use a hierarchy relationship between keywords that is indicated by a “:” (colon). For example, in the SYST:ERR? query, the “:” between SYST and ERR? indicates ERR? is subordinate to SYST.

**Program Message
Syntax**

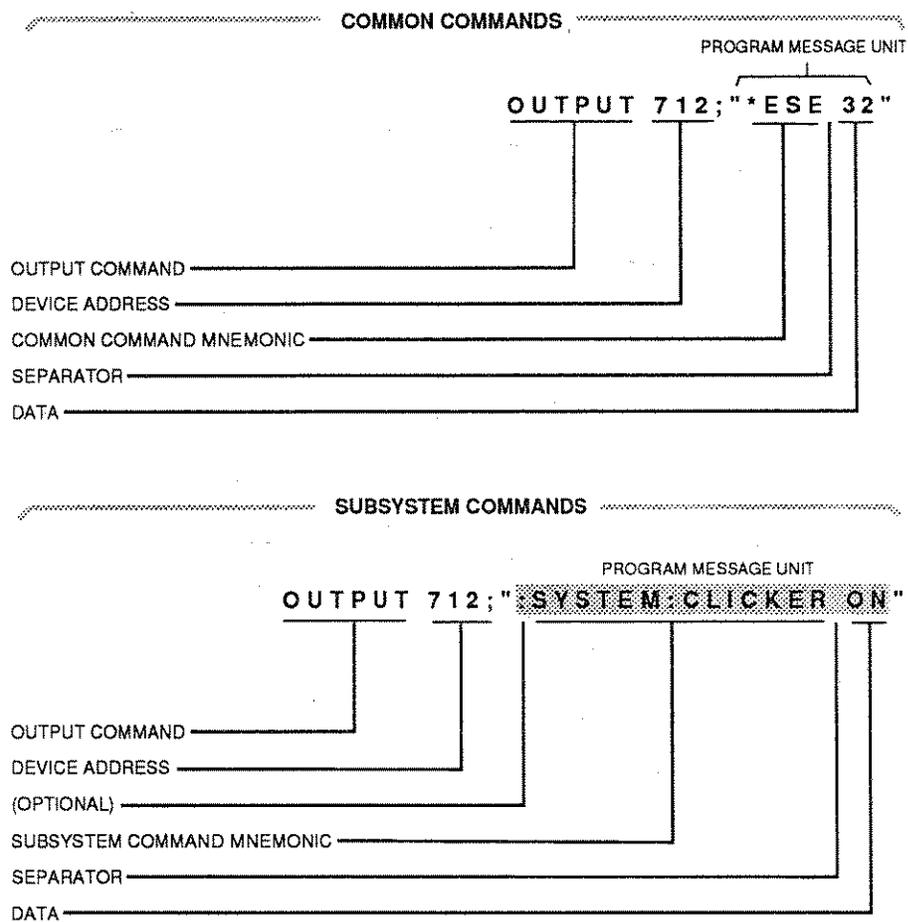
To program the Analyzer over the Hewlett-Packard Interface Bus (HP-IB), you must understand the command format and structure expected by the Analyzer. As defined in the SCPI standard, the Analyzer is remotely programmed with program messages. These messages are composed of sequences of program message units, with each unit representing a program command or query.

A program command or query is composed of functional elements that include separators, headers, program data, and terminators. These elements are sent to the Analyzer over the HP-IB as a sequence of ASCII data messages. *Figure 1-2* shows an example of a typical Common Command and an example of a typical subsystem command. For more information on optional mnemonics, parameter types, linking commands, and so on, refer to the “SCPI Summary” paragraph later in this chapter.

NOTE

See appendix C, SCPI CONCEPTS, for additional information on specific SCPI concepts and structures that apply to the Analyzer.

SCPI Commands



Note

The program examples in this manual assume the Analyzer is at factory set device address 712.

MPMS_V31

Figure 1-2. SCPI Command format

SCPI Conformance Information

SCPI Conformance Information

The SCPI commands used in the Analyzer are in conformance with the SCPI Standard Version 1990.0. The Analyzer's SCPI command set consists of the following:

- Common Commands as defined in IEEE 488.2-1987.
- Subsystem commands as confirmed (and listed) in the SCPI Standard. These are the commands defined in *table 1-1* as Std.
- Subsystem commands designed for the Analyzer in conformance with SCPI standards but not yet listed in the SCPI Standard. These are the commands defined in *table 1-1* as New.

Subsystem Commands Summary Table

Table 1-1, "Subsystem Commands Summary" lists the subsystem commands, their hierarchical relationship, related parameters (if any), and any associated information and comments.

The Subsystem Commands Summary Table is organized as follows:

- The **Keyword** column.
- The **Parameter Form** column.
- The **Std** column.
- The **Reset Values (*RST) Plus Comments** column.

NOTE

It is easier to understand the following explanations of the conventions used in the Subsystem Commands Summary Table if you remove a page from the table and compare the entries with the explanations.

Subsystem Commands Summary Table

Keyword Column

The **Keyword** column lists the keywords used to specify a command. Because SCPI commands have a hierarchical structure, the actual name of the command can consist of one or more keywords. This structure is also referred to as a tree system. In such a system, associated commands are grouped together under a subsystem (that can include additional subordinate subtrees) in the hierarchy.

Two optional levels that are always allowed in any :ARM or :TRIG path are

[:SEquence[1] | :START] [:LAYer]

These levels are not shown in the table because they are not required and add unnecessary complexity to the hierarchy.

To use a particular command, the full path to that command must be specified. This path is represented in the table by placing the highest level (subsystems) in the leftmost position. Lower levels (subtrees) in the hierarchy are indented to the right (within the subsystem).

The case of the letters in the keywords in the table is used to differentiate between the acceptable short form (ALL UPPER CASE LETTERS) and the acceptable long form (UPPER CASE and lower case letters). You can only use the exact characters shown for each of these two forms.

However, the Analyzer's parser is not case sensitive. That is, Rang2 (or RANge2) is acceptable for the keyword RANGe2 (but RAN2 is not acceptable). Sending a keyword that is not the short form, nor the complete long form, causes the Analyzer to generate an error.

Square brackets ([]) are used to enclose a keyword (or a numerical choice in a keyword) that is optional. That is, the Analyzer processes the command to have the same effect whether the optional keyword is used or not. Such a keyword is called a default or optional keyword. For example, to use the SENSE subsystem to specify that the maximum frequency expected on Input B is 20 MHz, you can use the following command:

Subsystem Commands Summary Table

:FREQuency:RANGe2 20 MHZ

instead of

:SENSe:FREQuency:RANGe2:UPPer 20 MHZ

because [:SENSe] and [:UPPer] are optional keywords.

The “|” is read as “or” and indicates that any of the choices shown can be used. For example, :RANGe[1|2|3] allows you to select channels A, B, or C. In other commands, the numerical choice can select other capabilities. For example, :MARKer:FREQuency:POSition[1|2] allows you to select between frequency markers 1 and 2. The explanation of the numerical choices is contained in the Comments column. If no numerical choice is specified, 1 is implied. The brackets around the numerical choice means that specifying a number is optional. If you do not specify a number, the default is 1 (for example :RANG and :RANG1 are equivalent).

Parameter Form Column

The **Parameter Form** column shows the parameters used in a command and their legal values. A command may allow the use of a SCPI-defined parameter type, a literal, or a combination of the two. The SCPI-defined parameters are distinguished by enclosing the type name in angle brackets (<>). A literal is a word that enumerates a setting that cannot be described using the SCPI-defined parameter types. For example, the AC|DC parameters in the INPut:COUPling AC|DC command are literal parameters.

Any command (and its associated query) that has a <numeric value> parameter, can use the literal parameters MINimum and MAXimum. When these literal parameters are used as a parameter for a command, they set the value to the minimum or maximum value currently allowed for that command. When they are used with the query form, they return the minimum or maximum value currently allowed for that command. In the query form, the “?” comes after the command header and before the literal parameter. For example, the following program:

Subsystem Commands Summary Table

```
10 OUTPUT 712;"EVEN:HYST:REL? MIN"  
20 ENTER 712;A$  
30 PRINT A$  
40 END
```

prints "0" on the display (0 is the minimum allowed value for this command).

Because using the MINimum and MAXimum parameter form causes the Analyzer to evaluate the range of values that a command can currently be set to, this capability reflects that some parameters have dynamic limits. That is, the limits of some parameters are dependent on the current state of the Analyzer.

NOTE

The program examples in this manual assume the Analyzer is at device address 712.

Std/New Column

The Std/New column gives the status of the command with respect to the SCPI standard. The "Std" commands operate as defined in the SCPI standard. The parts of these definitions that apply to the Analyzer are contained in this manual. The manual also contains any additional information that helps make the command easier to understand and use with the Analyzer. If you have learned how to use one of the standard SCPI commands on another instrument, you can use it in the same way on the Analyzer.

The category of "New" consists of commands that could be:

- SCPI approved but are not yet in the SCPI Manual.
- HP approved and submitted for SCPI approval.
- Not approved at all.

However, it is important to realize that the status of a new command has no affect on how it is used in the Analyzer. The new commands operate as defined in this manual.

Subsystem Commands Summary Table

Comments Plus Reset Values (*RST=) Column

The **Comments Plus Reset Values (*RST=)** column provides additional information about each command when required.

The ***RST=** value is the value of the command parameter immediately after the receipt of the ***RST** Common Command.

All of the Analyzer's commands, unless otherwise noted, have an additional query form. You generate the query form of a command by appending a question mark to the last keyword and omitting any parameters. (In most cases, a query returns the current status of the related command. Exceptions to this response are noted in the individual command descriptions.)

As already stated in the description of parameter types, any command that has a numeric value parameter can be queried for the **MINimum** or **MAXimum** allowed value. For example, if you want to know the **MAXimum** value allowed for the **TIME:POSition2** command, you can send the **"TIME:POSition2? MAXimum"** query. This capability is noted in the **Comments** for each applicable command.

Any command in the table that is shown with a **"?"** at the end, is a **"Query Only"** command. Not all commands have a query form. Unless a command is specified as **"No Query"** or **"Query Only"** in this column, it has both a command and a query form.

Some commands cause events and, generally, they cannot be queried. Most events have no corresponding setting. They cause something to happen in the Analyzer at a particular instant in time. For example, the **:INITiate:IMMEDIATE** command causes a single measurement to be initiated and there is no query form for **:INITiate:IMMEDIATE**. However, **:INITiate:CONTinuous ON** also causes an event (it initiates continuous measurements), but its status can be queried. That is, you can query it to determine whether or not continuous measurements are being made. Again, any exceptions to

Subsystem Commands Summary Table

Table 1-1. Subsystem Commands Summary (Continued)

| KEYWORD | PARAMETER FORM | STD/ NEW | COMMENTS, RELATED FIGURES, AND RESET VALUES (*RST=) |
|---|----------------|--|--|
| :CALCulate (Cont.) :ANALysis (Cont.) :PROBability [:STATe] | ON 1 OFF 0 | New New | Subtree. Use only with Histogram. See fig 1-4. Maximum of 4 analysis functions enabled at a time. *RST=OFF. |
| :RESult? :PTPeak [:STATe] | ON 1 OFF 0 | New New New | Query Only. STATe must be ON to query results. Subtree. Peak-To-Peak analysis function. See fig 1-4. Maximum of 4 analysis functions enabled at a time. *RST=OFF. |
| :RESult? :SDEVIation [:STATe] | ON 1 OFF 0 | New New New | Query Only. STATe must be ON to query results. Subtree. Standard Deviation analysis function. See fig 1-4. Maximum of 4 analysis functions enabled at a time. *RST=OFF. |
| :RESult? :DATA? | ON 1 OFF 0 | New Std | Query Only. STATe must be ON to query results. Query Only. Response is Histogram from vs Time when CALC:HIST ON and sensing vs Time. |
| :DIMension[1 2] :ENCode :NOValue? :FORMat? :LABel? :OFFSet? :SCALe? :SIZE? :TYPE? :HISTogram [:STATe] | ON 1 OFF 0 | New New New New New New New New New New New New | Subtree. 1=frequency/time-interval probability dimension (Y-axis). 2=frequency/time-interval bin dimension (X-axis). Subtree. Histogram from vs Time. Applies when sensing vs Time. ON= Display Histogram from vs Time. OFF= Display vs Time. See figs. 1-11 and 1-14. *RST=OFF. |
| :ACCumulate :SAMPles :ABOVe? :BELow? :IN? :IMMediate :PATH? | ON 1 OFF 0 | New New New New New Std Std | ON= accumulates if STAT ON or OFF. Coupled to :SENS:HIST:ACC. See figs. 1-11 and 1-14. *RST=OFF. Subtree. Histogram from vs Time samples. Query Only. Number of samples at or above RANG:UPP. Query Only. Number of samples at or below RANG:LOW. Query Only. Number of samples in histogram. Event; No Query. Recalculates without reacquiring data. Query Only. Couples to SENSE:FUNCTION. |
| :CLEar | | New | Event; No Query. Clears displayed data, invalidates data available. See fig. 1-3. |
| :CONFigure | | Std | Part of the AUTOSCALE GROUP (see table 1-2, "Autoscale Group Commands Summary"). |

Subsystem Commands Summary Table

Table 1-1. Subsystem Commands Summary (Continued)

| KEYWORD | PARAMETER FORM | STD/NEW | COMMENTS, RELATED FIGURES, AND RESET VALUES (*RST=) |
|--|--|--|--|
| :DIAGnostic :CALibration :INPut[1 2] :GAIN :AUTO :OFFSet :AUTO :INTerp :AUTO :STATus? | ONCE OFF ONCE OFF ONCE OFF | Std New New New New New New New New | Subsystem. Subtree. Subtree. 1= A and 2= B. Subtree. ONCE calibrates input gain. See fig. 1-17. *RST= OFF. Subtree. ONCE calibrates input offset. See fig. 1-17. *RST= OFF. Subtree. INTerp is an abbreviation for interpolator. ONCE calibrates interpolators. See fig 1-17. *RST= OFF. Query Only. Returns 0 for calibration completed successfully and 1 for calibration failed. |
| :DISPlay [:WINDow] :TRACe :CONNect :GRATICule :AXIS [:STATe] :FRAMe [:STATe] :GRID [:STATe] :PERsistence :PROBability [:SCALe] :AUTO :MAXimum :MINimum :SPACing :MENU [:NAME] | ON 1 OFF 0 ON 1 OFF 0 ON 1 OFF 0 ON 1 OFF 0 SINGle INFinite ON 1 OFF 0 <numeric value> [PCT] <numeric value> [PCT] LINear LOGarithmic STATus FUNCtion VERTical TIMEbase TRIGger DISPlay FHISTogram MARKers SAMPLing UTILity | Std Std Std New Std Std Std Std Std Std New New New New New New Std Std | Subsystem. Subtree. Subtree. ON= measured samples connected on display. See fig. 1-11. *RST= ON. Subtree. Setting any command to OFF, sets all OFF. Subtree. Only used with vs Time. ON couples FRAME ON and GRID OFF. See fig. 1-11. *RST= ON. Subtree. ON couples AXIS OFF and GRID OFF. See fig. 1-11. *RST= ON. Subtree. ON couples AXIS ON and FRAME ON. See fig. 1-11. *RST= OFF. Applies when :SENS:SWE:TIME:REP OFF. See fig. 1-11. *RST= SING. Subtree. Y-axis of Histogram display. Subtree. ON= display is scaled based on data instead of MIN and MAX. See fig. 1-11. *RST= ON. See Note 1. See fig. 1-11. *RST= 100. See Note 1. See fig. 1-11. *RST= 0. Type of spacing used on display for Histogram data (Y-axis). See fig. 1-11. *RST= LIN. Subtree. Event; no query. Displays selected menu. See fig. 1-3. |
| :FETCh | | Std | Part of the AUTOSCALE GROUP (see table 1-2, "Autoscale Group Commands Summary"). |

Subsystem Commands Summary Table

Table 1-1. Subsystem Commands Summary (Continued)

| KEYWORD | PARAMETER FORM | STD/NEW | COMMENTS, RELATED FIGURES, AND RESET VALUES (*RST=) |
|---------------------|---------------------------------------|---------|--|
| [[:SENSE] | | Std | Subsystem. |
| :DATA? | | New | Query Only. Response is vs Time or Fast Histogram data (uses :FORMat). |
| :DIMension[1 2] | | New | Subtree. DIM1= Y-axis. DIM2= X-axis. For vs Time: 1=frequency or TI, 2=time. For Histogram 1=frequency or TI probability, 2=frequency or TI bins. |
| :ENCode | | New | Subtree. |
| :NOValue? | | New | Query Only. |
| :FORMat? | | New | Query Only. |
| :LABel? | | New | Query Only. |
| :OFFSet? | | New | Query Only. |
| :SCALE? | | New | Query Only. |
| :SIZE? | | New | Query Only. |
| :TYPE? | | New | Query Only. |
| :EVENT[1 2] | | New | Subtree. 1= A and 2= B. |
| :HYSTeresis | | New | Subtree. |
| :RELative | <numeric value> [PCT] | New | 0 - 100%. See Note 1. See fig. 1-6. *RST= 0 |
| :LEVel | <numeric value> [V] | New | See Notes 1 and 2. See fig. 1-6. *RST= 0 V. |
| :AUTO | ONCE OFF | New | Event; with query. ONCE = find 50% voltage threshold of input signal. See fig. 1-6. *RST= OFF. |
| :FREQuency | | Std | Subtree. |
| :ARM | | New | Subtree. Frequency sampling. |
| :CFINterval[1 2 3] | <numeric value> [S] | New | Sampling interval at Center Frequency for ARM:SOUR AUTO. 1= A, 2=B, and 3= C. See Notes 1 and 2. See fig. 1-15. *RST= 5E-6. |
| :AUTO | ON 1 OFF 0 | New | ON couples interval to SWEep:TIME:SPAN. See fig. 1-15. *RST= ON. |
| :ECOunt[1 2]? | | New | Query only. Returns actual arm event count. 1= A, 2= B. See fig. 1-15. |
| :ECOunt[1 2] | | New | Subtree. |
| :MULTiplier | <numeric value> | New | Arm event count multiplier. Only applies when SOURCE LINK and frequency A or B is selected. 1= A. 2= B. See Note 1. See fig. 1-15. *RST= ECO1 is 2. ECO2 is 1. |
| :FCFINterval[1 2 3] | <numeric value> [S] | New | Sampling interval at Center Frequency for FREQ:ARM:SOUR FAST. See fig. 1-15. *RST= 5E-6. |
| :AUTO | ON 1 OFF 0 | New | ON couples interval to SWEep:TIME:SPAN. See fig. 1-15. *RST=ON. |
| :LEVel | <numeric value> [V] | New | External arm level. Only applies when SOURCE EXTERNAL, or SOURCE LINK and frequency C selected. See Note 1. See fig. 1-15. *RST= 1.5 V. |
| :LINK? | | New | Query Only. Returns INT, INT2, or EXT for frequency A, B, or C, respectively. |
| :SLOPe | POSitive NEGative | New | Ext Arm slope. Only applies when SOURCE EXTERNAL, or SOURCE LINK and frequency C selected. See fig. 1-15. *RST= POS. |
| :SOURce | AUTO EXTERNAL FAST LINK TIMer | New | LINK specifies sampling on edge of A or B (frequency A or B, respectively), or external (frequency C). See fig. 1-15. *RST= AUTO. |

Subsystem Commands Summary Table

Table 1-1. Subsystem Commands Summary (Continued)

| KEYWORD | PARAMETER FORM | STD/NEW | COMMENTS, RELATED FIGURES, AND RESET VALUES (*RST=) |
|--|--|------------|--|
| [:SENSE] (Cont.) :FREQuency (Cont.) :ARM (Cont.) :TIMer | <numeric value> [S] | New | Time sampling (in SOUR TIMER). See Note 1. See fig. 1-15. *RST= 50.0E-6 |
| :STATus? | | New | Returns a 0 or 1. 0 = Use selected time sampling. 1 = selected time sampling < approximate minimum sampling interval. |
| :RANGe[1 2 3] [:UPPer] | <numeric value> [HZ] | Std Std | Subtree. 1 = A, 2 = B, and 3 = C. See Note 1. See fig. 1-7. *RST= A is 125 MHZ, B is 70 MHZ, and C is 1.375 GHZ. |
| :AUTO | ONCE OFF | Std | Event with query. ONCE autoscales UPP and LOW for input signal. See fig. 1-7. *RST= OFF for all three ranges. |
| :CENTer :AUTO | ONCE OFF | New Std | Subtree. Event with query. ONCE autoscales UPP and LOW retaining difference. See fig. 1-7. *RST= OFF for all three ranges. |
| :LOWer | <numeric value> [HZ] | Std | See Note 1. See fig. 1-7. *RST= A is 75 MHZ, B is 30 MHZ, and C is 625 MHZ. |
| :FUNction | "FREQ[1 2 3]" "FREQ[1 2 3]: HIST" "TINT" "TINT:HIST" | Std | Only the short form of the sensor function strings (in quotes) are shown. 1= A, 2= B, and 3= C. HIST corresponds to Fast Histogram. See fig. 1-5. *RST= FREQ. |
| :HISTogram :ACCumulate | ON 1 OFF 0 | New New | Subtree. Fast Histogram. Coupled to :CALC:HIST:ACC. See fig. 1-14. *RST= OFF. |
| :ARM :LEVel | <numeric value> [V] | New New | Subtree. Event that starts Fast Histogram acquisition. External arm level (SOUR EXT required). See fig. 1-6. See Note 1. *RST= 1.5 V. |
| :SLOPe | POSitive NEGative | New | External arm slope (SOUR EXT required). See fig. 1-14. *RST= POS. |
| :SOURce :SAMPles | IMMEDIATE EXTERNAL | New New | IMM= as fast as possible. See fig. 1-14. *RST= IMM. Subtree. Fast Histogram number of samples. |
| :ABOVe? | | New | Query Only. Number of samples above RANG:UPP. |
| :BELow? | | New | Query Only. Number of samples below RANG:LOW. |
| :IN? | | New | Query Only. Number of samples in histogram. |
| :TRIGger :COUNT | <numeric value> | New New | Subtree. Number of frequency or T1 measurements required for Fast Histogram update. See Note 1. See fig. 1-14. *RST= 1000. |
| :ROSCillator :SOURce? | | New New | Subtree. Reference Oscillator. Query only because the Analyzer auto-detects the availability of an external reference oscillator. Returns INT, EXT, or ERR. |

Subsystem Commands Summary Table

Table 1-1. Subsystem Commands Summary (Continued)

| KEYWORD | PARAMETER FORM | STD/ NEW | COMMENTS, RELATED FIGURES, AND RESET VALUES (*RST=) |
|---|---|---|---|
| [:SENSe] (Cont.) :TINterval (Cont.) :ROUTE :SLOPe[1 2] :TYPE | COMMOn SEParate POSitive NEGative PLUS PMINus | New New New | See fig. 1-5. *RST= SEP. 1= A (TI start edge) and 2= B (TI stop edge). See fig. 1-5. *RST= POS. +TI (PLUS) or ± TI (PMIN). See fig. 1-5. *RST= PLUS. |
| :STATus :OPERation [:EVENT]? :CONDition? :ENABle :NTRansition :PTRansition :PRESet :QUESTionable [:EVENT]? :CONDition? :ENABle :FREQuency [:EVENT]? :CONDition? :ENABle :NTRansition :PTRansition :NTRansition :PTRansition | <NRf> <NRf> <NRf> <NRf> <NRf> <NRf> <NRf> <NRf> <NRf> <NRf> <NRf> | Std Std Std Std Std Std Std Std Std Std Std Std Std Std Std Std Std | Subsystem. Subtree. Query Only. Query Only. Negative transition. Positive transition. Event; No Query. Presets :STATus Enable and Transition registers. Subtree. Query Only. Query Only. Subtree. Query Only. Query Only. Negative transition. Positive transition. Negative transition. Positive transition. |
| :SYSTem :CLICkEr :ERRor? :KEY :LOG? :PRESet :PRINT? :SSAVer :VERSion? | ON 1 OFF 0 <numeric value> ON 1 OFF 0 | Std New Std New Std New New Std | Subsystem. Enables/disables audible front-panel key click. See fig. 1-16. *RST= ON Query Only. Responds with first (oldest) error in error queue. Simulates front-panel key press. See Notes 1 and 2. Query Only. Response is contents of key queue (empties queue). Event; No Query. Sets Analyzer to same state as front-panel Preset key. See fig. 1-3. Query Only. Puts PCL-formatted display image data in Output Queue. See fig. 1-3. Enables or disables screen saver capability. See fig. 1-3. *RST= OFF. Query Only. Returns 1990.0. |

Subsystem Commands Summary Table

Table 1-1. Subsystem Commands Summary (Continued)

| KEYWORD | PARAMETER FORM | STD/NEW | COMMENTS, RELATED FIGURES, AND RESET VALUES (*RST=) |
|----------------|------------------------|---------|--|
| :TRIGger | | Std | Subsystem. Used to synchronize vs Time measurements. |
| :AUTO | ON 1 OFF 0 | New | ON= auto trigger if necessary. OFF= wait indefinitely for trigger. See fig. 1-10. *RST= ON. |
| :FREquency | | New | Subtree. Applies when SOUR LINK and measuring frequency. |
| :EEMode[1 2 3] | AONLy TREference | New | External edge arm only or time reference. TRIG:SOUR EXT required. See fig 1-10. *RST= AONLy. |
| :LEVel[1 2 3] | <numeric value> [HZ] | New | Frequency-trigger value. 1= frequency on A, 2= frequency on B, and 3= frequency on C. See Notes 1 and 2. See fig. 1-10. *RST= A is 100 MHZ, B is 50 MHZ, and C is 1 GHZ. |
| :AUTO | ON 1 OFF 0 | New | ON couples LEV to center of FREQ:RANG. See fig. 1-10. *RST= ON. |
| :SLOPe | POSitive NEGative | New | Trigger on rising (POS) or falling (NEG) edge of frequency vs Time. See fig. 1-10. *RST= POS. |
| :HFReject | ON 1 OFF 0 | New | High Frequency Reject. See fig. 1-10. *RST= OFF. |
| :LEVel | <numeric value> [V] | New | External-trigger level (SOUR EXT). See Note 1. See fig. 1-10. *RST= 1.5 V. |
| :LINK? | | Std | Query Only. Returns string: "SENS:FUNC" |
| :SLOPe | POSitive NEGative | Std | External-trigger slope (SOUR EXT). See fig. 1-10. *RST= POS. |
| :SOURce | EXTernal LINK | Std | LINK= frequency or TI value triggering. See fig. 1-10. *RST= LINK. |
| :TINTerval | | New | Subtree. Applies when SOUR LINK and measuring TI. |
| :EEMode | AONLy TREference | New | External edge arm only or time reference. TRIG:SOUR EXT required. See fig. 1-10. *RST= AONLy |
| :LEVel | <numeric value> [S] | New | Time-interval-trigger value. See Notes 1 and 2. See fig. 1-10. *RST= 1E-3. |
| :AUTO | ON 1 OFF 0 | New | ON couples LEV to center of TINT:RANG. See fig. 1-10. *RST= ON. |
| :SLOPe | POSitive NEGative | New | Trigger on rising (POS) or falling (NEG) edge of TI vs Time. See fig. 1-10. *RST= POS. |

NOTE:

- Can use the MINimum or MAXimum literal parameters instead of <numeric value> [suffix].
- Acts as a command when a numeric value is specified but also acts as a subtree when a subordinate command is used. For example, :LEV acts as a subtree in :TRIG:FREQ:LEV:AUTO, but as a command in :TRIG:FREQ:LEV 5 MHZ.

Autoscale Group Commands Summary Table

Autoscale Group Commands Summary Table

The Autoscale Group Commands consist of the following subsystems:

- CONFigure Subsystem
- FETCh Subsystem
- MEASure Subsystem
- READ Subsystem

The column definitions for *table 1-2* are the same as for the Subsystem Commands Summary Table except there is no Parameter Form column. The SCPI Standard requires specific parameters for some of these commands and you can use them, but they are ignored by the Analyzer. However, the Analyzer works fine if you never use these parameters. Therefore, unless you have an explicit problem of program compatibility with existing code, there is no need to use them.

Table 1-2. AUTOSCALE GROUP Commands Summary

| KEYWORD | STD NEW | COMMENTS |
|-------------------|------------|--|
| :CONFigure | Std | Subsystem. Configure for specified measurement. |
| :HISTogram | New | Subtree. Configure for Fast Histogram measurement. |
| [:VOLTage] | New | Subtree. |
| :FREQUency[1 2 3] | New | 1 = A, 2 = B, and 3 = C. Configure for frequency Fast Histogram measurement. |
| :TINTerval | New | Configure for time interval Fast Histogram measurement. |
| :XTIME | New | Subtree. Configure for vs Time measurement. |
| [:VOLTage] | New | Subtree. |
| :FREQUency[1 2 3] | New | 1 = A, 2 = B, and 3 = C. Configure for frequency vs Time measurement. |
| :TINTerval | New | Configure for time interval vs Time measurement. |
| :CONFigure? | Std | Returns function configured by last :CONF or :MEAS command. |

Autoscale Group Commands Summary Table

Table 1-2. AUTOSCALE GROUP Commands Summary (Continued)

| KEYWORD | STD NEW | COMMENTS |
|---|--|---|
| :FETCh [:SCALar] [:VOLTage] :FREQuency :TINTerval :IMEan? :IMRate? :MAXimum? :MEAN? :MINimum? :MRATe? :PROBability? :PTPeak? :SDEVIation? :HISTogram [:VOLTage] :FREQuency? :TINTerval? :XTIME [:VOLTage] :FREQuency? :TINTerval? | Std Std Std New New New New New New New New New New New New New New New New New New New | Subsystem. Places measurement data in Output Queue. Subtree. Response is single numeric value in ASCII. Subtree. Subtree. Query Only. Inverse Mean analysis function. Query Only. Inverse Modulation Rate analysis function (vs Time required). Query Only. Query Only. Query Only. Query Only. Modulation Rate analysis function (vs Time required). Query Only. Histogram Probability analysis function. Query Only. Peak-To-Peak analysis function. Query Only. Standard Deviation analysis function. Subtree. Histogram from vs Time or Fast Histogram (depending on :CONF). Subtree. Frequency Histogram data (format determined by :FORMat). Time Interval Histogram data (format determined by :FORMat). Subtree. Vs Time measurement data fetched. Subtree. Frequency vs Time data (format determined by :FORMat). Time Interval vs Time data (format determined by :FORMat). |
| :FETCh? | Std | Last function fetched, read, configured, or measured is retrieved. |
| :MEASure :HISTogram [:VOLTage] :FREQuency[1 2 3]? :TINTerval? :XTIME [:VOLTage] :FREQuency[1 2 3]? :TINTerval? | Std New New New New New New New | Subsystem. Stops measuring, configures, makes single measurement, and places measurement data in the Output Queue. Subtree. Fast Histogram measurement. Subtree. 1= A, 2= B, and 3= C. Frequency Fast Histogram data (format determined by :FORMat). Time Interval Fast Histogram data (format determined by :FORMat). Subtree. Vs Time measurement. Subtree. 1= A, 2= B, and 3= C. Frequency vs Time data (format determined by :FORMat). Time Interval vs Time data (format determined by :FORMat). |

Autoscale Group Commands Summary Table

Table 1-2. AUTOSCALE GROUP Commands Summary (Continued)

| KEYWORD | STD NEW | COMMENTS |
|-------------------------|------------|--|
| :READ | Std | Subsystem. Stops measuring, makes single measurement, and places measurement data in the Output Queue. |
| [:SCALar] | Std | Subtree. Response is single numeric value in ASCII. |
| [:VOLTage] | Std | Subtree. |
| :FREQUency :TINTerval | New | Subtree. |
| :IMEan? | New | Query Only. Inverse Mean analysis function. |
| :IMRate? | New | Query Only. Inverse Modulation Rate analysis function (vs Time required). |
| :MAXimum? | New | Query Only. |
| :MEAN? | New | Query Only. |
| :MINimum? | New | Query Only. |
| :MRATE? | New | Query Only. Modulation Rate analysis function (vs Time required). |
| :PROBability? | New | Query Only. Histogram Probability analysis function. |
| :PTPeak? | New | Query Only. Peak-To-Peak analysis function. |
| :SDEVIation? | New | Query Only. Standard Deviation analysis function. |
| :HISTogram | New | Subtree. Histogram from vs Time or Fast Histogram (depending on :CONF). |
| [:VOLTage] | New | Subtree. |
| :FREQUency? | New | Frequency Histogram data (format determined by :FORMat). |
| :TINTerval? | New | Time Interval Histogram data (format determined by :FORMat). |
| :XTIME | New | Subtree. Vs Time measurement data fetched. |
| [:VOLTage] | New | Subtree. |
| :FREQUency? | New | Frequency vs Time data (format determined by :FORMat). |
| :TINTerval? | New | Time Interval vs Time data (format determined by :FORMat). |
| :READ? | Std | Last function fetched, read, configured, or measured is retrieved. |

Common Commands Summary Table

Common Commands Summary Table

The Common Commands Summary Table lists and briefly describes the common commands used with the Analyzer. Chapter 3, Common Commands Reference Dictionary, describes these commands in more detail.

Table 1-3. Common Commands Summary

| COMMAND | DESCRIPTION |
|--|--|
| *CAL? | Performs an internal interpolator self-calibration. |
| *CLS | Clears Status data structures (Event Registers and Error Queue). |
| *ESE <NRf> | Sets the Standard Event Status Enable Register. |
| *ESE? | Queries the Standard Event Status Enable Register. |
| *ESR? | Queries the Standard Event Status Register. |
| *IDN? | Queries the Analyzer identification. |
| *OPC | Causes Analyzer to set the operation complete bit in the Standard Event Status Register when all pending operations (see note) are finished. |
| *OPC? | Places an ASCII "1" in the Output Queue when all pending operations (see note) are completed. |
| *OPT? | Identifies the options installed in the Analyzer. |
| *RCL <NRf> | Restores state of the Analyzer from a copy stored in local non-volatile memory (0 through 9 are valid memory registers). |
| *RST | Resets the Analyzer to a known state. |
| *SAV <NRf> | Stores the current state of the Analyzer in local non-volatile memory (1 through 9 are valid memory registers). |
| *SRE <NRf> | Sets the Service Request Enable Register. |
| *SRE? | Queries the Service Request Enable Register. |
| *STB? | Queries the Status Byte and Master Summary Status bit. |
| *TRG | This trigger command is the device-specific analog of the IEEE 488.1 defined GET. It initiates a single measurement. |
| *TST? | Executes an internal self-test and reports the results. |
| *WAI | Makes Analyzer wait until all pending operations (see note) are complete before executing commands following *WAI command. |
| Note: Pending operations include measurements in progress. | |

Front Panel To SCPI Command Mapping

Because the SCPI commands must conform to the SCPI standard and the front-panel interface must be easy to use and organized for your convenience, the one-to-one relationship of front-panel keys and the SCPI commands is not always readily apparent. *Figures 1-3 through 1-17* provide maps of the front-panel keys to the related SCPI commands. In addition to these figures referencing the related SCPI command, *table 1-1* "Subsystem Command Summary" and each SCPI command description references the appropriate key and figure.

Front Panel To SCPI Command Mapping

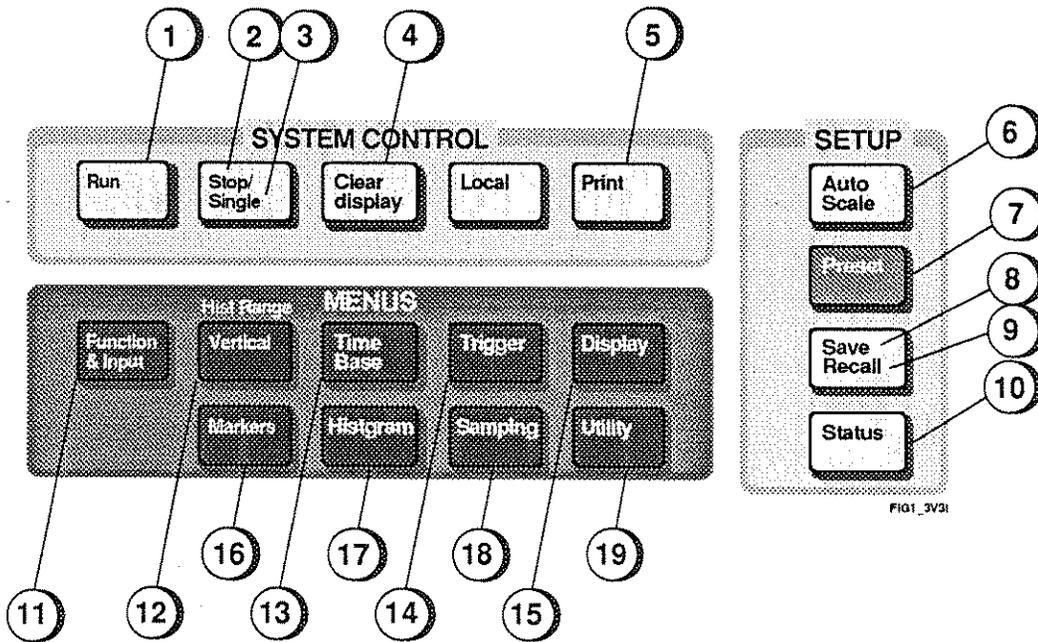


FIG1_3V3

NOTE

Individual menus can be displayed using the DISPlay:MENU command.

Figure 1-3. Front-panel Keys to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① :INITiate:CONTInuous ON
- ② :INITiate:CONTInuous OFF (if running) or :ABORt (if single measurement in progress)
- ③ :INITiate:IMMediate
- ④ :CLEar
- ⑤ :SYSTem:PRINt?
- ⑥ Autoscale Group (includes :CONFIgure, :FETCh, :MEASure, and :READ Subsystems)
- ⑦ :SYSTem:PRESet
- ⑧ *SAV <NRf>
- ⑨ *RCL <NRf>
- ⑩ Most queries
- ⑪ [:SENSe]:FUNctIon command and :INPUt Subsystem
- ⑫ [:SENSe]:FREQUency:RANGe[1|2|3] and [:SENSe]:TINTerval:RANGe Subtrees
- ⑬ [:SENSe]:SWEep:TIME Subtree
- ⑭ :TRIGger Subsystem
- ⑮ :DISPlay Subsystem and :CALCulate:HISTogram Subtree
- ⑯ :MARKer Subsystem
- ⑰ [:SENSe]:FUNctIon, [:SENSe]:HISTogram, and :CALCulate:HISTogram Subtrees
- ⑱ [:SENSe]:FREQUency:ARM and [:SENSe]:TINTerval:ARM Subtrees
- ⑲ :SYSTem Subsystem, :DIAGnostic:CALibration Subtree, and *CAL? and *TST? Common Commands

Figure 1-3. Front-panel Keys to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping

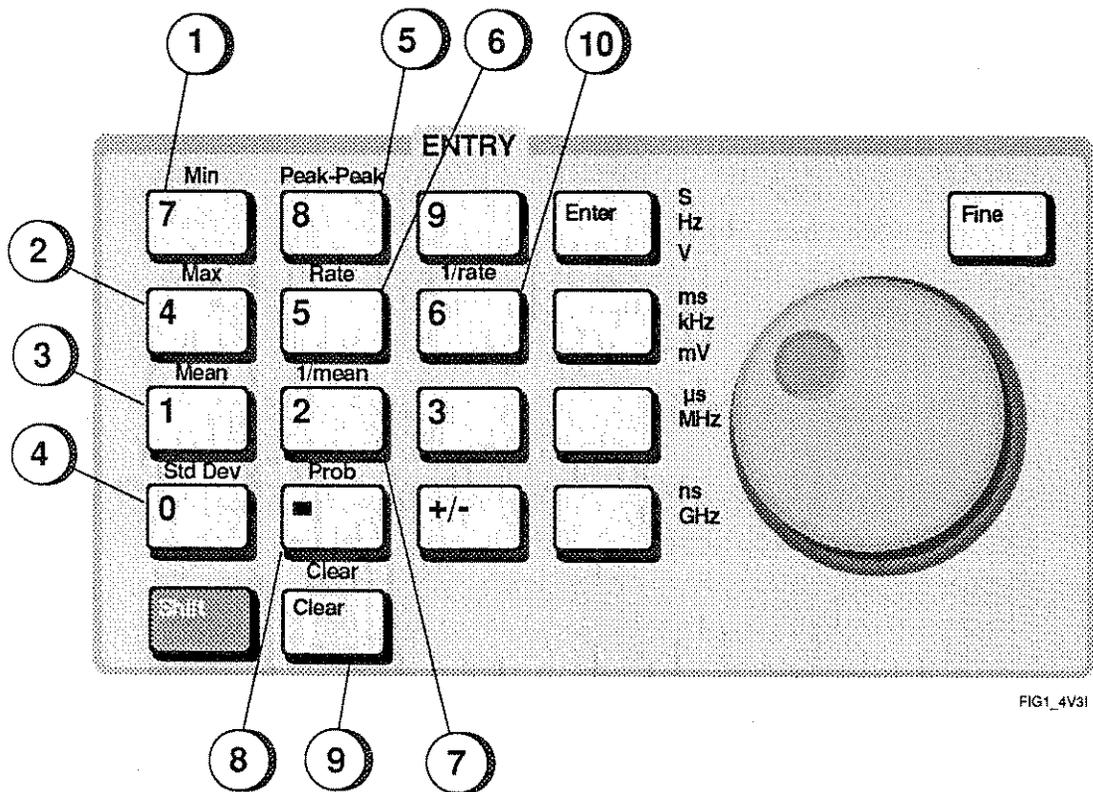


FIG1_4V31

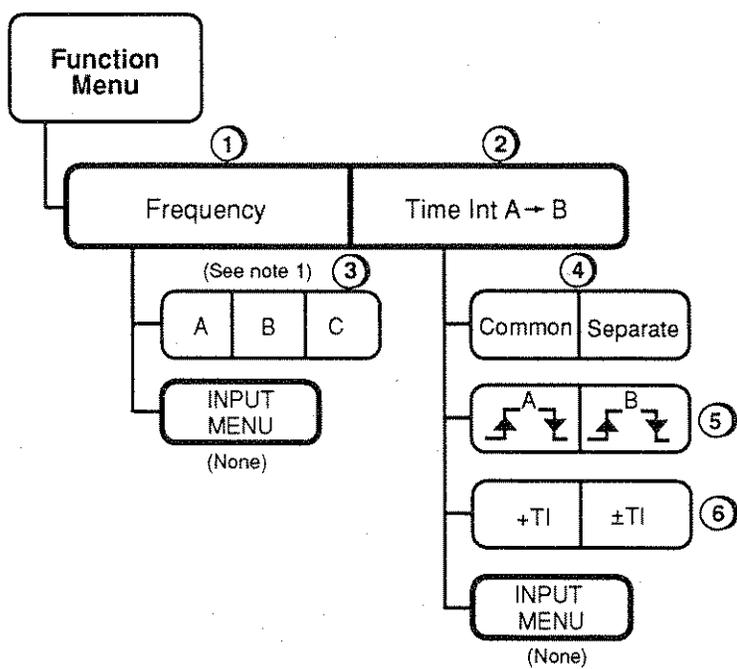
Figure 1-4. Analysis Keys to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① :CALCulate:ANALysis:MINimum Subtree or
:FETCh:FREQuency | :TINTerval:MINimum? or
:READ:FREQuency | :TINTerval:MINimum?
- ② :CALCulate:ANALysis:MAXimum Subtree or
:FETCh:FREQuency | :TINTerval:MAXimum? or
:READ:FREQuency | :TINTerval:MAXimum?
- ③ :CALCulate:ANALysis:MEAN Subtree or
:FETCh:FREQuency | :TINTerval:MEAN? or
:READ:FREQuency | :TINTerval:MEAN?
- ④ :CALCulate:ANALysis:SDEviation Subtree or
:FETCh:FREQuency | :TINTerval:SDEviation? or
:READ:FREQuency | :TINTerval:SDEviation?
- ⑤ :CALCulate:ANALysis:PTPeak Subtree or
:FETCh:FREQuency | :TINTerval:PTPeak? or
:READ:FREQuency | :TINTerval:PTPeak?
- ⑥ :CALCulate:ANALysis:MRATe Subtree or
:FETCh:FREQuency | :TINTerval:MRATe? or
:READ:FREQuency | :TINTerval:MRATe?
- ⑦ :CALCulate:ANALysis:IMEan Subtree or
:FETCh:FREQuency | :TINTerval:IMEan? or
:READ:FREQuency | :TINTerval:IMEan?
- ⑧ :CALCulate:ANALysis:PROBability Subtree or
:FETCh:FREQuency | :TINTerval:PROBability? or
:READ:FREQuency | :TINTerval:PROBability?
- ⑨ :CALCulate:ANALysis:AOFF
- ⑩ :CALCulate:ANALysis:IMRate Subtree or
:FETCh:FREQuency | :TINTerval:IMRate? or
:READ:FREQuency | :TINTerval:IMRate?

Figure 1-4. Analysis Keys to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping



Notes

1. Channel C appears only when Option 030 C channel is installed.

Figure 1-5. Function Menu to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① [:SENSe]:FUNCTION "FREQuency[1|2|3]" or
[:SENSe]:FUNCTION "FREQuency[1|2|3]:HISTogram"
- ② [:SENSe]:FUNCTION "TINTerval" or
[:SENSe]:FUNCTION "TINTerval:HISTogram"
- ③ [:SENSe]:FUNCTION "FREQuency[1|2|3]" or
[:SENSe]:FUNCTION "FREQuency[1|2|3]:HISTogram"
- ④ [:SENSe]:TINTerval:ROUTE COMMON|SEParate
- ⑤ [:SENSe]:TINTerval:SLOPe[1|2] POSitive|NEGative
- ⑥ [:SENSe]:TINTerval:TYPE PLUS|PMinus

Figure 1-5. Function Menu to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping

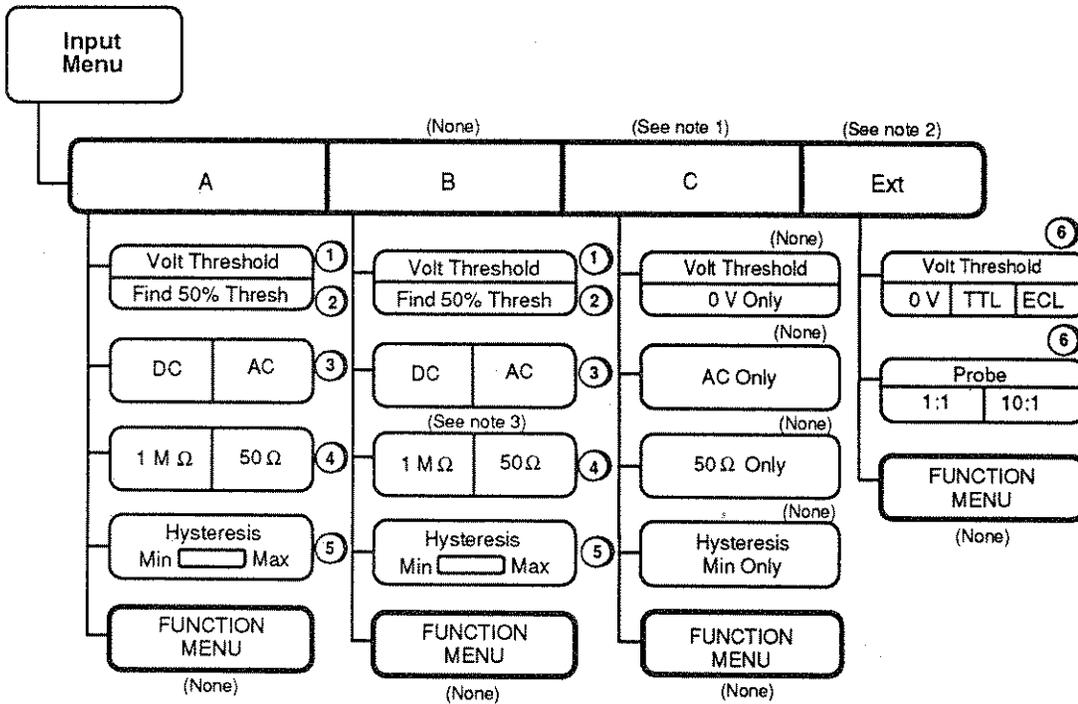


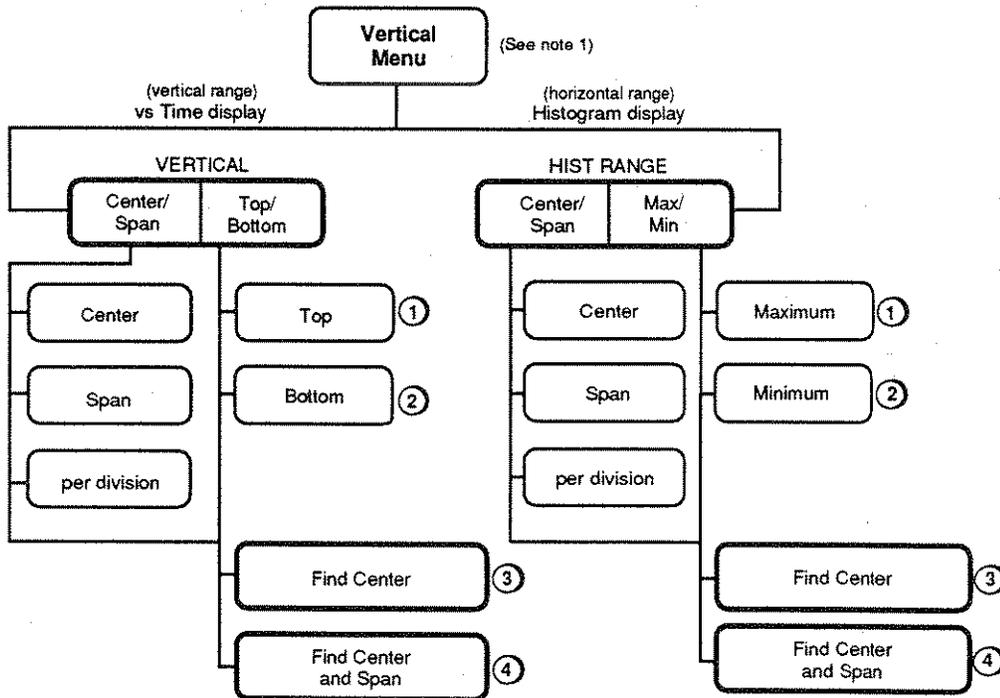
Figure 1-6. Input Menu to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① [:SENSe]:EVENT[1|2]:LEVel <numeric value>
- ② [:SENSe]:EVENT[1|2]:LEVel:AUTO ONCE
- ③ :INPut[1|2]:COUPling AC|DC
- ④ :INPut[1|2]:IMPedance <numeric value>
- ⑤ [:SENSe]:EVENT[1|2]:HYSTeresis:RELative <numeric value>
- ⑥ [:SENSe]:FREQUency:ARM:LEVel <numeric value> or
[:SENSe]:HISTogram:ARM:LEVel <numeric value> or
[:SENSe]:TINTerval:ARM:LEVel <numeric value> or
:TRIGger:LEVel <numeric value>

Figure 1-6. Input Menu to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping



Notes

- The Vertical Menu saves separate values for the following configurations:

Frequency A
 Frequency B
 Frequency C (Option 030 C channel)
 Time Int A→B

For each configuration, the VERTICAL and HIST RANGE values are shared.

For example: VERTICAL Center = HIST RANGE Center
 VERTICAL Top = HIST RANGE Max

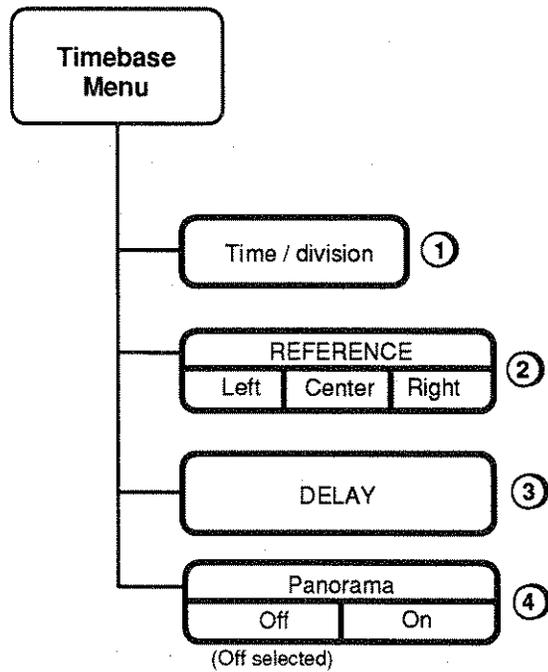
Figure 1-7. Vertical Menu to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① [:SENSe]:FREQuency:RANGe[1|2|3]:[UPPer] <numeric value> or
[:SENSe]:TINTerval:RANGe:[UPPer] <numeric value>
- ② [:SENSe]:FREQuency:RANGe[1|2|3]:LOWer <numeric value> or
[:SENSe]:TINTerval:RANGe:LOWer <numeric value>
- ③ [:SENSe]:FREQuency:RANGe[1|2|3]:CENTer:AUTO ONCE or
[:SENSe]:TINTerval:RANGe:CENTer:AUTO ONCE
- ④ [:SENSe]:FREQuency:RANGe[1|2|3]:AUTO ONCE or
[:SENSe]:TINTerval:RANGe:AUTO ONCE

Figure 1-7. Vertical Menu to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping



Note

1. The last LINK command used in effect for both TIME and PANorama commands.

Figure 1-8. Timebase Menu (Panorama Off) to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① [:SENSe]:SWEep:TIME:SPAN <numeric value>
- ② [:SENSe]:SWEep:TIME:DELay:LINK START|STOP|CENTer or
[:SENSe]:SWEep:TIME:PANorama:DELay:LINK START|STOP|CENTer
- ③ [:SENSe]:SWEep:TIME:DELay <numeric value>
- ④ [:SENSe]:SWEep:TIME:PANorama:STATe OFF

Figure 1-8. Timebase Menu (Panorama Off) to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping

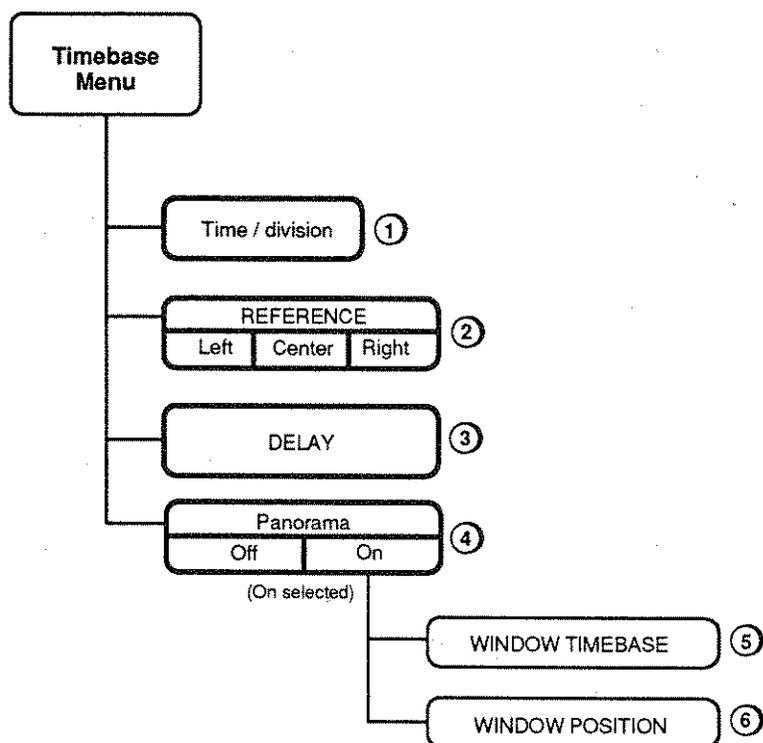


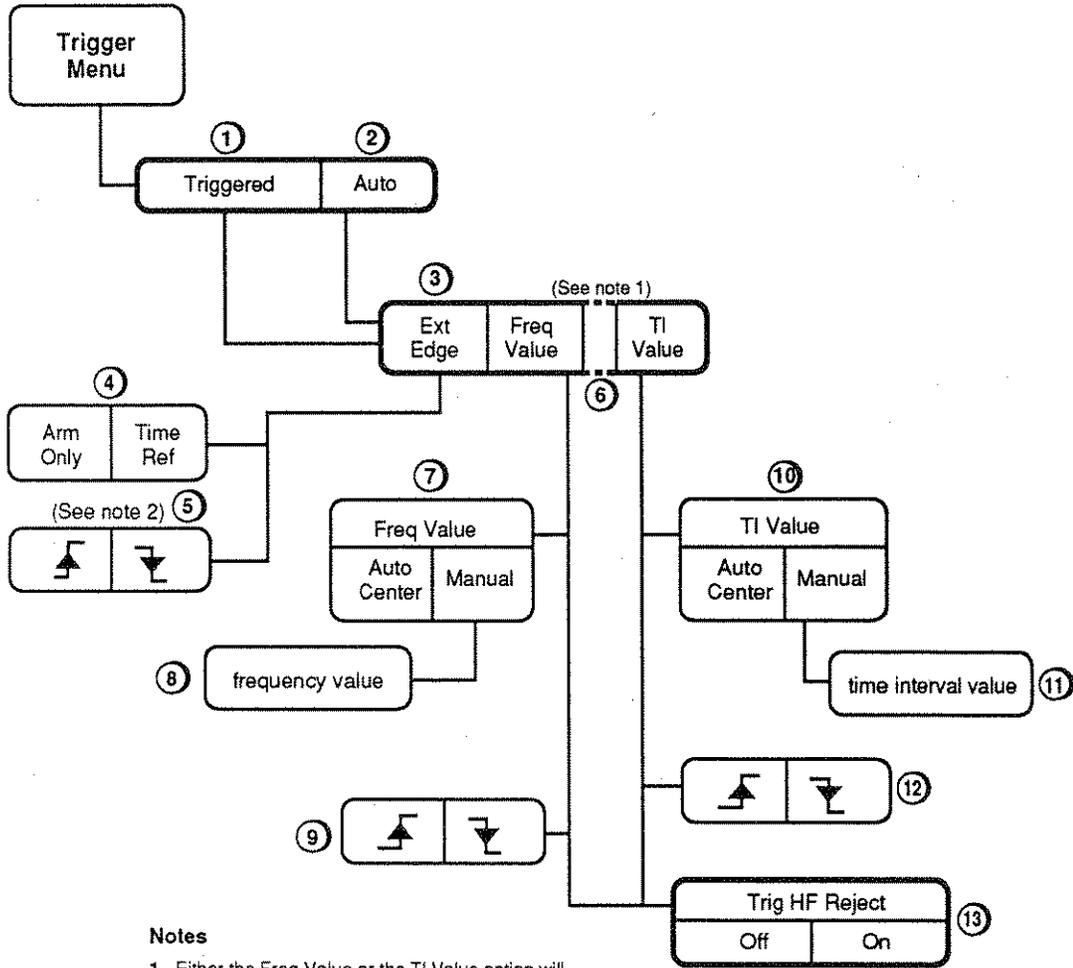
Figure 1-9. Timebase Menu (Panorama On) to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① [:SENSe]:SWEep:TIME:PANorama:SPAN <numeric value>
- ② [:SENSe]:SWEep:TIME:DELay:LINK START|STOP|CENTer or
[:SENSe]:SWEep:TIME:PANorama:DELay:LINK START|STOP|CENTer
- ③ [:SENSe]:SWEep:TIME:PANorama:DELay <numeric value>
- ④ [:SENSe]:SWEep:TIME:PANorama:STATe ON
- ⑤ [:SENSe]:SWEep:TIME:SPAN <numeric value>
- ⑥ [:SENSe]:SWEep:TIME:DELay <numeric value>

Figure 1-9. Timebase Menu (Panorama On) to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping



Notes

1. Either the Freq Value or the TI Value option will appear, depending on the selection of Frequency or Time Int A→B on the Function menu.
2. This slope setting is shared with the Fast Hist slope setting on the Histogram menu.

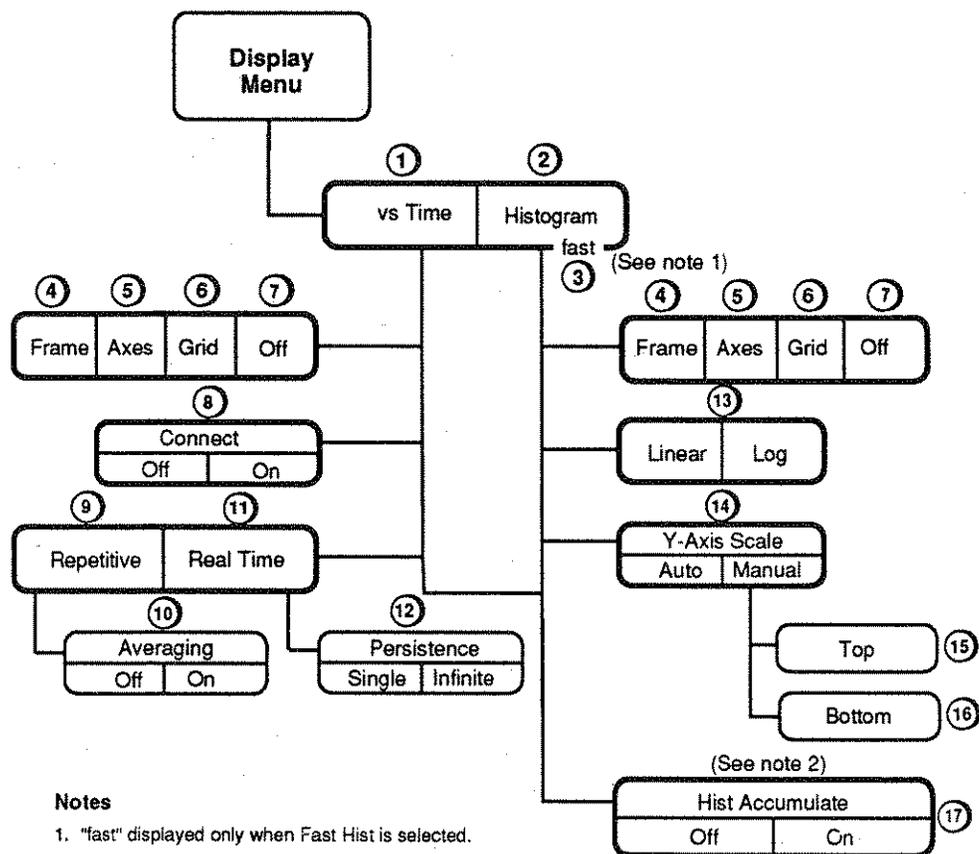
Figure 1-10. Trigger Menu to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① :TRIGger:AUTO OFF
- ② :TRIGger:AUTO ON
- ③ :TRIGger:SOURce EXTernal
- ④ :TRIGger:FREQuency:EEMode[1|2|3]AONLY|TREFerence
:TRIGger:TINTerval:EEMode AONLY|TREFerence
- ⑤ :TRIGger:SLOPe POSitive|NEGative
- ⑥ :TRIGger:SOURce LINK
- ⑦ :TRIGger:FREQuency:LEVel[1|2|3]:AUTO <Boolean>
- ⑧ :TRIGger:FREQuency:LEVel[1|2|3] <numeric value>
- ⑨ :TRIGger:FREQuency:SLOPe POSitive|NEGative
- ⑩ :TRIGger:TINTerval:LEVel:AUTO <Boolean>
- ⑪ :TRIGger:TINTerval:LEVel <numeric value>
- ⑫ :TRIGger:TINTerval:SLOPe POSitive|NEGative
- ⑬ :TRIGger:HFReject <Boolean>

Figure 1-10. Trigger Menu to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping



Notes

1. "fast" displayed only when Fast Hist is selected.
2. When Hist Accumulate is on, data is accumulated in the histogram of the vs Time even when the display is set to vs Time. This is not true for Fast Hist measurements. Data will accumulate for Fast Hist only when Fast Hist is selected. (See Histogram menu.)

Figure 1-11. Display Menu to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① [:SENSE]:FUNCTION "FREQUENCY[1|2|3]" or [:SENSE]:FUNCTION "TINTERVAL" and
:CALCULATE:HISTOGRAM OFF

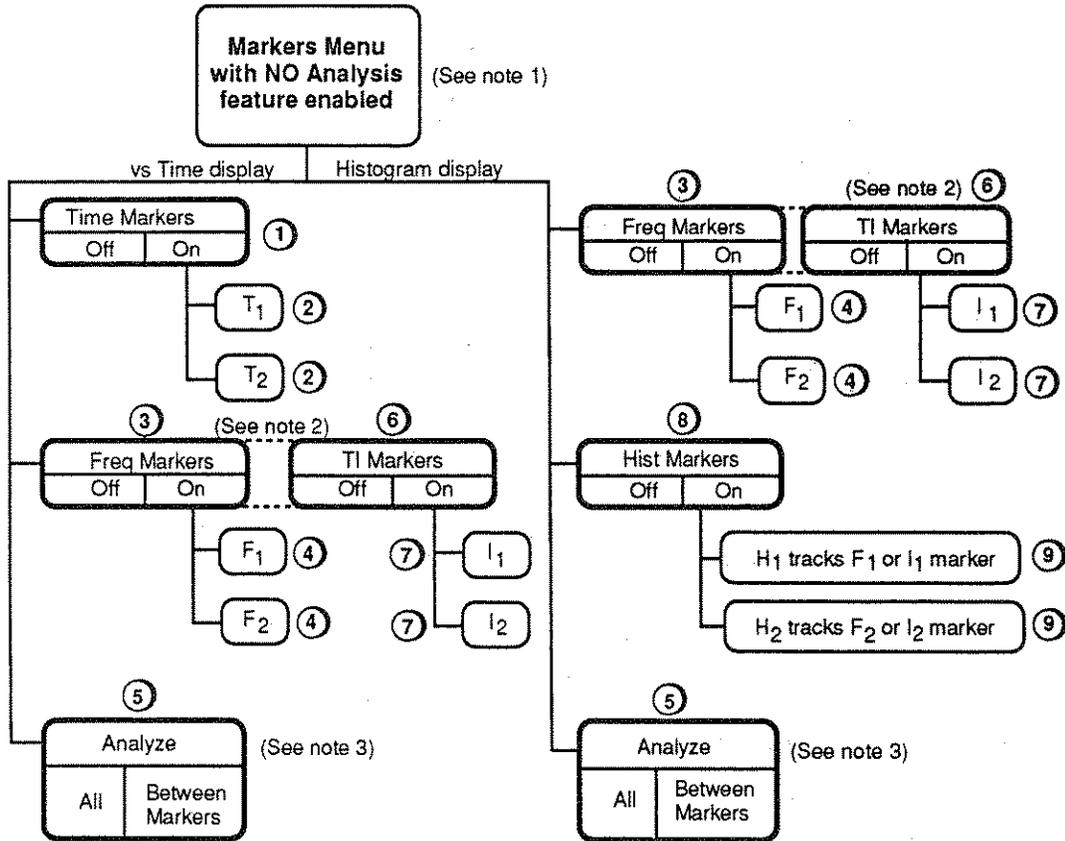
NOTE

The command set in 2 is for Histogram from vs Time and the command set in 3 is for Fast Histogram.

- ② [:SENSE]:FUNCTION "FREQUENCY[1|2|3]" or [:SENSE]:FUNCTION "TINTERVAL" and
:CALCULATE:HISTOGRAM ON
- ③ [:SENSE]:FUNCTION "FREQUENCY[1|2|3]:HISTOGRAM" or
[:SENSE]:FUNCTION "TINTERVAL:HISTOGRAM"
- ④ :DISPLAY:TRACE:GRATICULE:FRAME ON
- ⑤ :DISPLAY:TRACE:GRATICULE:AXIS ON
- ⑥ :DISPLAY:TRACE:GRATICULE:GRID ON
- ⑦ :DISPLAY:TRACE:GRATICULE:FRAME OFF or :DISPLAY:TRACE:GRATICULE:AXIS OFF or
:DISPLAY:TRACE:GRATICULE:GRID OFF
- ⑧ :DISPLAY:TRACE:CONNECT OFF|LINE <Boolean>
- ⑨ [:SENSE]:SWEPT:TIME:REPETITIVE ON
- ⑩ [:SENSE]:SWEPT:TIME:REPETITIVE:AVERAGE <Boolean>
- ⑪ [:SENSE]:SWEPT:TIME:REPETITIVE OFF
- ⑫ :DISPLAY:TRACE:PERSISTENCE SINGLE|INFINITE
- ⑬ :DISPLAY:TRACE:PROBABILITY:SPACING LINEAR|LOG
- ⑭ :DISPLAY:TRACE:PROBABILITY[:SCALE]:AUTO <Boolean>
- ⑮ :DISPLAY:TRACE:PROBABILITY[:SCALE]:MAXIMUM <numeric value>
- ⑯ :DISPLAY:TRACE:PROBABILITY[:SCALE]:MINIMUM <numeric value>
- ⑰ :CALCULATE:HISTOGRAM:ACCUMULATE <Boolean> or
[:SENSE]:HISTOGRAM:ACCUMULATE <Boolean>

Figure 1-11. Display Menu to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping



Notes

1. Analysis features are the shift-key features labeled in blue on the front panel.
2. Either Freq Markers or TI Markers option will appear depending on the selection of Frequency or Time Int A→B on the Function menu.
3. The Analyze option has no effect when there are no analysis features enabled.

Figure 1-12. Markers Menu (No Analysis Functions Enabled) to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① :MARKer:TIME <Boolean>
- ② :MARKer:TIME:POSition[1|2] <numeric value>
- ③ :MARKer:FREQuency <Boolean>
- ④ :MARKer:FREQuency:POSition[1|2] <numeric value>
- ⑤ :CALCulate:ANALysis:DELimit[1|2] ALL|MARKers
- ⑥ :MARKer:TINTerval <Boolean>
- ⑦ :MARKer:TINTerval:POSition[1|2] <numeric value>
- ⑧ :MARKer:PROBability <Boolean>
- ⑨ :MARKer:PROBability:POSition[1|2]?

Figure 1-12. Markers Menu (No Analysis Functions Enabled) to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping

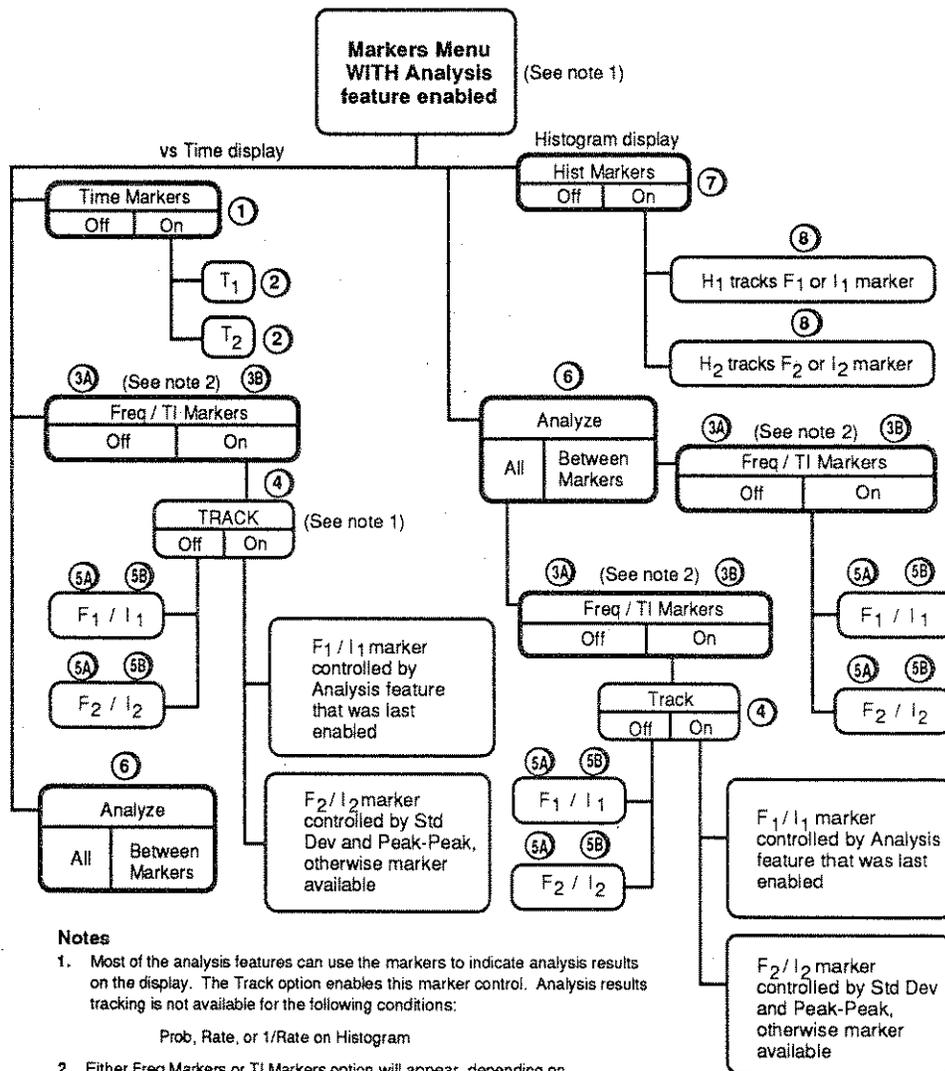


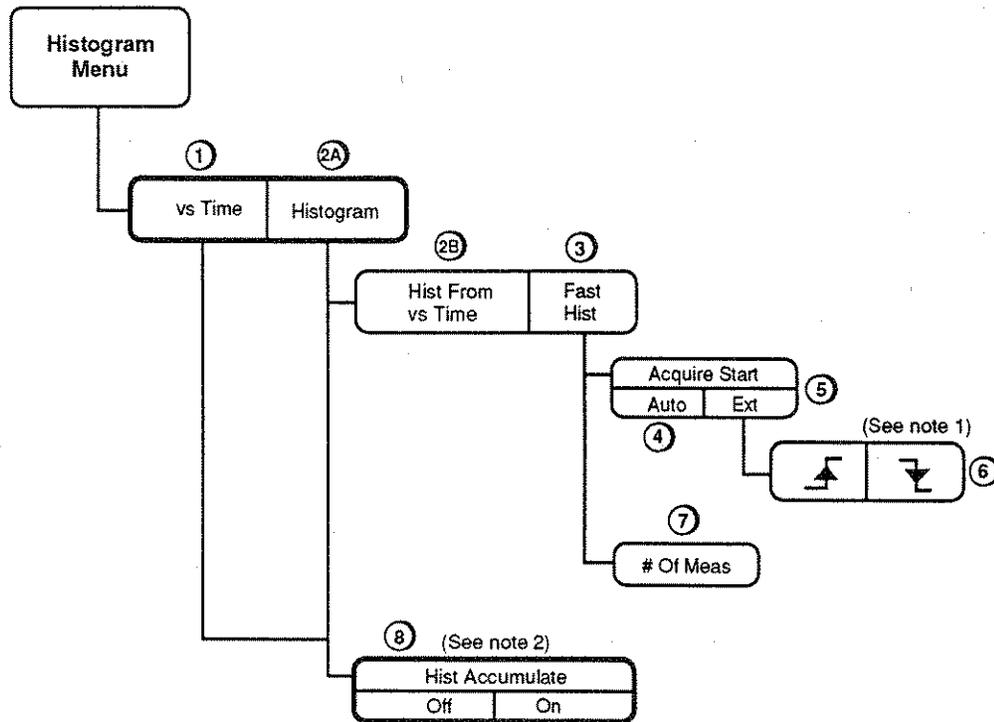
Figure 1-13. Markers Menu (Analysis Functions Enabled) to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① :MARKer:TIME <Boolean>
- ② :MARKer:TIME:POSition[1|2] <numeric value>
- ③A :MARKer:FREQuency <Boolean>
- ③B :MARKer:TINTerval <Boolean>
- ④ :MARKer:FREQuency:POSition[1|2]:AUTO <Boolean> or
:MARKer:TINTerval:POSition[1|2]:AUTO <Boolean>
- ⑤A :MARKer:FREQuency:POSition[1|2] <numeric value>
- ⑤B :MARKer:TINTerval:POSition[1|2] <numeric value>
- ⑥ :CALCulate:ANALysis:DELimit[1|2] ALL|MARKers
- ⑦ :MARKer:PROBability <Boolean>
- ⑧ :MARKer:PROBability:POSition[1|2]?

Figure 1-13. Markers Menu (Analysis Functions Enabled) to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping



Notes

1. This slope setting is shared with the Ext Edge setting on the Trigger menu.
2. When Hist Accumulate is on, data is accumulated in the histogram of the vs Time even when the display is set to vs Time. This is not true for Fast Hist measurements. Data will accumulate for Fast Hist only when Fast Hist is selected.

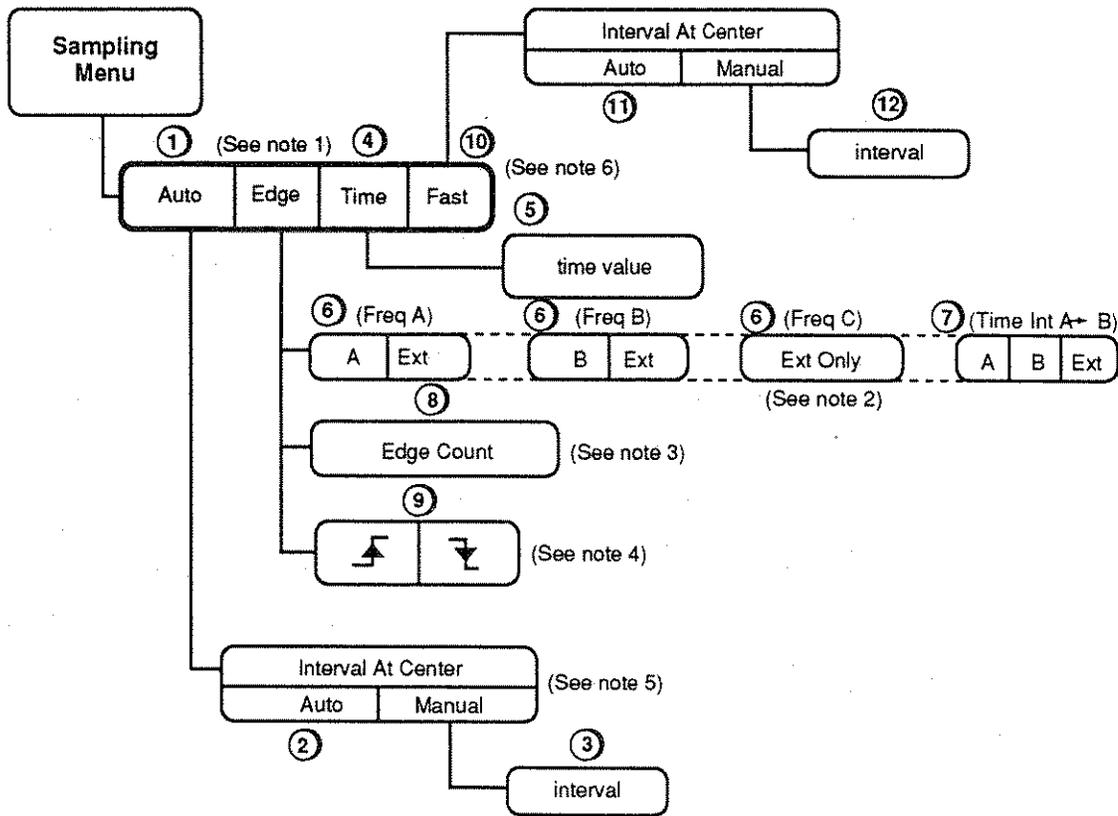
Figure 1-14. Histogram Menu to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① [:SENSe]:FUNCTION "FREQuency[1|2|3]" or
[:SENSe]:FUNCTion "TINTerval" and
:CALCulate:HISTogram OFF
- ②A :CALCulate:HISTogram ON
(This command selects the post-processing histogram)
- ②B [:SENSe]:FUNCTION "FREQuency[1|2|3]" or
[:SENSe]:FUNCTion "TINTerval" and
:CALCulate:HISTogram ON
- ③ [:SENSe]:FUNCTION "FREQuency[1|2|3]:HISTogram" or
[:SENSe]:FUNCTion "TINTerval:HISTogram"
(These commands selects the fast histogram)
- ④ [:SENSe]:HISTogram:ARM:SOURce IMMEDIATE
- ⑤ [:SENSe]:HISTogram:ARM:SOURce EXTERNAL
- ⑥ [:SENSe]:HISTogram:ARM:SLOPe POSitive|NEGative
- ⑦ [:SENSe]:HISTogram:TRIGger:COUNT <numeric value>
- ⑧ [:SENSe]:HISTogram:ACCumulate <Boolean>

Figure 1-14. Histogram Menu to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping



Notes

1. The Sampling type can be set separately for Frequency and Time Int A→B functions.
2. Measurement function and channel settings (on the Function menu) determine the edge sampling choices.
3. The edge count is fixed at 1 for Ext sampling. When Frequency is selected, the edge count value for channel A must be an even number. Channel B edge count can be an even or odd number. With Time Int A→B selected, the edge count for either channel can be even or odd.
4. Three slope settings can be used: one for Ext, one for channel A, and one for channel B. Slope settings for channel A and B can be specified only for Time Int A→B measurements.
5. This option appears only when Auto sampling is selected for a Frequency measurement.
6. Fast sampling appears only when Frequency is selected.

Figure 1-15. Sampling Menu to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① [:SENSe]:FREQuency:ARM:SOURce AUTO or
[:SENSe]:TINTerval:ARM:SOURce AUTO
- ② [:SENSe]:FREQuency:ARM:CFINterval[1|2|3]:AUTO <Boolean>
- ③ [:SENSe]:FREQuency:ARM:CFINterval[1|2|3] <numeric value>
- ④ [:SENSe]:FREQuency:ARM:SOURce TIMer or
[:SENSe]:TINTerval:ARM:SOURce TIMer
- ⑤ [:SENSe]:FREQuency:ARM:TIMer <numeric value> or
[:SENSe]:TINTerval:ARM:TIMer <numeric value>
- ⑥ [:SENSe]:FREQuency:ARM:SOURce LINK|EXTernal
- ⑦ [:SENSe]:TINTerval:ARM:SOURce INTernal[1|2]|EXTernal
- ⑧ [:SENSe]:FREQuency:ARM:ECOut[1|2]? and
[:SENSe]:FREQuency:ARM:ECOut[1|2]:MULTiplier <numeric value> or
[:SENSe]:TINTerval:ARM:ECOut <numeric value>
- ⑨ [:SENSe]:FREQuency:ARM:SLOPe POSitive|NEGative or
[:SENSe]:TINTerval:ARM:SLOPe[1|2|3] POSitive|NEGative
- ⑩ [:SENSe]:FREQuency:ARM:SOURce FAST
- ⑪ [:SENSe]:FREQuency:ARM:FCFINterval[1|2|3]:AUTO <Boolean>
- ⑫ [:SENSe]:FREQuency:ARM:FCFINterval[1|2|3] <numeric value>

Figure 1-15. Sampling Menu to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping

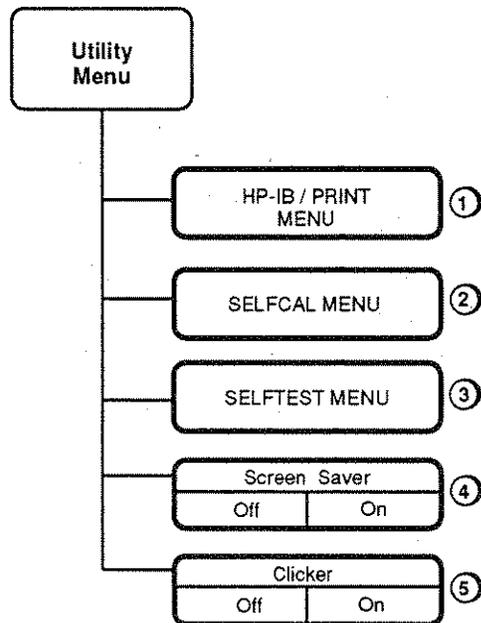


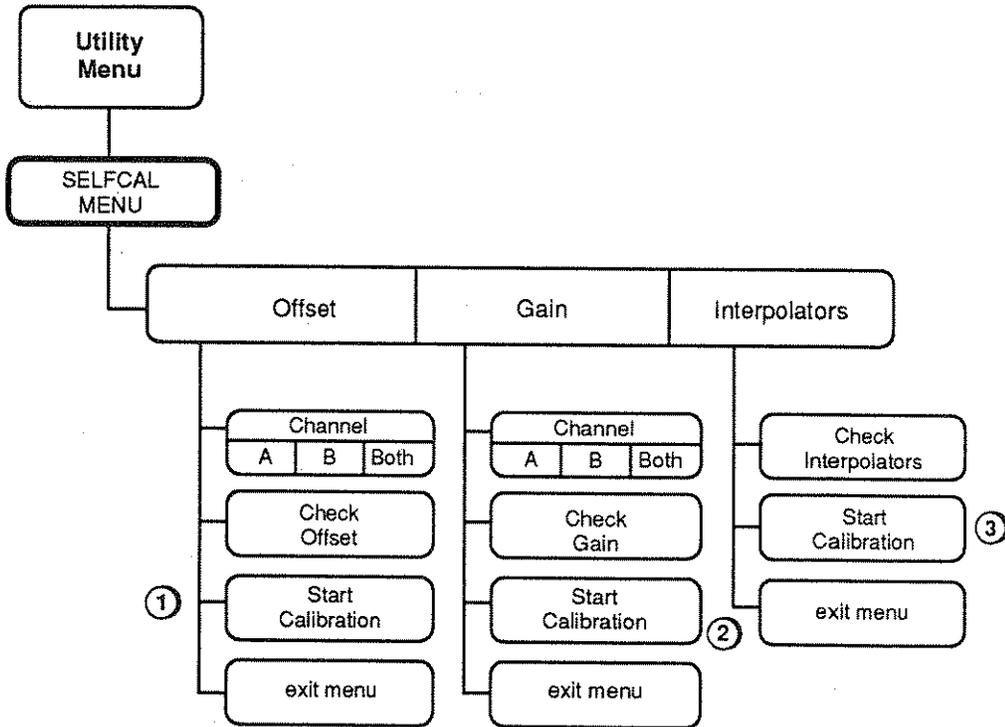
Figure 1-16. Utilities Menu to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① None
- ② :DIAGnostic:CALibration Subtree and *CAL? Common Command
- ③ *TST? Common Command
- ④ :SYSTEM:SSAVer <Boolean>
- ⑤ :SYSTEM:CLICker <Boolean>

Figure 1-16. Utility Menu to SCPI Command Map (Part 2 of 2)

Front Panel To SCPI Command Mapping



Note
A single SCPI command selects and starts the calibration sequences.

Figure 1-17. SELFCAL Menu to SCPI Command Map (Part 1 of 2)

Front Panel To SCPI Command Mapping

- ① :DIAGnostic:CALibration:INPut[1|2]:OFFSet:AUTO ONCE
- ② :DIAGnostic:CALibration:INPut[1|2]:GAIN:AUTO ONCE
- ③ :DIAGnostic:CALibration:INTerp:AUTO ONCE

Figure 1-17. Calibrate Menu to SCPI Command Map (Part 2 of 2)

Introduction To SCPI Programming

If you have programmed other instruments using HP-IB commands and programs, you will find that using the SCPI commands is very similar. All of the techniques you used before remain the same. The only difference is the hierarchy structure of the subsystem commands (and this type of structure has been used on other instruments previously). The best of these previous methods have been compiled into the SCPI standard.

If you are relatively new to using a programming language for instrument control, the remainder of this chapter contains some general information that you will find helpful in getting started. It contains brief descriptions of remote operation and initialization and a brief summary of the SCPI command syntax and parameter requirements. However, it is not intended to be a complete tutorial on how to use a specific language or to recommend any specific programming techniques.

For additional information on writing programs for instrument control, see appendix E, "Related Documentation."

Interfacing to an External Computer

Interfacing the Analyzer to an external computer consists of connecting the external computer and setting the HP-IB address for the Analyzer. See appendix A, HP-IB Interconnection, for additional information. This appendix also contains information on connecting a printer to the Analyzer.

NOTE

For the Analyzer to respond to your programs, the correct HP-IB address must be entered (12 is the factory set address) and the addressed (talk/listen) mode must be selected before you run your program. If you do not know how to set these conditions, see appendix A, HP-IB Interconnection.

Programming Languages

The example programs in this manual are shown in HP BASIC (5.0) language and assume the Analyzer is controlled from an HP 9000 Series 200/300 over the HP-IB. (However, there are four Microsoft QuickBASIC program examples in appendix d "Programming Examples.")

The ASSIGN, OUTPUT, and ENTER statements are defined in the following paragraphs.

NOTE

Additional information on the HP BASIC language can be found in the HP BASIC (5.0) manual set.

The ASSIGN Statement

Many of the programs in this manual contain an ASSIGN statement. ASSIGN opens an input/output (I/O) path between the computer and the Analyzer. I/O paths enable you to determine how the data that is transferred between the computer and the Analyzer is represented. Using an ASSIGN statement also allows the data to be transferred faster. This can be significant if a large amount of data is transferred. However, the ASSIGN statement (when specified without any attributes) assumes ASCII communication. If you are going to use the :FORMat BLOCk command, you must set the FORMAT attribute of the ASSIGN statement to OFF to read the data.

The following ASSIGN statement is used in the programs:

```
ASSIGN @Anlz TO 712
```

This statement opens a path @Anlz between the computer and the Analyzer on interface select code 7 at HP-IB address 12. This example assumes the Analyzer's HP-IB address is 12 and the computer is at select code 7.

NOTE

In short examples, when an ASSIGN statement is not used, the Analyzer's default address (712) is used instead of @Anlz.

The OUTPUT Statement

When using HP BASIC, a command is sent to the Analyzer with the OUTPUT statement.

```
OUTPUT @Anlz; "DISP:MENU FUNC"
```

The destination specified (@Anlz) is the I/O path opened by the ASSIGN statement. The command is enclosed between the quotation marks.

The ENTER Statement

When using HP BASIC, data is read from the Analyzer with the ENTER statement.

```
ENTER @Anlz;A$
```

The destination specified (@Anlz) is the I/O path opened by the ASSIGN statement. The data from the Analyzer is entered into a properly dimensioned string variable (A\$).

Initialization

To make sure the bus and all appropriate interfaces are in a known state, begin every program with initialization statements. For example, to initialize the interface of the Analyzer:

```
CLEAR 712
```

To set the Analyzer to its reset state:

```
OUTPUT 712;"*RST"
```

To initialize status reporting:

```
ASSIGN @Anlz TO 712  
!Assigns I/O path between computer and Analyzer  
OUTPUT @Anlz;"*CLS"  
!Clears all Event Registers and the Error Queue  
OUTPUT @Anlz;"*SRE 0"  
!Clears Service Request Enable register  
OUTPUT @Anlz;"*ESE 0"  
!Clears Event Status Enable register  
OUTPUT @Anlz;"STATUS:PRESet"  
!Presets Operation and Questionable enable registers  
and transition filters
```

NOTE

Refer to your controller manual and programming language reference manual for information on initializing your interface.

Remote Mode

Remote Mode

The Analyzer is in remote mode when **Addressed** is selected on the HP-IB/Print menu and the computer sets the REN (Remote Enable) line true. Or, in more general terms, when the computer is controlling the operation of the Analyzer.

Remote-Only Capabilities

Many Analyzer capabilities are available only in remote mode. These capabilities include status reporting, data transfer, etc.

Local Mode versus Remote Mode

In local mode, you are only allowed to enter valid parameters. The front panel displays a message if you try to enter data that is out of range. In remote mode, you must keep track of the required parameters to complete the measurement configuration. You can use the SYSTem:ERRor? query to determine if any errors have occurred.

Local-Only Capabilities

Most of the Analyzer capabilities that are available in local operation are also available for remote operation. The following capabilities are available only in local operation:

- Specifying HP-IB information such as HP-IB address, Addressed/Talk Only, etc.
- Power on and power off.

Remote-to-Local Changes

Pressing LOCAL on the front panel will return the Analyzer to local control, unless the Analyzer is in the Local Lockout state. No settings are changed. The Analyzer returns to the state defined by the commands issued to the Analyzer in remote mode.

Local Lockout

Local Lockout prevents the Analyzer from returning to local control from the front panel.

Example of setting Local Lockout:

```
LOCAL LOCKOUT 7
```

SCPI Summary

Local Lockout out is cleared by sending the Local message over the HP-IB.

Example of clearing Local Lockout:

```
LOCAL 7
```

NOTE

Refer to your controller manual and programming language reference manual for information on sending the Local and Local Lockout messages.

SCPI Summary

The following information is a brief summary of SCPI requirements. It provides a quick introduction to the command requirements. Additional information is contained in appendix C, SCPI Concepts and appendix E, Related Documentation.

Optional Keyword

Optional keywords are those which appear in square brackets ([]) in the command syntax. (Note that the brackets are not part of the command and are not sent to the Analyzer.) Suppose you send a second level command but do not send the preceding optional keywords. In this case, the Analyzer assumes you intend to use the optional keyword and it responds as if you had sent it. Examine the portion of the SENSE subsystem shown below:

```
[ :SENSE ]  
:EVENT  
    :HYSTERESIS  
        :RELATIVE <numeric value>
```

The root keyword :SENSE is an optional keyword. To set the Analyzer's hysteresis level to 50%, you can send either of the following command statements:

```
:SENS:EVENT:HYST:REL 50 or  
:EVENT:HYST:REL 50
```

SCPI Summary

Program Message Parameter Types

Table 1-4 contains explanations and examples of parameter types. Parameter types may be numeric value, Boolean, literal, string, or NRf.

Table 1-4. Program Message Parameter Types

| TYPE | EXPLANATIONS AND EXAMPLES |
|-----------------------|---|
| <numeric value> | Accepts all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation: 123, 123E2, -123, -1.23E2, .123, 1.23E-2, 1.23000E-01. Special cases include MiNimum and MAXimum (also DEFAult in the :CONFIgure and :MEASure Subsystems only). MiNimum selects minimum value available. MAXimum selects maximum value available. DEFAult selects the default value. Queries using MiNimum or MAXimum return the associated numeric value. |
| <Boolean> | Represents a single binary condition that is either true or false: 1 or ON, 0 or OFF (Query response returns only 1 or 0.) |
| <literal> | Selects from a finite number of choices. These parameters use mnemonics to represent each valid setting. An example is the INPut:COUPling AC DC command parameters (AC DC). |
| <NRf> | Flexible numeric representation. Only positive integers are used for NRf parameters in the Analyzer. |
| <string> | The <sensor function string> is a string parameter and is enclosed in double quotes. For example, OUTPUT 712;"FUNC ""FREQ"" is a valid program statement using a string parameter. |
| <non-decimal numeric> | Format for specifying hexadecimal (#H10FA), octal (#Q1077), and binary (#B10101011) numbers using ASCII characters. Used in :STATus subsystem commands. |

Optional Numeric Keyword Suffixes

Optional Numeric Keyword Suffixes are shown within square brackets ([]) (Note that the brackets are not part of the suffix and are not sent to the Analyzer.) If you do not specify a value for an optional suffix, the Analyzer assumes it is 1.

Query Parameters

All selectable <numeric value> parameters can be queried to return the minimum or maximum values they are capable of being set to by sending a MiNimum or MAXimum parameter after the "?". For example, consider the INPut:IMPedance? query.

If you send the query without specifying a parameter (INP:IMP?), the present impedance value is returned. If you send the MIN parameter (using INP:IMP? MIN), the command returns the minimum level currently available. If you send the MAX parameter, the command returns the maximum level currently available. Be sure to place a space between the question mark and the parameter.

Using Multiple Commands

When using IEEE 488.2 Common Commands with SCPI Commands on the same line, use a semicolon between the commands. For example:

```
*RST;INP:COUP AC
```

Multiple SCPI commands with corresponding parameters can also be sent at the same time. The first command is always referenced to the root node. Subsequent commands, separated by “;”, are referenced to the same level as the previous command if no “.” is present immediately after the command separator (the semicolon). For example, sending:

```
:INP:COUP AC;IMP 50
```

is equivalent to sending:

```
:INP:COUP AC  
:INP:IMP 50
```

OR

```
:INP:COUP AC;:INP:IMP 50
```

The “.” must be present to distinguish another root level command. For example:

```
INP:COUP AC;:INIT:CONT
```

is equivalent to sending:

```
INP:COUP AC  
INIT:CONT
```

SCPI Summary

If the ":" in front of INIT is omitted, the Analyzer assumes that the second command is "INP:COUP AC:INIT:CONT" and generates a syntax error.

See the syntax diagrams in chapters 2 and 3 for additional information on how to use the SCPI commands.

Suffixes

A suffix is the combination of suffix elements and multipliers that can be used to interpret the <numeric value> sent. If a suffix is not specified, the Analyzer assumes that <numeric value> is unscaled (that is, hertz, seconds, etc.)

For example the following two commands are equivalent:

```
OUTPUT 712;"FREQ:RANG:LOW 1 MHZ"
```

```
OUTPUT 712;"FREQ:RANG:LOW 1E+6"
```

Suffix Elements

Suffix elements, such as HZ (Hertz), V (volts), OHM (Ohms), PCT (percent), and S (seconds), are allowed within this format.

Suffix Multipliers

Table 1-5 lists the suffix multipliers that can be used with suffix elements.

How to Write SCPI Programs

Table 1-5. Suffix Multipliers

| Definition | Mnemonic | Name |
|------------|----------------------------|-------|
| 1E18 | EX | EXA |
| 1E15 | PE | PETA |
| 1E12 | T | TERA |
| 1E9 | G | GIGA |
| 1E6 | MA (or M for OHM and HZ)* | MEGA |
| 1E3 | K | KILO |
| 1E-3 | M (except for OHM and HZ)* | MILLI |
| 1E-6 | U | MICRO |
| 1E-9 | N | NANO |
| 1E-12 | P | PICO |
| 1E-15 | F | FEMTO |
| 1E-18 | A | ATTO |

*The suffix units, MHZ and MOHM, are special cases that should not be confused with <suffix multiplier>HZ and <suffix multiplier>OHM.

How to Write SCPI Programs

Figure 1-18 is a general summation of how to write SCPI programs. It shows a typical sequence you might go through in the process of writing a program. You do not have to follow this exact sequence, but it will help you to become familiar with the Analyzer's capabilities and the associated programming commands.

How to Write SCPI Programs

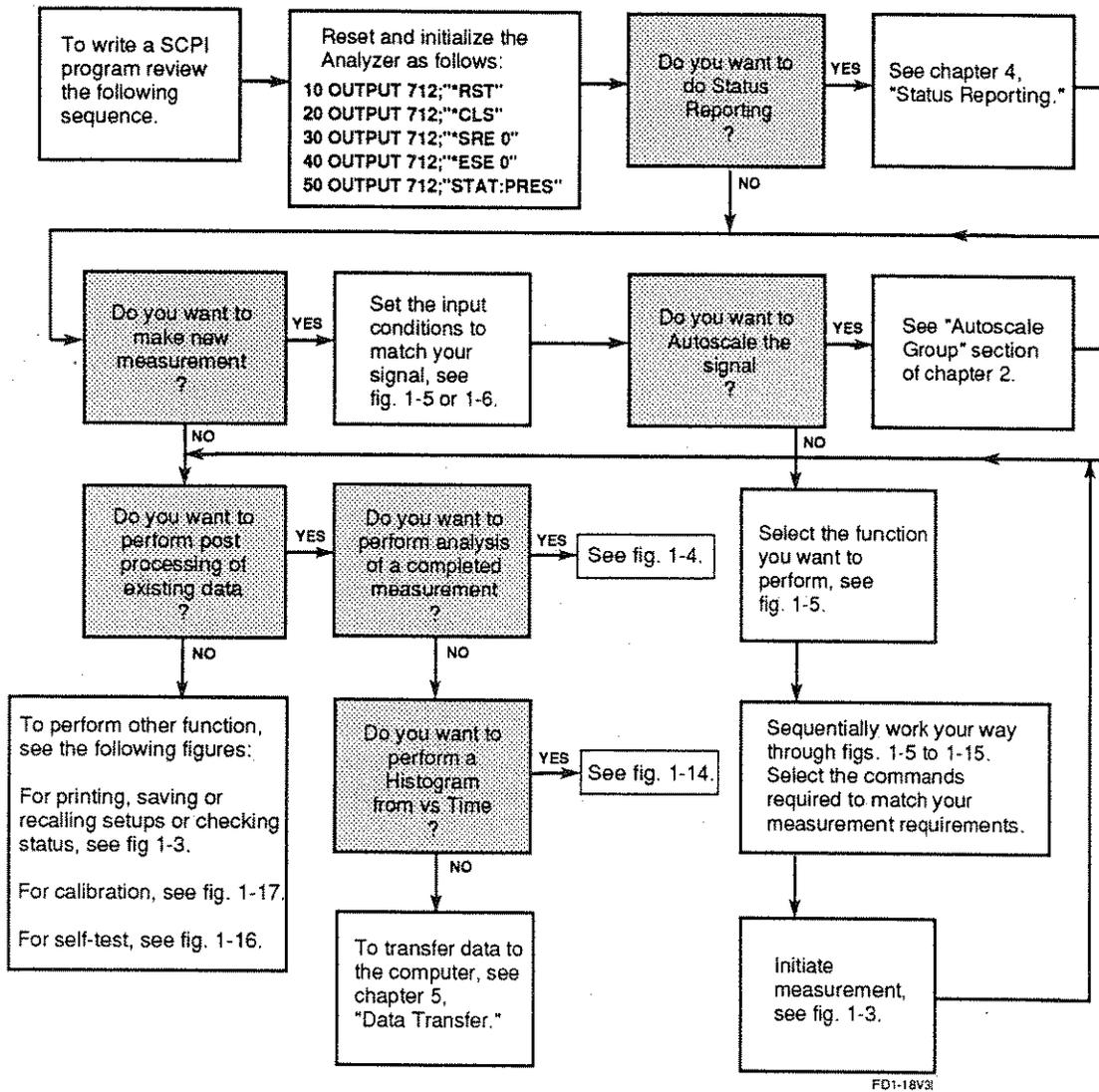


Figure 1-18. SCPI Programming Flowchart

SUBSYSTEM COMMANDS REFERENCE DICTIONARY

In This Chapter

This chapter describes the Standard Commands for Programmable Instruments (SCPI) subsystem commands as they apply to the HP 53310A Modulation Domain Analyzer (Analyzer). It contains descriptions and examples of the individual subsystems and commands. The subsystems are described in alphabetical order and contain syntax diagrams for all the commands in each subsystem. Individual commands within the subsystem are described in alphabetical order at each level of the hierarchy. (Any command in brackets is placed at the start of the hierarchy. For example, :DISPlay[:WINDow] is listed before :DISPlay:MENU.)

The subsystems and commands use upper-case letters only to indicate the short form for the command or parameter (for example, :CONF, :ConF, or :conf are all acceptable short forms for :CONFigure).

The only exception to this arrangement is the Autoscale Group. Because the Autoscale key is so important to local (front-panel) operation, the remote implementation of this capability is described separately in a section similar to the individual subsystems except that it has a green tab. The Autoscale Group uses the following subsystems:

- :CONFigure
- :FETCh
- :MEASure
- :READ

Subsystems

The tabbed sections are described in the following order:

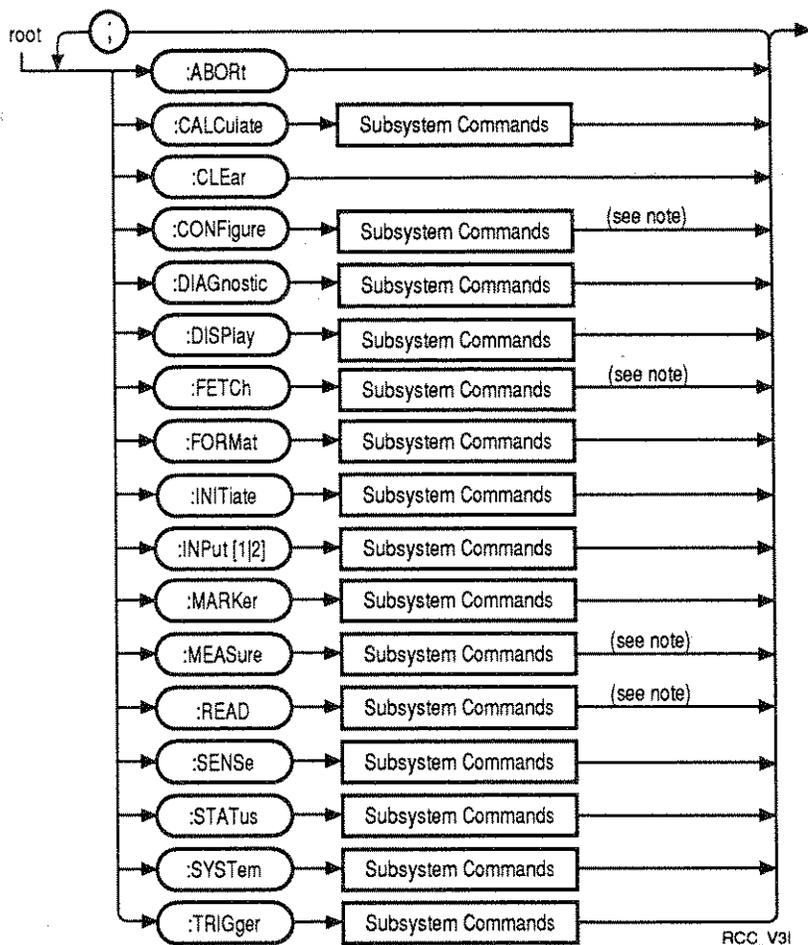
- :ABORt Subsystem
- Autoscale Group
- :CALCulate Subsystem
- :CLEAr Subsystem
- :DIAGnostic Subsystem
- :DISPlay Subsystem
- :FORMat Subsystem
- :INITiate Subsystem
- :INPut Subsystem
- :MARKer Subsystem
- [:SENSe] Subsystem
- :STATus Subsystem
- :SYSTem Subsystem
- :TRIGger Subsystem

Figure 2-1 is the top-level syntax diagram of the Subsystem Commands. The rectangles indicate that a more detailed drawing for this entry is contained later in this chapter.

NOTE

In the individual command descriptions, the headings are self-explanatory with the possible exception of "Related Front-panel Keys." The figures referenced under this heading are the front-panel key to SCPI command maps in chapter 1, "Programming Overview." These maps will probably be used most often to determine which SCPI command is related to a known front-panel key. However, it will sometimes help to clarify how to use a specific SCPI command if you know the related front-panel key.

Subsystems



Note:
These subsystem commands are part of the Autoscale Group
and are contained in the Autoscale Group section (Green Tab).

Figure 2-1. Subsystem Commands Syntax Chart

Subsystems

:ABORt Subsystem

The :ABORt subsystem contains a single event command that aborts any measurement in progress.

Subsystem Syntax *Figure 2-2 is a syntax diagram of the :ABORt subsystem.*

Example OUTPUT 712;":ABOR"

- Comments**
- The Pending Operation Flags that was set to true by the initiation of measuring is set to false by the :ABORt event command.
 - Event; No query.

Related Front-Panel Keys See *figure 1-3*.

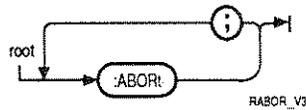


Figure 2-2. :ABORt Subsystem Command Syntax Chart

:ABORt Subsystem

Autoscale Group

The Autoscale Group consists of the following subsystems:

- :CONFigure
- :FETCh
- :MEASure
- :READ

The purpose of these subsystem commands is to acquire data using a set of high-level instructions. These commands are structured to allow you to trade off interchangeability with fine control of the measurement process. The :MEASure subsystem provides a complete capability where the Analyzer is configured, a measurement is taken, and the results are stored in the Output Queue in one operation.

When more precise control of the measurement is required, the :CONFigure and :READ subsystem commands can be used. The :CONFigure subsystem commands perform the configuration portion of the measurement. The :READ subsystem commands perform the data acquisition and post processing and then place the results in the Output Queue. This allows you to perform a generic configuration of the Analyzer using :CONFigure commands and then customize the measurement by changing particular Analyzer functions. The :READ subsystem commands complete the measurement process.

The :READ commands, in turn, are broken down into two additional commands :INITiate[:IMMEDIATE] and :FETCh?. :INITiate[:IMMEDIATE] performs the data acquisition. :FETCh? performs the post processing function and places the result in the Output Queue. This allows you to perform several different FETCh? functions on a single set of acquired data.

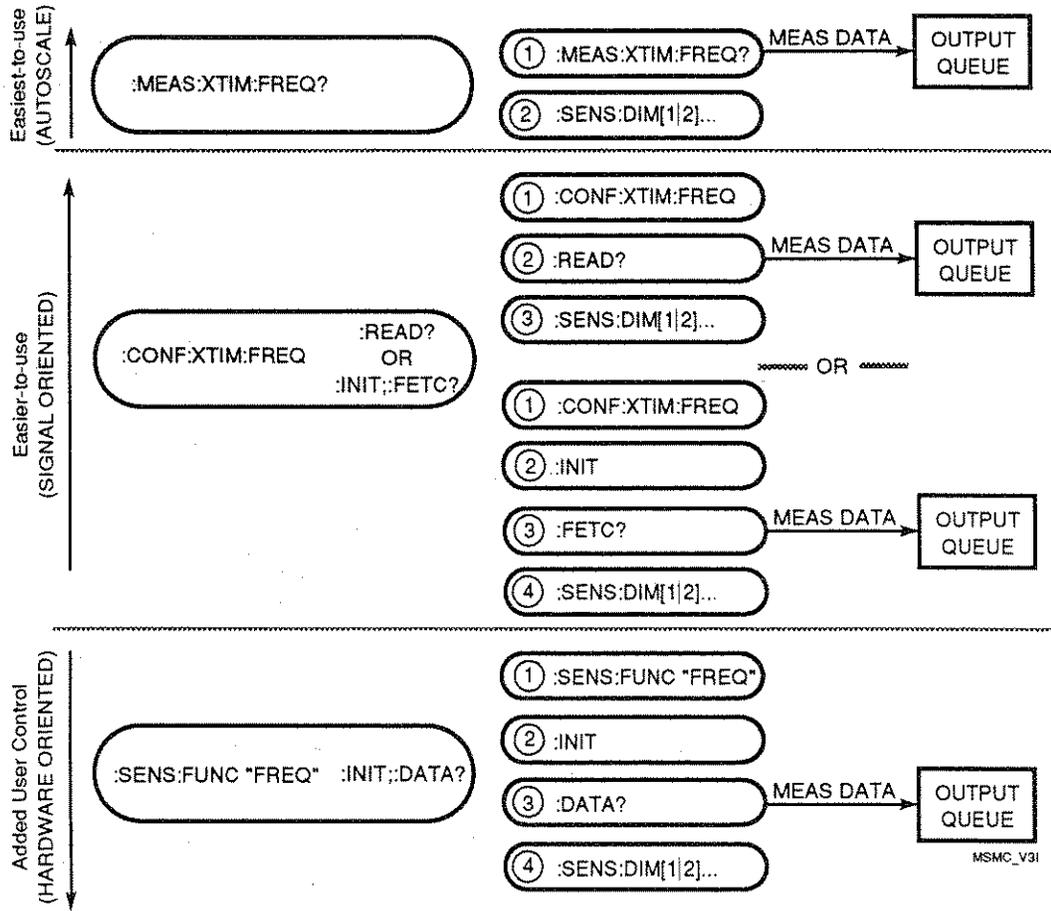


Figure 2-3. Autoscale vs :SENSe Measurements

Autoscale Group

For example, you can acquire measurement data that can yield different post-processing computations, such as standard deviation, mean, or histogram. Thus, a transient signal can be captured once using a :MEASure, :READ, or :INITiate subsystem command. A :FETCh subsystem command can then be used to obtain each of the different post-processed results without reacquiring a new measurement.

NOTE

Changing certain parts of the Analyzer's configuration invalidates existing measurement data. For example, the sequence:

:INIT;CONF:XTIM:FREQ:FETC?

causes an error to be generated because the data was invalidated by the :CONF command. Reconfiguring display or format commands does not cause an error.

Autoscale Vs :SENSE Measurements

The Analyzer offers you a wide range of choices in how to make measurements. You can specify measurements to meet your needs by using SCPI's three different measurement "levels" to select and perform a measurement. As shown in *figure 2-3*, you can use the :MEASure, :CONFigure, or :SENSE subsystems to make a frequency measurement. (*Figure 2-3* is only intended to show the relationship between these subsystems and does not show the full capability or requirements of the Analyzer.)

In comparing these subsystems with the front-panel operation, the use of :MEASure and :CONFigure subsystems is very similar to using the Autoscale key. Using the :SENSE subsystem is similar to using the front-panel keys and menus to setup and make a measurement.

Autoscale Group

Use :MEASure or :CONFigure to have the Analyzer make the first attempt at setting parameters such as vertical (frequency or time interval) range and the timebase parameters. This capability is especially useful if you are relatively new at operating the Analyzer.

Then, you can fine tune these parameters using the :SENSe subsystem to measure your signal exactly as you want.

NOTE

*Prior to using any of these commands, use the Common Command *RST to set the Analyzer to a known state. The reset conditions (*RST =) are listed in table 1-1 in chapter 1, "Programming Overview." It is also a good idea to initialize the Analyzer as shown in figure 1-18.*

:MEASure and :CONFigure Subsystem Commands (Autoscale)

The :MEASure and :CONFigure subsystem commands are input signal oriented. When using these commands, the Analyzer automatically sets required measurement parameters based on the input signal.

Although an extra command is required when using the :CONFigure subsystem commands, the Analyzer's actual setup changes little when using either :CONFigure or :MEASure subsystem commands.

:SENSe Subsystem Commands

The :SENSe subsystem commands let you specify measurement parameters if the :MEASure and :CONFigure selected values are not appropriate for your needs. When the measurement parameters are specifically selected to match your input signal's requirements, they can be made quicker and with more precise control. Use the :SENS:DATA? commands to return the measurement results and :SENS:DIM[1|2] subtree commands to interpret the data (see figure 2-3).

When the :SENSe subsystem commands are used, the Analyzer's ability to automatically set necessary measurement parameters is limited. Measurements are performed using the techniques you specify and all parameters are under your control. However, you can still select the AUTO options of individual commands.

Using These Commands

The :MEASure, :CONFigure, and :SENSe:FUNCTion can each set up any measurement. In all three cases, you must first configure the input signal parameters (that is, select Coupling, Impedance, etc.) The :MEASure or :CONFigure command then automatically selects the remaining parameters required to make the measurement. However, the :SENSe:FUNCTion command requires you to select the remaining parameters necessary to make the measurement.

Figure 2-3 also shows how the :MEASure, :CONFigure, and :SENSe commands differ in how they are used with the :READ?, :INITiate, :FETCh?, and :SENS:DATA? commands. The end result is the same in each case, the measurement is made and the results are stored in the Output Queue.

Before Using :MEASure or :CONFigure

You must configure the input conditions before using :MEASure or :CONFigure. That is, you must set the correct input conditions for the measurement you intend to select. These subsystems can operate on the following Analyzer configurations:

- Frequency measurements at channel A, B, or C (optional).
- Time Interval measurements, Channel A → B (separate or common inputs).
- Analyzer set to sensing vs Time (with or without post-processed histogram) or Fast Histogram.
- Value trigger set to: Auto Center or Manual.

Autoscale Group

Conditions Preset By :MEASure or :CONFigure

When :MEASure or :CONFigure commands are used the following conditions are preset:

- Main and panorama timebase delays are set to 0.
- Trigger is set to Auto-triggered; Trigger source is value-trigger.
- For frequency, sampling is set to Auto and interval at center frequency is set to Auto.
- For positive time interval, sampling is set to Auto.
- For plus/minus time interval, edge sampling on B with an edge count of 1 is set. The sampling slope of B is selected as a function of separate/common and the time interval start/stop edge slopes.

Conditions Set By :MEASure or :CONFigure

When :MEASure or :CONFigure commands are used the following conditions are set:

- Voltage threshold (:SENS:EVENT[1|2]:LEV) at 50% of the peak-to-peak voltage for the input signals being measured.
- Frequency or time interval measurement range (:SENS:FREQ:RANG[1|2|3] or :SENS:TINT:RANG).
- Main timebase (:SENS:SWE:TIME:SPAN).
- Value-trigger level (TRIG:FREQ:LEV or :TRIG:TINT:LEV).

Using :MEASure

Use the :MEASure subsystem commands to make a measurement after the input signal configuration is selected. No additional commands are needed to initiate the measurement and store the results. You can select and perform a measurement using one of the following commands:

```
:MEASure  
  :HISTogram  
    [:VOLTage]  
    :FREQuency[1|2|3]?  
    :TINTerval?  
  :XTIME  
    [:VOLTage]  
    :FREQuency[1|2|3]?  
    :TINTerval?
```

For example, executing the following command:

```
:MEAS:XTIM:FREQ2?
```

makes a Frequency vs Time measurement on input channel B and places the results in the Output Queue.

Use the :SENS:DIM[1|2] subtree commands to interpret the data returned by :MEASure.

If you want to transfer the measurement data from the Analyzer to the computer, see chapter 4, "Data Transfer" for additional information.

Using :CONFigure

Use the :CONFigure subsystem commands when you need to specify a measurement function prior to using :READ? or :INITiate and :FETCh?. The :CONFigure commands only set up the configuration specified by the command, they do not perform the measurement. You can select the configuration using one of the following commands:

```
:CONFigure  
  :HISTogram  
    [:VOLTage]  
    :FREQuency[1|2|3]  
    :TINTerval  
  :XTIME  
    [:VOLTage]  
    :FREQuency[1|2|3]  
    :TINTerval
```

For example, executing the following command:

```
:CONF:XTIM:FREQ2
```

configures a frequency measurement on input channel B. Executing a :READ?, or an :INITiate and :FETCh? performs the measurement and places the results in the Output Queue.

If you want to transfer the measurement data from the Analyzer to the computer, see chapter 4, "Data Transfer" for additional information.

Using :SENSe

Use the :SENSe subsystem commands when you need to configure not only the measurement function but also for special characteristics of the input signals. How to perform measurements using the :SENSe subsystem (and the remaining subsystems) is described in the rest of this manual.

NOTE

After you have configured a measurement using :SENSe subsystem commands, the :INITiate and :SENS:DATA? commands can be used to perform the measurement and place the data in the Output Queue.

:CONFigure Subsystem

This command configures the Analyzer to perform a specified measurement but does not perform the actual measurement. Use the INITiate and :FETCh? commands, or the :READ? command to initiate and read the measurement. Additional information on transferring data is contained in chapter 5, "Data Transfer."

Subsystem Syntax Figure 2-4 is a syntax diagram of the CONFigure subsystem.

Comments The :CONFigure? query (without any additional parameters) returns the setup configured by the :CONFigure command or :MEASure query.

If the Analyzer state has changed through receiving commands other than :CONF or :MEAS, the response reflects these changes.

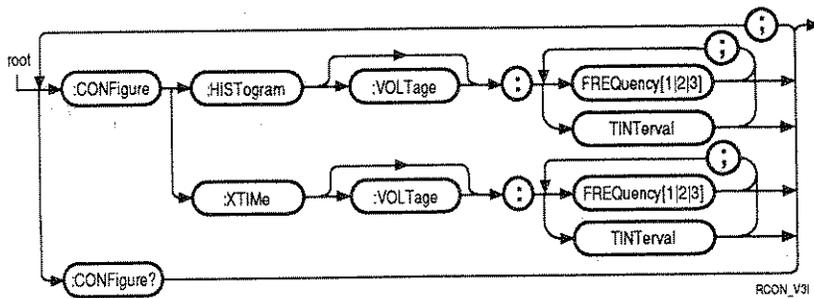


Figure 2-4. :CONFigure Subsystem Commands Syntax Chart

:CONFigure:HISTogram[:VOLTage]:FREQuency[1|2|3]

:CONFigure:HISTogram[:VOLTage]:FREQuency[1|2|3]

This command configures the Analyzer to perform a frequency Fast Histogram but does not perform the actual histogram. 1 = frequency on A, 2 = frequency on B, and 3 = frequency on C.

Use the :INITiate and :FETCh? commands, or the :READ? command to initiate and retrieve the measurement. Additional information on transferring data is contained in chapter 5, "Data Transfer."

Syntax :CONFigure:HISTogram[:VOLTage]:FREQuency[1|2|3]

Example OUTPUT 712;":CONF:HIST:FREQ2"

- Comments**
- When you use the :CONFigure commands, you must be sure that you use the appropriate :READ or :FETCh commands to obtain meaningful data. For example, if you use the example above to configure the Analyzer you must use the :READ?, :READ:HIST:FREQ?, :FETCh?, or :FETC:HIST:FREQ? query commands to return the data.
 - Use :SENS:DIM[1|2] subtree commands to interpret the data returned by these queries. Use :SENS:HIST:SAMP subtree commands to query number of samples in or out of the Fast Histogram.
 - Error -241 is generated when FREQ3 specified without the C channel option installed.
 - This command can also generate errors +2008, +2009, and +2010.

Related Front-Panel Keys See figure 1-3.

:CONFigure:HISTogram[:VOLTage]:TINTerval

:CONFigure:HISTogram[:VOLTage]:TINTerval

This command configures the Analyzer to perform a time interval Fast Histogram but does not perform the actual histogram.

Use the INITiate and :FETCh? commands, or the :READ? command to initiate and retrieve the measurement. Additional information on transferring data is contained in chapter 5, "Data Transfer."

Syntax :CONFigure:HISTogram[:VOLTage]:TINTerval

Example OUTPUT 712;":CONF:HIST:TINT"

- Comments**
- When you use the :CONFigure commands, you must be sure that you use the appropriate :READ or :FETCh commands to obtain meaningful data. For example, if you use the example above to configure the Analyzer you must use the :READ?, :READ:HIST:TINT?, :FETCh?, or :FETC:HIST:TINT? query commands to return the data.
 - Use :SENS:DIM[1|2] subtree commands to interpret the data returned by these queries. Use :SENS:HIST:SAMP subtree commands to query number of samples in or out of the Fast Histogram.
 - This command can also generate errors +2008, +2009, and +2010.

Related Front-Panel Keys See *figure 1-3*.

:CONFigure:XTIME[:VOLTage]:FREQuency[1|2|3]

:CONFigure:XTIME[:VOLTage]:FREQuency[1|2|3]

This command configures the Analyzer to perform a frequency vs Time measurement but does not perform the actual measurement. 1 = frequency on A, 2 = frequency on B, and 3 = frequency on C.

Use the INITiate and :FETCh? commands, or the :READ? command to initiate and retrieve the measurement. Additional information on transferring data is contained in chapter 5, "Data Transfer."

Syntax :CONFigure:XTIME[:VOLTage]:FREQuency[1|2|3]

Example OUTPUT 712;":CONF:XTIM:FREQ2"

- Comments**
- When you use the :CONFigure commands, you must be sure that you use the appropriate :READ or :FETCh commands to obtain meaningful data. For example, if you use the example above to configure the Analyzer you must use the :READ?, :READ:XTIM:FREQ?, :FETCh?, or :FETC:XTIM:FREQ? query commands to return the data.
 - Use :SENS:DIM[1|2] subtree commands to interpret the data returned by these queries.
 - Error -241 is generated when FREQ3 specified without the C channel option installed.
 - This command can also generate errors +2008, +2009, and +2010.

Related Front-Panel Keys See *figure 1-3*.

:CONFigure:XTIME[:VOLTage]:TINTerval

:CONFigure:XTIME[:VOLTage]:TINTerval

This command configures the Analyzer to perform a time interval vs Time measurement but does not perform the actual measurement.

Use the INITiate and :FETCh? commands, or the :READ? command to initiate and retrieve the measurement. Additional information on transferring data is contained in chapter 5, "Data Transfer."

Syntax :CONFigure:XTIME[:VOLTage]:TINTerval

Example OUTPUT 712;":CONF:XTIM:TINT"

- Comments**
- When you use the :CONFigure commands, you must be sure that you use the appropriate :READ or :FETCh commands to obtain meaningful data. For example, if you use the example above to configure the Analyzer you must use the :READ?, :READ:XTIM:TINT?, :FETCh?, or :FETC:XTIM:TINT? query commands to return the data.
 - Use :SENS:DIM[1|2] subtree commands to interpret the data returned by these queries.
 - This command can also generate errors +2008, +2009, and +2010.

Related Front-Panel Keys See *figure 1-3*.

:FETCh Subsystem

:FETCh Subsystem

The :FETCh subsystem commands retrieve the measurements taken by the :INITiate commands and place them in the Output Queue. By specifying a function as part of the command, the Analyzer retrieves the value of the function requested derived from the data taken by the last :INITiate command.

For example, even if a :CONF:XTIM:FREQ was the original function when the measurement was taken, functions that can be calculated from the stored data can be fetched.

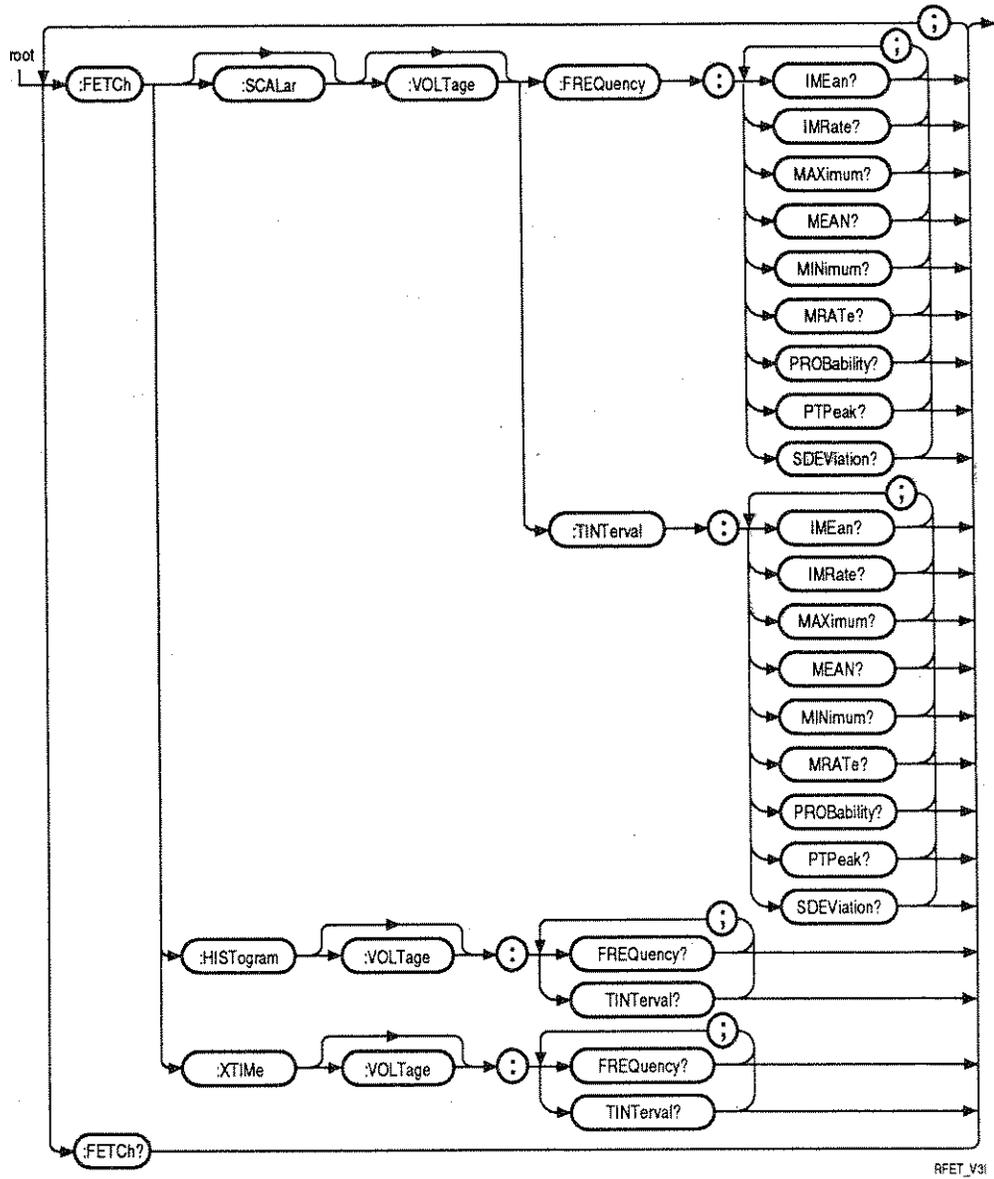
The :FETCh? subsystem commands do the following:

- The :FETCh:SCALar:VOLTage:FREQuency and :TINTerval subtree query commands cause the specified analysis function to be computed on the data stored on the last :INITiate command and the results placed in the Output Queue.
- The :FETCh:HISTogram and :XTIME subtree query commands retrieve the data stored in the Analyzer's memory by the last :INITiate command and places the data in the Output Queue. You must use the correct query for the stored data to obtain meaningful results.
- The :FETCh:HISTogram subtree query commands can be used to fetch Histogram from vs Time (when :CONF:XTIM: or :MEAS:XTIM) data or Fast Histogram data (when :CONF:HIST or :MESA:XTIM).
- The :FETCh? query command fetches the results of the last function used.

Subsystem Syntax *Figure 2-5 is a syntax diagram of the :FETCh subsystem.*

- Comments**
- The :FETCh commands have the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only

:FETCh Subsystem



RFET_V31

Figure 2-5. :FETCh Subsystem Commands Syntax Chart

:FETCh Subsystem

be canceled by the measurement completing, device clear, or power-on.

- For a description of the analysis functions, see the detailed description of the corresponding function in the :CALCulate Subsystem.
- The format of the returned data is determined by the :FORMat command, except for the analysis results which is always ASCII.
- Use :SENS:DIM[1 | 2] subtree commands to interpret sensed data (vs Time or Fast Histogram). Use :CALC:DIM[1 | 2] to interpret post-processed data (Histogram from vs Time).
- Use :SENS:HIST:SAMP subtree commands to query number of samples in or out of Fast Histogram. Use :CALC:HIST:SAMP subtree commands to query number of samples in or out of Histogram from vs Time.
- If no valid data is available, error -230 is generated.
- If requested function could not be computed, error +2011 is generated and 9.91E+37 is generated as the response.

:MEASure Subsystem

The:MEASure subsystem commands provide a complete measurement sequence, including configuration and reading of the data. It is used when an autoscaled measurement is acceptable and fine adjustment of the Analyzer is unnecessary. After making the measurement, the data is placed in the Output Queue.

Subsystem Syntax *Figure 2-6 is a syntax diagram of the MEASure subsystem.*

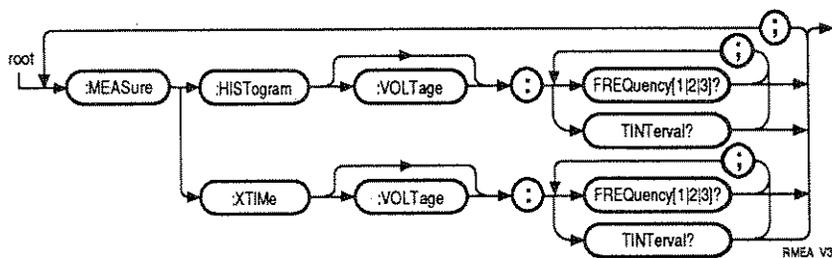


Figure 2-6. :MEASure Subsystem Commands Syntax Chart

:MEASure:HISTogram[:VOLTage]:FREQuency[1|2|3]?

:MEASure:HISTogram[:VOLTage]:FREQuency[1|2|3]?

This command configures the Analyzer to perform a frequency Fast Histogram and then performs the actual histogram. After making the measurement, the data is placed in the Output Queue. 1 = frequency on A, 2 = frequency on B, and 3 = frequency on C.

Syntax :MEASure:HISTogram[:VOLTage]:FREQuency[1|2|3]?

Example OUTPUT 712;":MEAS:HIST:FREQ?"

- Comments**
- The :MEASure commands have the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - The format of the returned data is determined by the :FORMat command.
 - Use :SENS:DIM[1|2] subtree commands to interpret the data returned by these queries. Use :SENS:HIST:SAMP subtree commands to query number of samples in or out of the Fast Histogram.
 - Error -241 is generated when FREQ3 specified without the C channel option installed.
 - If no valid data is available, error -230 is generated.
 - This command can also generate errors +2008, +2009, and +2010.

**Related
Front-Panel Keys**

See *figure 1-3*.

:MEASure:HISTogram[:VOLTage]:TINTerval?

:MEASure:HISTogram[:VOLTage]:TINTerval?

This command configures the Analyzer to perform a time interval Fast Histogram and then performs the actual Histogram measurement. After making the measurement, the data is placed in the Output Queue.

Syntax **:MEASure:HISTogram[:VOLTage]:TINTerval?**

Example **OUTPUT 712;":MEAS:HIST:TINT?"**

- Comments**
- The :MEASure commands have the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - The format of the returned data is determined by the :FORMat command.
 - Use :SENS:DIM[1|2] subtree commands to interpret the data returned by these queries. Use :SENS:HIST:SAMP subtree commands to query number of samples in or out of the Fast Histogram.
 - If no valid data is available, error -230 is generated.
 - This command can also generate errors +2008, +2009, and +2010.

Related Front-Panel Keys See *figure 1-3*.

:MEASure:XTIME[:VOLTage]:FREQuency[1|2|3]?

:MEASure:XTIME[:VOLTage]:FREQuency[1|2|3]?

This command configures the Analyzer to perform a frequency vs Time measurement and then performs the actual measurement. After making the measurement, the data is placed in the Output Queue. 1 = frequency on A, 2 = frequency on B, and 3 = frequency on C.

Syntax :MEASure:XTIME[:VOLTage]:FREQuency[1|2|3]?

Example OUTPUT 712;":MEAS:XTIM:FREQ?"

- Comments**
- The :MEASure commands have the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - The format of the returned data is determined by the :FORMat command.
 - Use :SENS:DIM[1|2] subtree commands to interpret the data returned by these queries.
 - Error -241 is generated when FREQ3 specified without the C channel option installed.
 - If no valid data is available, error -230 is generated.
 - This command can also generate errors +2008, +2009, and +2010.

Related Front-Panel Keys See *figure 1-3*.

:MEASure:XTIME[:VOLTage]:TINTerval?

:MEASure:XTIME[:VOLTage]:TINTerval?

This command configures the Analyzer to perform a time interval vs Time measurement and then performs the actual measurement. After making the measurement, the data is placed in the Output Queue.

Syntax **:MEASure:XTIME[:VOLTage]:TINTerval?**

Example **OUTPUT 712;":MEAS:XTIM:TINT?"**

- Comments**
- The :MEASure commands have the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - The format of the returned data is determined by the :FORMat command.
 - Use :SENS:DIM[1|2] subtree commands to interpret the data returned by these queries.
 - If no valid data is available, error -230 is generated.
 - This command can also generate errors +2008, +2009, and +2010.

Related Front-Panel Keys See *figure 1-3*.

:READ Subsystem

:READ Subsystem

Executing a :READ subsystem command initiates a measurement, possibly post-processes the measurement data, and places the result in the Output Queue.

The :READ subsystem commands do the following:

- The :READ:SCALar:VOLTage:FREQuency and :TINTerval subtree query commands cause a new measurement to be taken and the specified analysis function to be computed on the new data. The results are placed in the Output Queue.
- The :READ:HISTogram and :XTIME subtree query commands initiate a single measurement and place the data in the Output Queue. You must use the correct query for the stored data to obtain meaningful results.
- The :READ:HISTogram subtree query commands can be used to retrieve Histogram from vs Time data or Fast Histogram data.
- The :READ? query command retrieves the results of the last function used.

Subsystem Syntax *Figure 2-7 is a syntax diagram of the READ subsystem.*

Example OUTPUT 712;":READ:XTIM:FREQ?"

- Comments**
- The :READ commands have the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.

:READ Subsystem

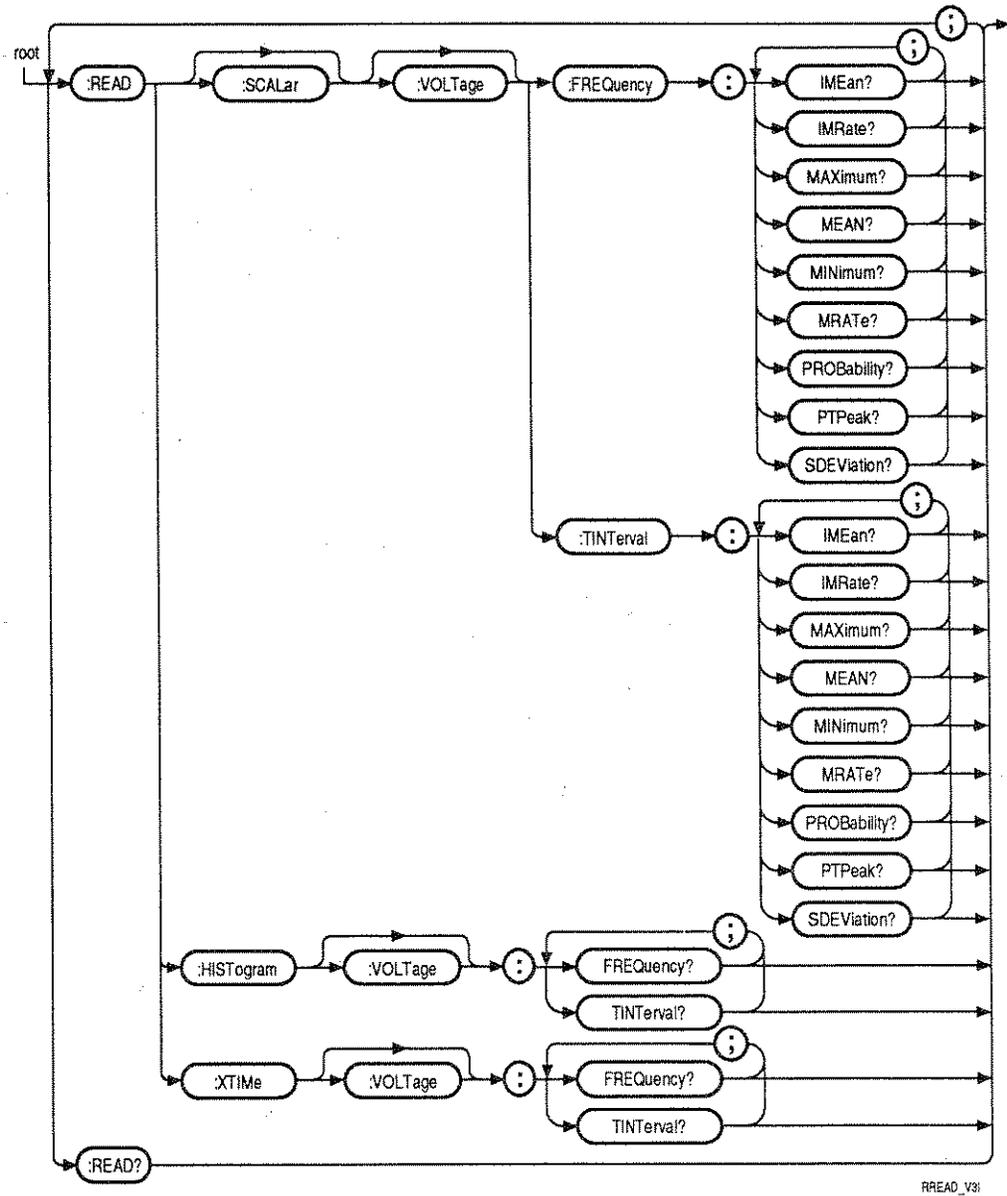


Figure 2-7. :READ Subsystem Commands Syntax

:READ Subsystem

- This command is most commonly used in conjunction with the :CONFIgure command in order to provide a :MEASure capability in which you can provide fine adjustments to the Analyzer's operation. You must use the :READ command that matches the command used to set up the measurement. For example, if you use the command :CONF:XTIM:FREQ to set up the measurement, you must use the commands :READ? or :READ:XTIM:FREQ to initiate a new measurement and place the data in the Output Queue.
- This command aborts any measurement in progress, places the Analyzer in idle, and performs the identical function :INITiate:IMMediate then :FETCh?.
- When :READ? is sent without any function specified, the last function used is retrieved.
- For a description of the analysis functions, see the detailed description of the corresponding function in the :CALCulate Subsystem.
- The format of the returned data is determined by the :FORMat command, except for the analysis results which is always ASCII.
- Use :SENS:DIM[1|2] subtree commands to interpret sensed data (vs Time or Fast Histogram). Use :CALC:DIM[1|2] to interpret post-processed data (Histogram from vs Time).
- Use :SENS:HIST:SAMP subtree commands to query number of samples in or out of Fast Histogram. Use :CALC:HIST:SAMP subtree commands to query number of samples in or out of Histogram from vs Time.
- This command can also generate errors -230, and +2011.

Related Front-Panel Keys

See *figure 1-3*.

:CALCulate Subsystem

The :CALCulate subsystem performs post-acquisition data processing and data transfer of the corresponding result. For example, the functions in the :SENSe subsystem are related to data acquisition, while the :CALCulate subsystem operates on the data acquired by a :SENSe function. The :CALCulate subsystem is logically between the :SENSe subsystem and data output to either an external controller or the Analyzer display.

When a measurement is initiated, the :SENSe subsystem collects data. This data may or may not be transformed by :CALCulate subsystem commands specified in the program. The data is passed on to the selected output.

Therefore, the collection of new data "initiates" the :CALCulate subsystem. The :CALCulate subsystem can also be directed by command to perform a transform or data transfer. This capability makes it possible to change the configuration of the :CALCulate subsystem and, consequently, derive a different set of results from the same sensed data set. This means that you can recalculate without re-acquiring any sense data.

Histogram From vs Time Features

Use the :CALC:HIST[:STAT] ON command to enable the Histogram from vs Time. This only applies when sensing vs Time (:SENS:FUNC "FREQ" or :SENS:FUNC "TINT"). Use :FORMat and :CALC:DATA? to retrieve the Histogram from vs Time data. Use the :CALC:DIM queries to interpret the retrieved data.

:CALCulate Subsystem

Analysis Functions

The Analyzer provides analysis functions for use on Vs Time or Histogram data. The analysis results can be transferred using the :CALC:ANAL:<analy_fcn>:RESult? queries listed in this subsystem. (Where <analy_fcn> refers to the analysis functions listed below.) The results are also displayed on the two lines at the bottom of the display (just below the marker readout lines).

A maximum of four analysis functions can be active (:CALC:ANAL:<analy_fcn>:STATe ON) at one time. Sending additional commands causes the previous commands to be deactivated (the last four activated functions remain active). :CALC:ANAL:<analy_fcn>:STATe must be ON to query results.

The following analysis functions are available:

- **IMEan (Inverse Mean)**
- **IMRate (Inverse Modulation Rate)**
- **MAXimum**
- **MEAN**
- **MINimum**
- **MRATe (Modulation Rate)**
- **PROBability**
- **PTPeak (Peak-to-Peak Modulation)**
- **SDEVIation (Standard Deviation)**

Enabled analysis functions are automatically updated each time new (sense) data is acquired. Analysis functions can be enabled or disabled individually. They can be disabled as a group with the :CALC:ANAL:AOFF command.

Because these commands are very similar, they are all covered in a single :CALCulate:ANALysis:<analy_fcn>[:STATe] and :CALCulate:ANALysis:<analy_fcn>:RESult? detailed description entry. For additional information on how individual analysis functions compute the results, see the *Operating Reference Manual*.

:CALCulate Subsystem

Analysis Range The analysis features operate on the measurement values within the analysis range that you specify using the :CALCulate:DELimit[1 | 2] ALL | MARKers command. DEL1 is for vs Time analysis and DEL2 is for Histogram analysis. If the MARKers parameter is chosen, use the :MARKer subsystem commands to specify the position of the **x-axis** markers. Your choices for the range of analysis are as follows:

- ALL includes the entire acquisition.
- In vs Time (DEL1), MARKers causes only the measurement values between the Time markers (:MARK:TIME:POS[1 | 2] <numeric value>) to be included in the calculations.
- In Histogram (DEL2), MARKers causes only the measurement values between the Frequency (:MARK:FREQ:POS[1 | 2] <numeric value>) or Time Interval (:MARK:TINT:POS[1 | 2] <numeric value>) markers to be included in the calculations.

Analysis/Marker Tracking Use the :MARK:FREQ:POS:AUTO or :MARK:TINT:POS:AUTO commands to enable or disable the markers tracking analysis function. See *Operating Reference Manual* for a description of marker tracking for each analysis function.

Subsystem Syntax *Figure 2-8* is a syntax diagram of the :CALCulate subsystem.

Comments

- The :CALCulate subsystem's HISTogram subtree has capabilities corresponding to the Histogram from vs Time (selected on the Histogram or Display menus). The Fast Histogram selection on the Histogram menu is equivalent to the Histogram selected using the :SENSE:FUNCTION command.

:CALCulate Subsystem

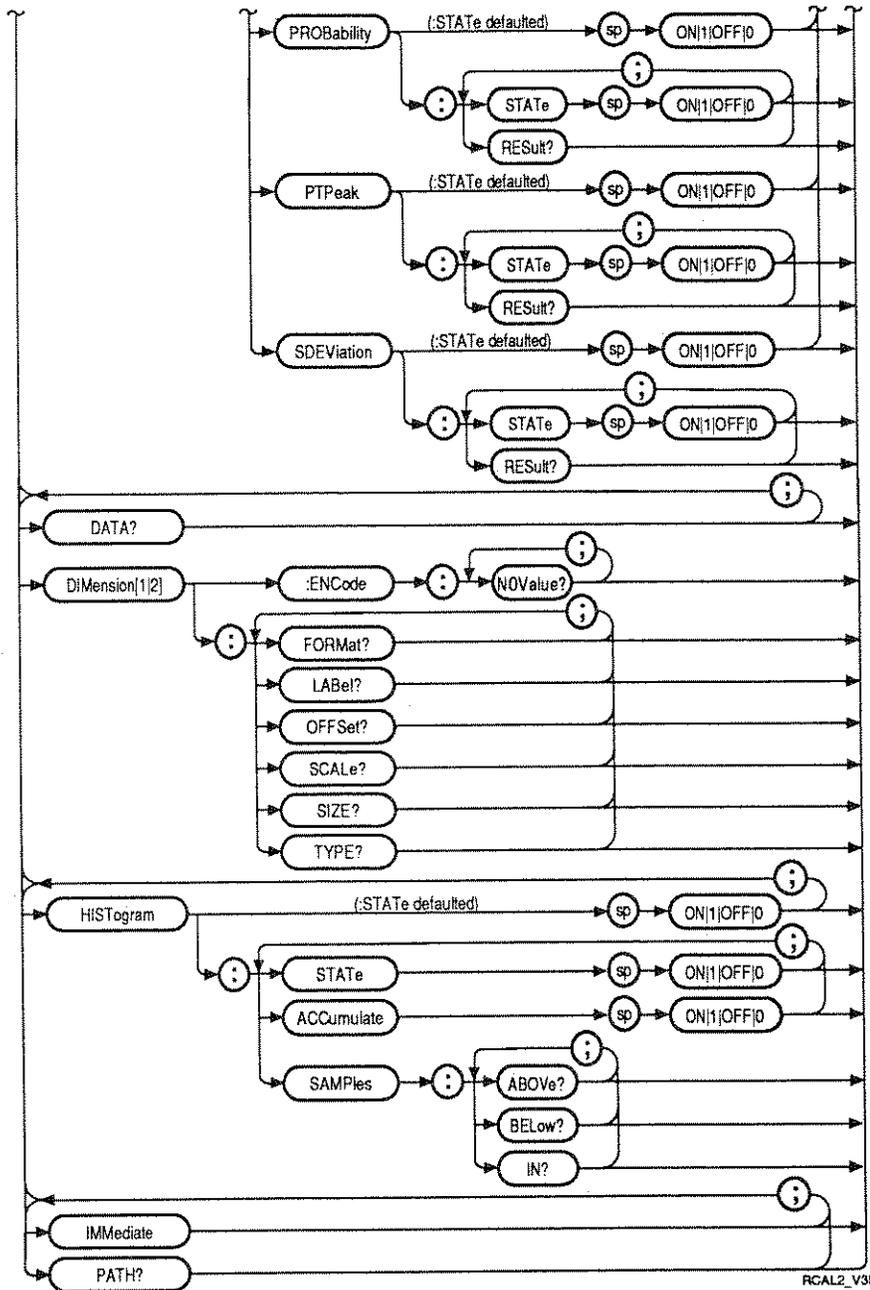


Figure 2-8. :CALCulate Subsystem Commands Syntax Chart (Continued)

:CALCulate:ANALysis:AOff

:CALCulate:ANALysis:AOff

This event command turns off all analysis functions by setting all :CALC:ANAL:<analy_fcn>[:STAT] commands to OFF.

Syntax :CALCulate:ANALysis:AOff

Example OUTPUT 712;":CALC:ANAL:AOff"

Comments • Event; No Query.

Related Front-Panel Keys See *figure 1-4*.

:CALCulate:ANALysis:DELimit[1|2]

This command selects the limits of the data to be analyzed. The analysis features operate on the measurement values within the analysis range that you specify using the :CALCulate:DELimit[1|2] ALL|MARKers command. DEL1 is for vs Time analysis and DEL2 is for Histogram analysis. If the MARKers parameter is chosen, use the :MARKer subsystem commands to specify the position of the **x-axis** markers. Your choices for the range of analysis are as follows:

- ALL includes the entire acquisition.
- In vs Time (DEL1), MARKers causes only the measurement values between the time markers (:MARK:TIME:POS[1|2] <numeric value>) to be included in the calculations.
- In Histogram (DEL2), MARKers causes only the measurement values between the Frequency (:MARK:FREQ:POS[1|2] <numeric value>) or Time Interval (:MARK:TINT:POS[1|2] <numeric value>) markers to be included in the calculations.

Syntax :CALCulate:ANALysis:DELimit[1|2] ALL | MARKers

Example OUTPUT 712;":CALC:ANAL:DEL ALL"

Comments • *RST= ALL

Related Front-Panel Keys See *figures 1-12* and *1-13*.

:CALCulate:ANALysis:<analy_fcn>[:STATe]

:CALCulate:ANALysis:<analy_fcn>[:STATe]

This command enables the specified analysis function where <analy_fcn> is one of the following:

- **IMEan (Inverse Mean)**
- **IMRate (Inverse Modulation Rate)**
- **MAXimum**
- **MEAN**
- **MINimum**
- **MRATe (Modulation Rate)**
- **PROBability**
- **PTPeak (Peak-to-Peak Modulation)**
- **SDEVIation (Standard Deviation)**

Syntax :CALCulate:ANALysis:<analy_fcn>[:STATe] ON|1|OFF|0

Example OUTPUT 712;":CALC:ANAL:IME ON"

- Comments**
- :CALC:ANAL:<analy_fcn> must be ON to query using the corresponding command :CALC:ANAL:<analy_fcn>:RES?.
 - A maximum of four analysis commands can be active (:CALC:ANAL:<analy_fcn> ON) at one time.
 - For additional information on how individual analysis functions compute the results, see the *Operating Reference Manual*.
 - Related commands are :MARK:FREQ:POS:AUTO, :MARK:TINT:POS:AUTO, and :CALC:ANAL:DEL[1|2] ALL | MARK.
 - *RST= OFF

Related Front-Panel Keys See *figure 1-4*.

:CALCulate:ANALysis:<analy_fcn>:RESult?

:CALCulate:ANALysis:<analy_fcn>:RESult?

This query returns the result of the specified analysis function where <analy_fcn> is one of the following:

- **IMEan (Inverse Mean)**
- **IMRate (Inverse Modulation Rate)**
- **MAXimum**
- **MEAN**
- **MINimum**
- **MRATe (Modulation Rate)**
- **PROBability**
- **PTPeak (Peak-to-Peak Modulation)**
- **SDEViation (Standard Deviation)**

Syntax :CALCulate:ANALysis:<analy_fcn>:RESult?

Example OUTPUT 712;":CALC:ANAL:IME:RES?"

- Comments**
- The :CALC:ANAL:<analy_fcn>:RES? commands have the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - :CALC:ANAL:<analy_fcn> must be ON to query using the corresponding command :CALC:ANAL:<analy_fcn>:RES?.
 - A maximum of four analysis commands can be active (:CALC:ANAL:<analy_fcn> ON) at one time.
 - For additional information on how individual analysis functions compute the results, see the *Operating Reference Manual*.

:CALCulate:ANALysis:<analy_fcn>:RESult?

- Related commands are :MARK:FREQ:POS:AUTO, :MARK:TINT:POS:AUTO, and :CALC:ANAL:DEL[1|2] ALL | MARK.
- 9.91E37 is returned if a result could not be computed (error +2011) or if the corresponding :CALC:ANAL:<analy_fcn>:STAT is OFF (error -221).

**Related
Front-Panel Keys**

See *figure 1-4*.

:CALCulate:DATA?

:CALCulate:DATA?

This query places the Histogram from vs Time data in the Output Queue. The data format is determined by the :FORMat command.

No measurement data is returned if one of the following conditions is true:

- :CALC:HIST[:STAT] OFF (error -221)
- :CALC:HIST[:STAT] ON while sensing Fast Histogram (error -221)
- No valid data exists (error -230)

Syntax :CALCulate:DATA?

Example OUTPUT 712;":CALC:DATA?"

- Comments**
- The :CALC:DATA? command has the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - Useful information about the data can be obtained by using the :CALC:DIM subtree queries.
 - Asking for data while a measurement is in progress results in this query waiting for the measurement to complete.
 - See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using this query.
 - Query only.

Related Front-Panel Keys None

:CALCulate:DIMension[1|2] Subtree

:CALCulate:DIMension[1|2]Subtree

This subtree's commands provide access to the structure and format of the Histogram from vs Time data returned by the :CALCulate:DATA? command. 1 = frequency/time-interval probability dimension (Y-axis). 2= frequency/time-interval bin dimension (X-axis).

Subtree Syntax The :CALC:DIM[1|2] subtree syntax is shown in *figure 2-8* (:CALCulate subsystem syntax diagram.)

Comments

- See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using these queries.

:CALCulate:DIMension[1|2]:ENCode:NOValue?

:CALCulate:DIMension[1|2]:ENCode:NOValue?

This query returns a numeric value that is to be treated as implying “no value recorded” in the :CALC:DATA? response (Histogram from vs Time data). For Histogram data, this value is 0 (zero).

Syntax :CALCulate:DIMension[1|2]:ENCode:NOValue?

Example OUTPUT 712;":CALC:DIM:ENC:NOV?"

- Comments**
- See chapter 5, “Data Transfer” and appendix D, “Programming Examples” for additional information and examples of using this query.
 - Query only

Related Front-Panel Keys None

:CALCulate:DIMension[1|2]:FORMat?

:CALCulate:DIMension[1|2]:FORMat?

This query returns a literal with a signed integer that specifies the data format used to format the :CALC:DATA? query response.

Syntax :CALCulate:DIMension[1|2]:FORMat?

Example OUTPUT 712;":CALC:DIM:FORM?"

- Comments**
- 1 = frequency/time-interval probability dimension (Y-axis). 2 = frequency/time-interval bin dimension (X-axis).
 - Use the :FORMat[:DATA] ASC | BLOC command to specify the desired format. If :FORM ASC is used, this query responds with ASC,+15 indicating that each ASCII formatted floating point value has 15 digits. If :FORM BLOC is used, this query responds with REAL,+64 indicating that the measurement data is transferred in a definite length block as IEEE 754 floating point numbers of 64 bits (8 bytes) per measurement sample.
 - See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using this query.
 - Query only.

Related Front-Panel Keys None

:CALCulate:DIMension[1|2]:LABel?

:CALCulate:DIMension[1|2]:LABel?

This query returns a literal corresponding to the :CALC:DATA? response (Histogram from vs Time) that uniquely distinguishes the dimensions.

| Measurement | :CALC:DIM1:LAB? | :CALC:DIM2:LAB? |
|--------------------------------------|-----------------|-----------------|
| Frequency Histogram from vs Time | FDR | FBIN |
| Time Interval Histogram from vs Time | TIPR | TIB |

Syntax :CALCulate:DIMension[1|2]:LABel?

Example OUTPUT 712;":CALC:DIM:LAB?"

- Comments**
- 1 = frequency/time-interval probability dimension (Y-axis). 2 = frequency/time-interval bin dimension (X-axis).
 - See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using this query.
 - Query only.

Related Front-Panel Keys None

:CALCulate:DIMension[1|2]:OFFSet?

:CALCulate:DIMension[1|2]:OFFSet?

This query returns the offset factor which needs to be applied to the values returned by the :CALCulate:DATA? query (Histogram vs Time data). The scaling and offset factors are defined in the following table:

| Dimension | Equation | Where |
|---------------|--|---|
| DIM1 (Y-axis) | $Y = (\text{scale} * D) + \text{offset}$ | D is a value returned by the :CALCulate:DATA? query. |
| DIM2 (X-axis) | $X = (\text{scale} * S) + \text{offset}$ | S = 1 to value returned by the :CALC:DIM:SIZE? query. |

Syntax :CALCulate:DIMension[1|2]:OFFSet?

Example OUTPUT 712;":CALC:DIM:OFFS?"

- Comments**
- 1 = frequency/time-interval probability dimension (Y-axis). 2 = frequency/time-interval bin dimension (X-axis).
 - The scale value is obtained using the :CALCulate:DIMension[1|2]:SCALE? query.
 - For DIM1, this query always returns a 0 because the :CALC:DATA? returns probability values which do not need to have a Y-axis offset applied to them.
 - See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using this query.
 - Query only.

Related Front-Panel Keys None

:CALCulate:DIMension[1|2]:SCALE?

:CALCulate:DIMension[1|2]:SCALE?

This query returns the scale factor which needs to be applied to the values returned by the :CALCulate:DATA? query (Histogram from vs Time). The scaling and offset factors are defined in the following table:

| Dimension | Equation | Where |
|---------------|--|---|
| DIM1 (Y-axis) | $Y = (\text{scale} * D) + \text{offset}$ | D is a value returned by the :CALCulate:DATA? query. |
| DIM2 (X-axis) | $X = (\text{scale} * S) + \text{offset}$ | S = 1 to value returned by the :CALC:DIM:SIZE? query. |

Syntax :CALCulate:DIMension[1|2]:SCALE?

Example OUTPUT 712;":CALC:DIM:SCAL?"

- Comments**
- 1 = frequency/time-interval probability dimension (Y-axis). 2 = frequency/time-interval bin dimension (X-axis).
 - For DIM1, this query always returns a 1 because the :CALC:DATA? returns probability values which do not need to have a Y-axis scale factor applied to them.
 - The offset value is obtained using the :CALCulate:DIMension[1|2]:OFFSet? query.
 - See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using this query.
 - Query only.

Related Front-Panel Keys None

:CALCulate:DIMension[1|2]:SIZE?

:CALCulate:DIMension[1|2]:SIZE?

This query returns the number of histogram bins (Histogram vs Time) present in the data.

The number of histogram bins in the Histogram from vs Time measurement is a function of the number of vs Time Y-axis pixels used. When the panorama is off, there are more possible vs Time Y-axis values as shown in the following table (table assumes valid data exists):

| Panorama | :CALC:DIM1:SIZE? | :CALC:DIM2:SIZE? |
|------------------|------------------|------------------|
| SWE:TIME:PAN OFF | 257 | 257 |
| SWE:TIME:PAN ON | 225 | 225 |

This query should be used when the Analyzer is idle because the validity of the data is stable at that time.

Syntax :CALCulate:DIMension[1|2]:SIZE?

Example OUTPUT 712;":CALC:DIM:SIZE?"

- Comments**
- 1 = frequency/time-interval probability dimension (Y-axis). 2 = frequency/time-interval bin dimension (X-axis).
 - If there is no valid data, :SIZE? returns 0.
 - See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using this query.
 - Query only.

Related Front-Panel Keys None

:CALCulate:DIMension[1 | 2]:TYPE?

:CALCulate:DIMension[1 | 2]:TYPE?

This query returns a literal that indicates whether the dimension is implicit (IMPL) or explicit (EXPL).

Syntax :CALCulate:DIMension[1 | 2]:TYPE?

Example OUTPUT 712;":CALC:DIM:TYPE?"

- Comments**
- 1 = frequency/time-interval probability dimension (Y-axis). 2 = frequency/time-interval bin dimension (X-axis).
 - See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using this query.
 - :CALC:DIM1:TYPE? (Y-dimension of Histogram from vs Time) is always explicit and :CALC:DIM2:TYPE? (X-dimension of Histogram from vs Time) is always implicit. This indicates that one dimension of data is always returned by :CALC:DATA? and this data corresponds to the Y-dimension of the Histogram from vs Time measurement.
 - Query only.

Related Front-Panel Keys None

:CALCulate:HISTogram[:STATe]

:CALCulate:HISTogram[:STATe]

This command enables vs Time data to be post-processed into a histogram. This affects what is displayed when sensing vs Time measurements (see :SENS:FUNC "FREQ[1|2|3]" or :SENS:FUNC "TINT" commands). ON = Histogram from vs Time. OFF= measurement data vs Time.

The Histogram is the result of a post processing of measurement data. Turning ON the Histogram from vs Time also enables querying Histogram from vs Time data using the CALC:DATA? query.

It is not the same as Fast Histogram (see :SENS:FUNC "FREQ[1|2|3]:HIST" or :SENS:FUNC "TINT:HIST" commands).

Syntax :CALCulate:HISTogram[:STATe] ON|1|OFF|0

Example OUTPUT 712;":CALC:HIST ON"

Comments . *RST= OFF

Related Front-Panel Keys See *figures 1-11* and *1-14*.

:CALCulate:HISTogram:ACCumulate

:CALCulate:HISTogram:ACCumulate

This command determines whether the Histogram from vs Time accumulates from sweep to sweep. ON= accumulate, OFF= no accumulating.

Syntax :CALCulate:HISTogram:ACCumulate ON|1|OFF|0

Example OUTPUT 712;":CALC:HIST:ACC ON"

- Comments**
- Coupled to :SENS:HIST:ACC in that these commands share the same value.
 - If :CALC:HIST:ACC is ON, accumulation occurs independent of :CALC:HIST[:STAT].
 - ***RST= OFF**

Related Front-Panel Keys See figure 1-11 and 1-14.

:CALCulate:HISTogram:SAMPles:ABOVe?

:CALCulate:HISTogram:SAMPles:ABOVe?

When calculating a Histogram from vs time, this query returns the number of samples at or above :SENS:FREQ:RANG[1|2|3]:UPP (if sensing frequency vs Time) or :SENS:TINT:RANG:UPP (if sensing time interval vs Time).

Syntax :CALCulate:HISTogram:SAMPles:ABOVe?

Example OUTPUT 712;":CALC:HIST:SAMP:ABOV?"

- Comments**
- The :CALC:HIST:SAMP:ABOV? command has the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - The response is 0 (zero) if queried when :CALC:HIST OFF (error -221), when sensing a Fast Histogram (error -221), or when no valid data exists (error -230).
 - Query only.

Related Front-Panel Keys None

:CALCulate:HISTogram:SAMPles:BELOW?

:CALCulate:HISTogram:SAMPles:BELOW?

When calculating a Histogram from vs time, this query returns the number of samples at or below :SENS:FREQ:RANG[1|2|3]:LOW (if sensing frequency vs Time) or :SENS:TINT:RANG:LOW (if sensing time interval vs Time).

Syntax :CALCulate:HISTogram:SAMPles:BELOW?

Example OUTPUT 712;":CALC:HIST:SAMP:BEL?"

- Comments**
- The :CALC:HIST:SAMP:BEL? command has the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - The response is 0 (zero) if queried when :CALC:HIST OFF (error -221), when sensing a Fast Histogram (error -221), or when no valid data exists (error -230).
 - Query only.

Related Front-Panel Keys None

:CALCulate:HISTogram:SAMPles:IN?

:CALCulate:HISTogram:SAMPles:IN?

When calculating a Histogram from vs time, this query returns the number of samples within :SENS:FREQ:RANG[1|2|3]:UPP and :SENS:FREQ:RANG[1|2|3]:LOW (if sensing frequency vs Time) or within :SENS:TINT:RANG:UPP and :SENS:TINT:RANG:LOW (if sensing time interval vs Time).

Syntax :CALCulate:HISTogram:SAMPles:IN?

Example OUTPUT 712;":CALC:HIST:SAMP:IN?

- Comments**
- The :CALC:HIST:SAMP:IN? command has the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - The response is 0 (zero) if queried when :CALC:HIST OFF (error -221); when sensing a Fast Histogram (error -221), or when no valid data exists (error -230).
 - Query only.

Related Front-Panel Keys None

:CALCulate:IMMEDIATE

:CALCulate:IMMEDIATE

This command recalculates existing data without reacquiring data. This also happens when any change is made to the :CALCulate subsystem.

Syntax :CALCulate:IMMEDIATE

Example OUTPUT 712;":CALC:IMM"

Comments • Event; no query.

Related Front-Panel Keys None

:CALCulate:PATH?

:CALCulate:PATH?

This query's response indicates the order in which the calculation functions are performed if they are active. For example, one of the following two returns is always received:

- If either "FREQ[1|2|3]" or "TINT" was selected by the SENSE:FUNCTION command, HIST,ANAL is returned. This response indicates that the Analyzer is sensing a vs Time measurement, so a Histogram from vs Time will be performed *before* any :CALC:ANAL computations when :CALC:HIST:STAT is ON.
- If either "FREQ:HIST" or "TINT:HIST" was selected by the :SENSE:FUNCTION command, ANAL is returned. This response indicates that the Fast Histogram is active and only the analysis functions can be performed. That is, no Histogram from vs Time will be performed even if :CALC:HIST:STAT is ON.

Syntax :CALCulate:PATH?

Example OUTPUT 712;":CALC:PATH?"

- Comments**
- Couples to :SENSE:FUNCTION.
 - Query only.

Related Front-Panel Keys None

:CLEar Subsystem

This event command clears the front-panel display of all measurement data. If the Analyzer is stopped, all data that is currently displayed is erased. If the Analyzer is running, the data is erased, however, new data is displayed after the next successful acquisition. The acquisition status of the Analyzer (running or stopped) is not affected.

The effect of clearing the data, is that it invalidates data available for transferring (using :SENS:DATA?, :CALC:DATA?, or FETCh subsystem commands) or performing post-processing functions.

Syntax *Figure 2-9 is a syntax diagram of the :CLEar subsystem.*

Example OUTPUT 712;":CLE"

Comments • Event; no query.

Related Front-Panel Keys See *figure 1-3*.

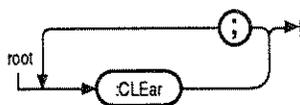


Figure 2-9. :CLEar Subsystem Commands Syntax Chart

:CLEar Subsystem

:DIAGnostic Subsystem

The :DIAGnostic subsystem controls the remote calibration of the Analyzer.

Subsystem Syntax

Figure 2-10 is the syntax diagram for the DIAGnostic subsystem.

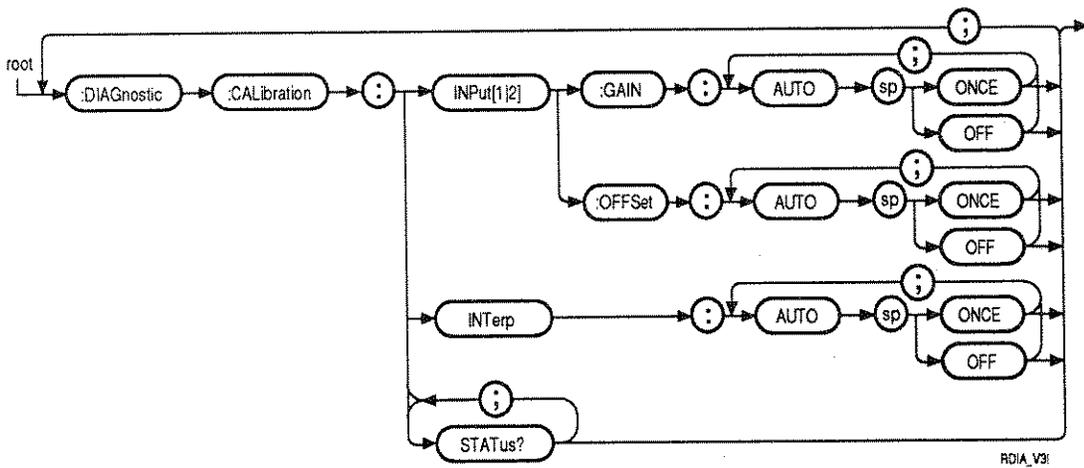


Figure 2-10. :DIAGnostic Subsystem Commands Syntax Chart

:DIAGnostic:CALibration:INPut[1|2]:GAIN:AUTO

:DIAGnostic:CALibration:INPut[1|2]:GAIN:AUTO

This command calibrates the gain of the selected input amplifier when ONCE is specified. 1 = channel A and 2 = channel B.

Syntax :DIAGnostic:CALibration:INPut[1|2]:GAIN:AUTO
ONCE|OFF

Example OUTPUT 712;":DIAG:CAL:INP2:GAIN:AUTO ONCE"

- Comments**
- Instructions on how to perform the calibration are displayed on the front-panel display.
 - Use the :DIAG:CAL:STAT? query to determine if calibration passed or failed.
 - *RST= OFF

Related Front-Panel Keys See *figure 1-17*.

:DIAGnostic:CALibration:INPut[1|2]:OFFSet:AUTO

:DIAGnostic:CALibration:INPut[1|2]:OFFSet:AUTO

This command calibrates the offset of the selected input amplifier when ONCE is specified. 1 = channel A and 2 = channel B.

Syntax :DIAGnostic:CALibration:INPut[1|2]:OFFSet:AUTO
ONCE|OFF

Example OUTPUT 712;":DIAG:CAL:INP2:OFFS:AUTO ONCE"

- Comments**
- Instructions on how to perform the calibration are displayed on the front-panel display.
 - Use the :DIAG:CAL:STAT? query to determine if calibration passed or failed.
 - *RST= OFF

**Related
Front-Panel Keys**

See *figure 1-17*.

:DIAGnostic:CALibration:INTerp:AUTO

:DIAGnostic:CALibration:INTerp:AUTO

This command calibrates both interpolators when ONCE is specified.

Syntax :DIAGnostic:CALibration:INTerp:AUTO ONCE | OFF

Example OUTPUT 712;":DIAG:CAL:INT:AUTO ONCE"

- Comments**
- Use the :DIAG:CAL:STAT? query to determine if calibration passed or failed.
 - *RST= OFF

Related Front-Panel Keys See *figure 1-17*.

:DIAGnostic:CALibration:STATus?

:DIAGnostic:CALibration:STATus?

This query returns 0 for calibration passed and 1 for calibration failed. It can be used after any calibration to determine if the calibration was successful.

Syntax :DIAGnostic:CALibration:STATus?

Example OUTPUT 712;":DIAG:CAL:STAT?"

Comments • Query only.

Related Front-Panel Keys None.

:DIAGnostic:CALibration:STATus?

:DISPlay Subsystem

The :DISPlay subsystem controls the selection and presentation of textual and graphical information on the Analyzer's display. This information includes measurement data, user-interaction displays, and data presented to the Analyzer by the controller. :DISPlay is independent of (and does not modify), the data returned to the controller.

Subsystem Syntax

Figure 2-11 is the syntax diagram for the DISPlay subsystem.

:DISPlay Subsystem

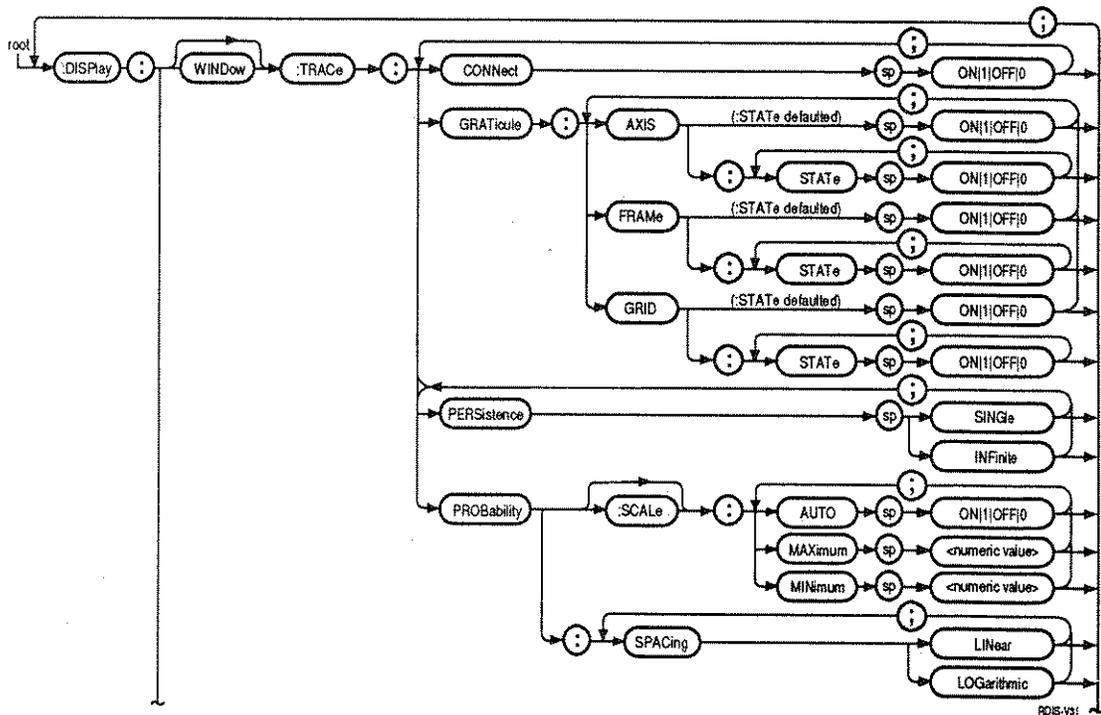


Figure 2-11. :DISPlay Subsystem Commands Syntax Chart

:DISPlay Subsystem

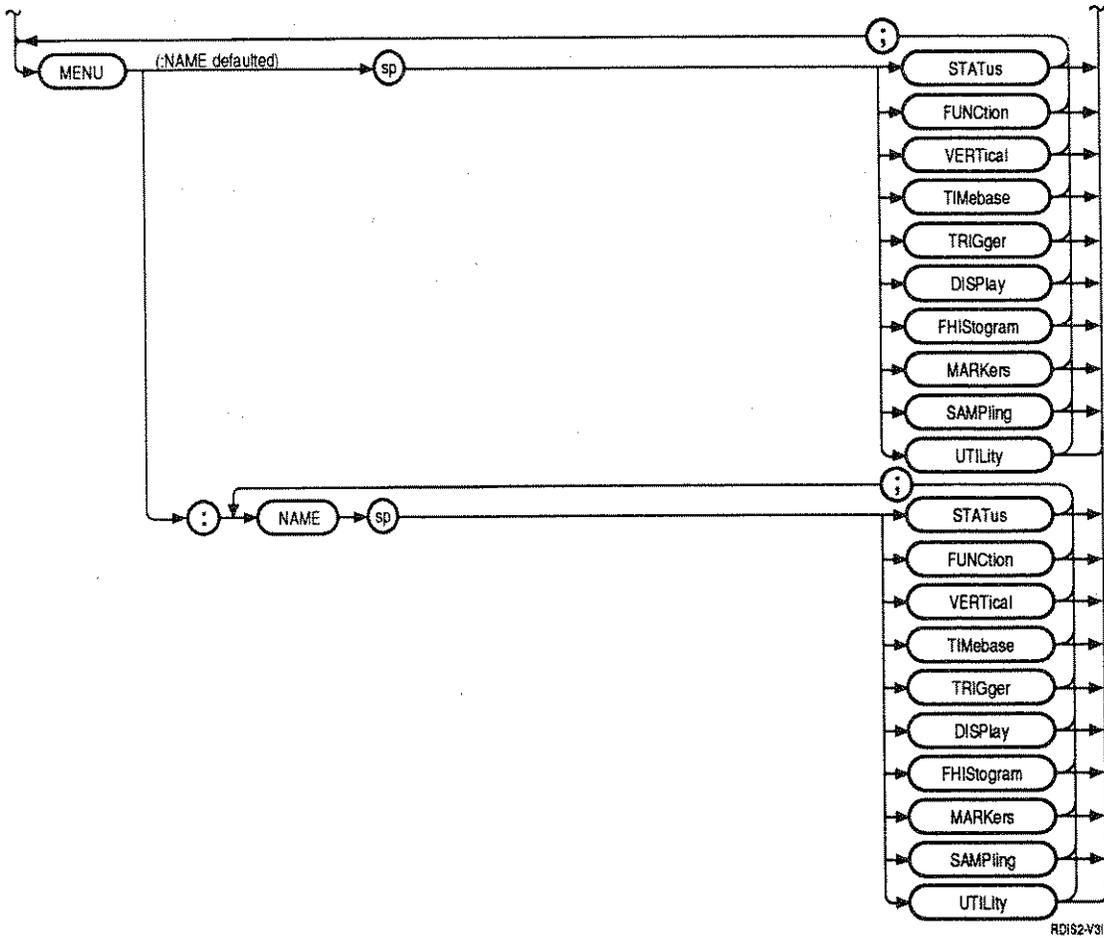


Figure 2-11. :DISPlay Subsystem Commands Syntax Chart (Continued)

:DISPlay[:WINDow]:TRACe:CONNect

:DISPlay[:WINDow]:TRACe:CONNect

When ON is specified, this command provides a continuous line on the display by connecting the individual measurement points. When OFF is specified, the individual measurement points are displayed.

Syntax :DISPlay[:WINDow]:TRACe:CONNect ON|1|OFF|0

Example OUTPUT 712;":DISP:TRAC:CONN ON"

- Comments**
- This command does not impact the data that is returned to the computer.
 - *RST = ON

Related Front-Panel Keys See *figure 1-11*.

:DISPlay[:WINDow]:TRACe:GRATicule:AXIS[:STATe]

:DISPlay[:WINDow]:TRACe:GRATicule:AXIS[:STATe]

This command determines if the axes are visible or not.

Syntax :DISPlay[:WINDow]:TRACe:GRATicule:AXIS[:STATe]
 ON|1|OFF|0

Example OUTPUT 712;":DISP:TRAC:GRAT:AXIS ON"

- Comments**
- The GRATICule commands are individually selectable but highly coupled. That is, displaying the axes using the :DISP:TRAC:GRAT:AXIS ON command also automatically turns the FRAME to ON and the GRID to OFF. If you cycle through the front panel softkey, you can see that it works in a similar manner.
 - Setting any graticule :STATe to OFF, sets all graticule states to OFF.
 - *RST= ON

Related Front-Panel Keys See *figure 1-11*.

:DISPlay[:WINDow]:TRACe:GRATicule:FRAMe[:STATe]

:DISPlay[:WINDow]:TRACe:GRATicule:FRAMe[:STATe]

This command determines if the FRAME is visible or not.

Syntax :DISPlay[:WINDow]:TRACe:GRATicule:FRAMe[:STATe]
ON|1|OFF|0

Example OUTPUT 712;":DISP:TRAC:GRAT:FRAM ON"

- Comments**
- The GRATicule commands are individually selectable but highly coupled. That is, displaying the frame using the :DISP:TRAC:GRAT:FRAM ON command also automatically turns both the AXIS and GRID to OFF. If you cycle through the front panel softkey, you can see that it works in a similar manner.
 - Setting any graticule :STATe to OFF, sets all graticule states to OFF.
 - *RST= ON

Related Front-Panel Keys See *figure 1-11*.

:DISPlay[:WINDow]:TRACe:GRATicule:GRID[:STATe]

:DISPlay[:WINDow]:TRACe:GRATicule:GRID[:STATe]

This command determines if the GRID is visible or not.

Syntax :DISPlay[:WINDow]:TRACe:GRATicule:GRID[:STATe]
ON|1|OFF|0

Example OUTPUT 712,":DISP:TRAC:GRAT:GRID ON"

- Comments**
- The GRATICule commands are individually selectable but highly coupled. That is, displaying the grid using the :DISP:TRAC:GRAT:GRID ON command also automatically turns both the AXIS and FRAME to ON. If you cycle through the front panel softkey, you can see that it works in a similar manner.
 - Setting any graticule :STATe to OFF, sets all graticule states to OFF.
 - *RST= OFF

Related Front-Panel Keys See *figure 1-11*.

:DISPlay[:WINDow]:TRACe:PERStence

:DISPlay[:WINDow]:TRACe:PERStence

This command defines the persistence of the Analyzer's vs Time display. It only applies when :SENS:SWE:TIME:REP OFF.

In infinite persistence, the data points remain on the display until you change a critical parameter or clear the display (using either the Clear display key on the front panel or the :CLEar command).

Syntax :DISPlay[:WINDow]:TRACe:PERStence SINGle | INFinite

Example OUTPUT 712;":DISP:TRAC:PERSt INF"

- Comments**
- This command does not impact the data returned to the computer.
 - *RST= SING

Related Front-Panel Keys See *figure 1-11*.

:DISPlay[:WINDow]:TRACe:PROBability[:SCALe]:AUTO

:DISPlay[:WINDow]:TRACe:PROBability[:SCALe]:AUTO

This command selects automatic scaling for the Y-axis of the Histogram display. ON= display is scaled based on data instead of :DISP:TRAC:PROB:MIN and :DISP:TRAC:PROB:MAX.

Syntax :DISPlay[:WINDow]:TRACe:PROBability[:SCALe]:AUTO
ON|1|OFF|0

Example OUTPUT 712;":DISP:TRAC:PROB:AUTO ON"

Comments

- Setting :DISP:TRAC:PROB:AUTO ON does not affect the minimum (:DISP:TRAC:PROB:MIN) and maximum (:DISP:TRAC:PROB:MAX) settings.

However, it does cause the display scale to be based on the Histogram data rather than the specified minimum and maximum settings.

- *RST= ON

Related Front-Panel Keys See *figure 1-11*.

:DISPlay[:WINDow]:TRACe:PROBability[:SCALe]:MAXimum

:DISPlay[:WINDow]:TRACe:PROBability[:SCALe]:MAXimum

This command selects the maximum probability for the Histogram display.

Syntax :DISPlay[:WINDow]:TRACe:PROBability[:SCALe]:MAXimum
<numeric value>

Example OUTPUT 712;":DISP:TRAC:PROB:MAX 20"

- Comments**
- Using this command turns the :DISP:TRAC:PROB:AUTO function OFF.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - **Units = Percent [PCT]**
 - ***RST= 100**

Related Front-Panel Keys See *figure 1-11*.

:DISPlay[:WINDow]:TRACe:PROBability[:SCALE]:MINimum

:DISPlay[:WINDow]:TRACe:PROBability[:SCALE]:MINimum

This command selects the minimum probability for the Histogram display.

Syntax :DISPlay[:WINDow]:TRACe:PROBability[:SCALE]:MINimum
<numeric value>

Example OUTPUT 712;":DISP:TRAC:PROB:MIN 1"

- Comments**
- Using this command turns the :DISP:TRAC:PROB:AUTO function OFF.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - Units = Percent [PCT]
 - *RST = 0

Related Front-Panel Keys

See *figure 1-11*.

:DISPlay[:WINDow]:TRACe:PROBability:SPACing

:DISPlay[:WINDow]:TRACe:PROBability:SPACing

This command selects the type of spacing used on the Histogram display's Y-axis (probability axis).

| Parameter Name | Description |
|----------------|-----------------------------|
| LINear | Selects linear display |
| LOGarithmic | Selects logarithmic display |

Syntax :DISPlay[:WINDow]:TRACe:PROBability:SPACing
LINEAR|LOGarithmic

Example OUTPUT 712;":DISP:TRAC:PROB:SPAC LOG"

Comments • *RST= LIN

Related Front-Panel Keys See *figure 1-11*.

:DISPlay:MENU[:NAME]

:FORMat Subsystem

The :FORMat subsystem sets a data format for transferring numeric array information. This data format is used for response data by the following commands:

- :SENSE:DATA?
- :CALCulate:DATA?
- :FETCh subsystem
- :MEASure subsystem
- :READ subsystem

Subsystem Syntax Figure 2-12 is the syntax diagram for the :FORMat subsystem.

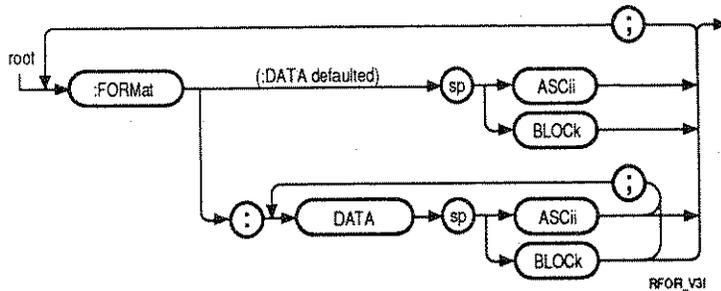


Figure 2-12. :FORMat Subsystem Commands Syntax Chart

:FORMat[:DATA]

:FORMat[:DATA]

This command selects the data format for transferring numeric array data. This data format is used for response data by the following commands:

- :SENSe:DATA?
- :CALCulate:DATA?
- :FETCh subsystem
- :MEASure subsystem
- :READ subsystem

| Type | Description |
|-------|---|
| AScii | Numeric data is transferred as ASCII bytes in <NR1>, <NR2>, or <NR3> format, as appropriate. The numbers are separated by commas as specified in IEEE 488.2-1897. The number of significant digits that are returned is automatically set by the Analyzer to optimize the data transfer rate and provide adequate resolution. |
| BLOCK | BLOCK numeric data is transferred in a definite length block as either 16-bit signed integers or 64-bit IEEE 754 floating point numbers. The choice of format is automatically selected by the type of data the Analyzer is going to send. |

If you do not know the format that is going to be returned, use the :CALC:DIM[1|2]:FORM? or :SENS:DIM[1|2]:FORM? queries because they are more specific.

Syntax :FORMat[:DATA] AScii | BLOCK

Example OUTPUT 712;":FORM BLOC"

:FORMat[:DATA]

- Comments**
- If the ASCii parameter is selected and the Analyzer is going to send vs Time data via :SENS:DATA?, :MEASure, READ, or FETCh, <NR1> format numbers are returned. If Histogram data is going to be specified, <NR3> format numbers with 15 digits are returned.
 - If the BLOCk parameter is selected and the Analyzer is going to send vs Time data via :SENS:DATA?, :MEASure, READ, or FETCh, the 16-bit integer format is automatically used. If Histogram data is going to be specified, the 64-bit floating point number format is automatically used.
 - Whichever parameter is used with vs Time data, the :SENS:DATA? command and the :MEASure, READ, and FETCh subsystem commands can be used to return the data. Histogram data can use the same commands, plus the :CALC:DATA? command.
 - *RST= ASC

Related Front-Panel Keys None

:FORMat[:DATA]

:INITiate Subsystem

The :INITiate subsystem is used to control the initiation of the trigger system that causes the initiation of a measurement acquisition.

For vs Time and Histogram from vs Time measurements, acquisition is controlled by the :TRIGger subsystem commands.

For Fast Histogram measurements, acquisition is controlled by the :SENSe:HISTogram:ARM and :SENSe:HISTogram:TRIGger subtree commands.

Subsystem Syntax

Figure 2-13 is the syntax diagram for the :INITiate subsystem.

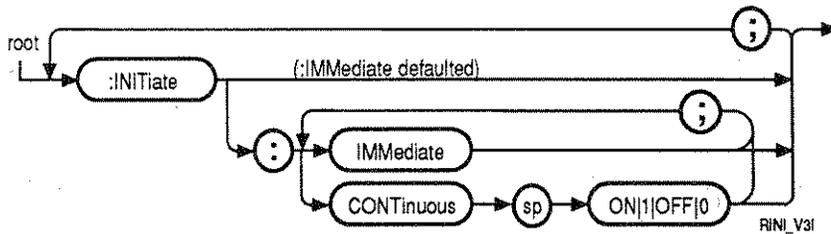


Figure 2-13. :INITiate Subsystem Commands Syntax Chart

:INITiate[:IMMEDIATE]

:INITiate[:IMMEDIATE]

This event command causes the Analyzer to initiate a single measurement sweep/Histogram update. If the Analyzer is already in the process of making a measurement or if :INIT:CONT is set to ON, an :INIT:IMM command has no effect and error -213 (initiate ignored) is generated.

Syntax :INITiate[:IMMEDIATE]

Example OUTPUT 712;":INIT"

- Comments**
- This command is an overlapped command. That is, beginning a measurement using the :INITiate[:IMMEDIATE] command sets the Pending Operation Flag to true. Completing the measurement sets the Pending Operation Flag to false. This flag is also set to false by the ABORT and *RST commands.
 - Event; no query.

Related Front-Panel Keys See *figure 1-3*.

:INITiate:CONTInuous

This command determines whether measurements are continuously initiated or not. When CONTInuous is off, measurements are not made until CONTInuous is set to ON or the :INITiate:IMMEDIATE command is received. When CONTInuous is ON, a new measurement immediately commences after the current measurement is completed.

Syntax :INITiate:CONTInuous ON|1|OFF|0

Example OUTPUT 712;:INIT:CONT ON"

- Comments**
- The beginning of the first measurement initiated by setting :INITiate:CONTInuous to ON sets the Pending Operation Flag associated with the initiated action true. The Pending Operation Flag is set false by an ABORt command, a *RST command, or the completion of a measurement sweep after the :INITiate:CONTInuous command is set to OFF.
 - When the measurements are continuous (that is, :INITiate:CONTInuous is ON), the ABORt command aborts any measurement in progress. However the value of :INITiate:CONTInuous is unaffected. Therefore, measuring starts again immediately.
 - If a single measurement is in progress (:INIT:CONT is OFF), error -213 is generated and the state of :INIT:CONT is unaffected if :INIT:CONT ON is specified.
 - If a single measurement is in progress (:INIT:CONT is OFF), error -210 is generated if :INIT:CONT OFF is specified.
 - ***RST= OFF**

**Related
Front-Panel Keys**

See figure 1-3.

:INITiate:CONTinuous

:INPut[1|2] Subsystem

The :INPut[1|2] subsystem controls the characteristics of the Analyzer's input ports. :INPut1 corresponds to channel A and :INPut2 corresponds to channel B. There is no input control for the optional channel C.

Subsystem Syntax *Figure 2-14 is the syntax diagram for the :INPut subsystem.*

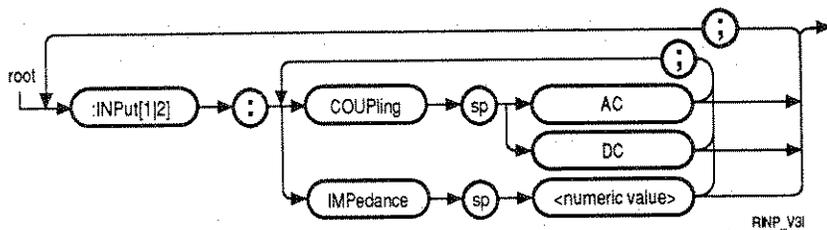


Figure 2-14. :INPut Subsystem Commands Syntax Chart

:INPut[1|2]:COUPling

:INPut[1|2]:COUPling

This command selects input coupling. :INP1:COUP selects the channel A input coupling. :INP2:COUP selects the channel B input coupling.

Syntax :INPut[1|2]:COUPling AC|DC

Example OUTPUT 712;":INP:COUP AC"

- Comments**
- Channel C defaults to AC only coupling (as indicated on the Input menu when channel C is selected).
 - *RST= DC

Related Front-Panel Keys See *figure 1-6*.

:INPut[1 | 2]:IMPedance

:INPut[1 | 2]:IMPedance

This command selects termination impedance for the input signal in Ohms. INPut1 selects the channel A input impedance. INPut2 selects the channel B input impedance (see Comments).

Syntax :INPut[1 | 2]:IMPedance <numeric value>

Example OUTPUT 712;":INP:IMP 50"

- Comments**
- The only two values allowed are 50 and 1.0E+6 Ohms.
 - Channel C defaults to 50 Ohms only (as indicated on the Input menu when channel C is selected).
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - If measuring Time Interval, the :SENS:TINT:ROUTE COMMON | SEParate selection determines whether the specified impedance for B is used.
 - **Units = Ohms [OHM]**
 - ***RST = 1.0E+6**

Related Front-Panel Keys See *figure 1-6*.

:INPut[1|2]:IMPedance

:MARKer Subsystem

:MARKer Subsystem

The :MARKer subsystem commands enable/disable markers, adjust marker settings, and specify whether or not analysis functions will control marker settings.

Subsystem Syntax

Figure 2-15 is the syntax diagram for the :MARKer subsystem.

:MARKer Subsystem

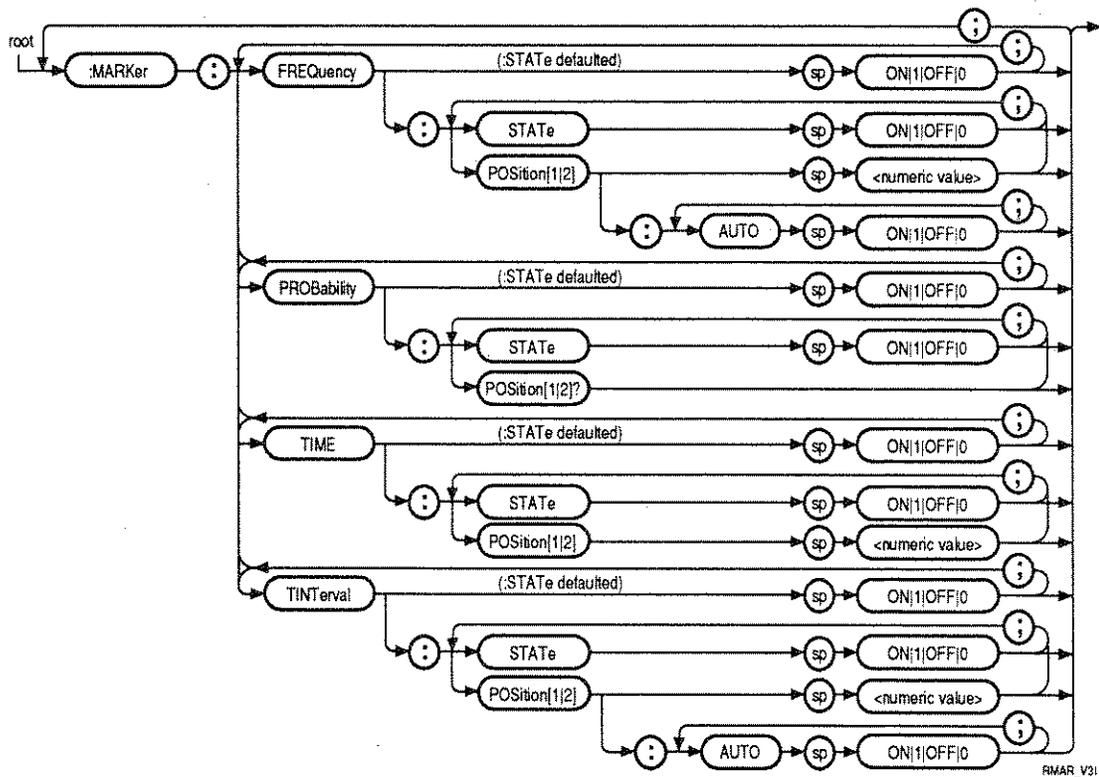


Figure 2-15. :MARKer Subsystem Commands Syntax Chart

:MARKer:FREQuency[:STATe]

:MARKer:FREQuency[:STATe]

This command turns the frequency markers on or off.

Syntax :MARKer:FREQuency[:STATe] ON|1|OFF|0

Example OUTPUT 712;":MARK:FREQ ON"

- Comments**
- The frequency markers are automatically turned on by :CALC:ANAL:DEL2 MARK if measuring a frequency histogram.
 - Turning off the frequency markers while measuring a frequency histogram couples :CALC:ANAL:DEL2 to ALL.
 - ***RST= OFF**

Related Front-Panel Keys See *figures 1-12 and 1-13.*

:MARKer:FREQuency:POSition[1|2]

:MARKer:FREQuency:POSition[1|2]

This command specifies the position of the frequency markers. FREQuency1 controls the F1 marker and FREQuency2 controls the F2 marker.

Syntax :MARKer:FREQuency:POSition[1|2] <numeric value>

Example OUTPUT 712;":MARK:FREQ:POS 100E+6"

- Comments**
- Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - Units= Hertz [HZ]
 - *RST= POS1 is 85E+6; POS2 is 115E+6

Related Front-Panel Keys See figures 1-12 and 1-13.

:MARKer:FREQuency:POSition[1|2]:AUTO

:MARKer:FREQuency:POSition[1|2]:AUTO

This command allows the frequency marker(s) to track the results of the analysis functions.

Syntax :MARKer:FREQuency:POSition[1|2]:AUTO ON|1|OFF|0

Example OUTPUT 712;":MARK:FREQ:POS:AUTO ON"

- Comments**
- :MARK:FREQ:POS1:AUTO, :MARK:FREQ:POS2:AUTO, :MARK:TINT:POS1:AUTO, and :MARK:TINT:POS2:AUTO are all coupled in that they share the same value (all are either ON or OFF). Affecting one, affects all the others.
 - *RST = ON

Related Front-Panel Keys See *figures 1-12* and *1-13*.

:MARKer:PROBability[:STATe]

:MARKer:PROBability[:STATe]

This command turns the histogram markers on or off.

Syntax :MARKer:PROBability[:STATe] ON|1|OFF|0

Example OUTPUT 712;":MARK:PROB ON"

Comments • *RST= OFF

Related Front-Panel Keys See *figures 1-12* and *1-13*.

:MARKer:PROBability:POSition[1|2]?

:MARKer:PROBability:POSition[1|2]?

This query returns the position of the histogram (probability) markers. POSition1 returns the position of the H1 marker and POSition2 returns the position of the H2 marker.

Syntax :MARKer:PROBability:POSition[1|2]?

Example OUTPUT 712;":MARK:PROB:POS?"

- Comments**
- Query only because the probability markers track the frequency markers on a Frequency Histogram or the time interval markers on a Time Interval Histogram.
 - 9.91E37 is returned if an error occurs. Error -221 is generated if the x-axis markers (frequency or time interval) are not enabled or if the x-axis markers are at positions outside the histogram's frequency or time interval range. Error -230 is generated if a single measurement is in progress or if no valid data exists.
 - This query should be used when the Analyzer is idle because the validity of the data is stable at that time.
 - **Units = Percent [PCT]**

Related Front-Panel Keys See *figures 1-12 and 1-13.*

:MARKer:TIME[:STATe]

:MARKer:TIME[:STATe]

This command turns the time markers on or off.

Syntax :MARKer:TIME[:STATe] ON|1|OFF|0

Example OUTPUT 712;":MARK:TIME ON"

- Comments**
- The time markers are automatically turned on by :CALC:ANAL:DEL1 MARK if sensing vs Time and not calculating a Histogram from vs Time.
 - Turning off the time markers while sensing vs Time and not calculating a Histogram from vs Time couples :CALC:ANAL:DEL1 to ALL.
 - *RST= OFF

Related Front-Panel Keys See figures 1-12 and 1-13.

:MARKer:TIME:POSition[1|2]

:MARKer:TIME:POSition[1|2]

This command specifies the position of the time markers. POSition1 controls the T1 marker and POSition2 controls the T2 marker.

Syntax :MARKer:TIME:POSition[1|2] <numeric value>

Example OUTPUT 712;":MARK:TIME:POS 1E-3"

- Comments**
- Can use MINimum and MAXimum instead of <numeric value> [suffix].
 - Units = Seconds [S]
 - *RST= POS1 is -0.4E-3; POS2 is +0.4E-3

Related Front-Panel Keys See *figures 1-12 and 1-13*.

:MARKer:TINterval[:STATe]

:MARKer:TINterval[:STATe]

This command turns the time interval markers on or off.

Syntax :MARKer:TINterval[:STATe] ON | 1 | OFF | 0

Example OUTPUT 712;":MARK:TINT ON"

- Comments**
- The time interval markers are automatically turned on by :CALC:ANAL:DEL2 MARK if measuring a time interval histogram.
 - Turning off the time interval markers while measuring a time interval histogram couples :CALC:ANAL:DEL2 to ALL.
 - *RST= OFF

Related Front-Panel Keys See *figures 1-12* and *1-13*.

:MARKer:TINTerval:POSition[1|2]

:MARKer:TINTerval:POSition[1|2]

This command specifies the position of the time interval markers. POSition1 controls the I1 marker and POSition2 controls the I2 marker. Can use MINimum and MAXimum instead of <numeric value> [suffix].

Syntax :MARKer:TINTerval:POSition[1|2] <numeric value>

Example OUTPUT 712;":MARK:TINT:POS 1E-3"

- Comments**
- Units = Seconds [S]
 - *RST= POS1 is +0.7E-3; POS2 is +1.3E-3

Related Front-Panel Keys See *figures 1-12* and *1-13*.

:MARKer:TINTerval:POSition[1|2]:AUTO

:MARKer:TINTerval:POSition[1|2]:AUTO

This command allows the time interval marker to track the results of the analysis functions.

Syntax :MARKer:FREQuency:POSition[1|2]:AUTO ON|1|OFF|0

Example OUTPUT 712;":MARK:FREQ:POS:AUTO ON"

- Comments**
- :MARK:FREQ:POS1:AUTO,
:MARK:FREQ:POS2:AUTO,
:MARK:TINT:POS1:AUTO, and
:MARK:TINT:POS2:AUTO are all coupled in that they share the same value (all are either ON or OFF). Affecting one, affects all the others.
 - *RST= ON

Related Front-Panel Keys See figures 1-12 and 1-13.

[[:SENSe] Subsystem

The SENSe subsystem setup commands are divided into nine subtrees to match the Analyzer's capabilities. The subtree's commands set controls that directly affect the Analyzer's settings.

Subsystem Syntax *Figure 2-16 is the syntax diagram for the :SENSe subsystem.*

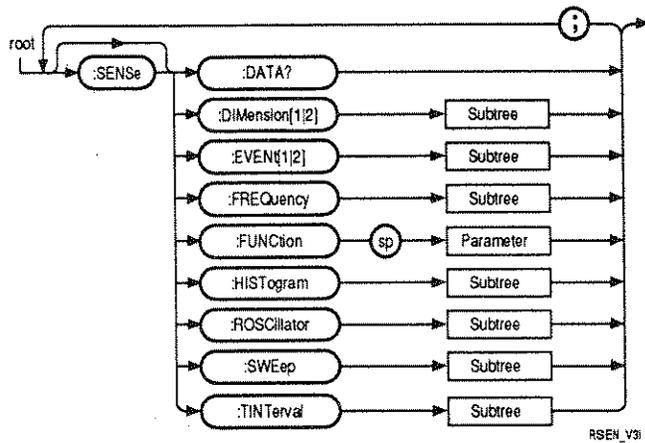


Figure 2-16. :SENSe Subsystem Commands Syntax Chart

[[:SENSe]:DATA?

[[:SENSe]:DATA?

This query provides access to the result data of the :SENSe subsystem (vs Time or Fast Histogram data). This query returns data values according to the format specified by the :FORMat subsystem command.

If Panorama is turned on(:SWE:TIME:PAN ON), Panorama data is returned, but no repetitive points or averaging is reflected in the data.

Syntax *Figure 2-17* is the syntax diagram for the :DATA? command.

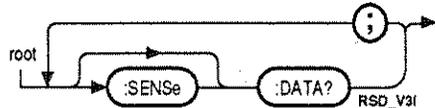


Figure 2-17. :DATA? Subtree Commands Syntax Chart

[[:SENSe]:DATA?

Example OUTPUT 712;":DATA?"

- Comments**
- The :SENS:DATA? command has the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - Useful information about the data to be transferred can be obtained by using the :SENS:DIM subtree queries.
 - If no valid data exists, no measurement data is returned. Instead, error -230 is generated.
 - Asking for data while a measurement is in progress will result in this query waiting for the measurement to complete.
 - Additional information on how to use :SENS:DATA? and :SENS:DIM subtree queries is contained in chapter 5, "Data Transfer" and the examples in appendix D, "Programming Examples."
 - Query only.

Related Front-Panel Keys None

[[:SENSe]:DIMension[1|2] Subtree

[[:SENSe]:DIMension[1|2] Subtree

The :DIMension[1|2] subtree commands provide access to the structure and format of the data returned by the :SENS:DATA? query (vs Time or Fast Histogram data).

DIMension1 = Y-axis. DIMension2 = X-axis.

For vs Time: 1 = frequency or time interval and 2 = time.

For Fast Histogram: 1 = frequency or time interval probability and 2 = frequency or time interval bins.

Syntax *Figure 2-18 is the syntax diagram for the :DIMension[1|2] subtree.*

Comments

- Additional information on how to use :SENS:DATA? and :SENS:DIM subtree queries is contained in chapter 5, "Data Transfer" and the examples in appendix D, "Programming Examples."

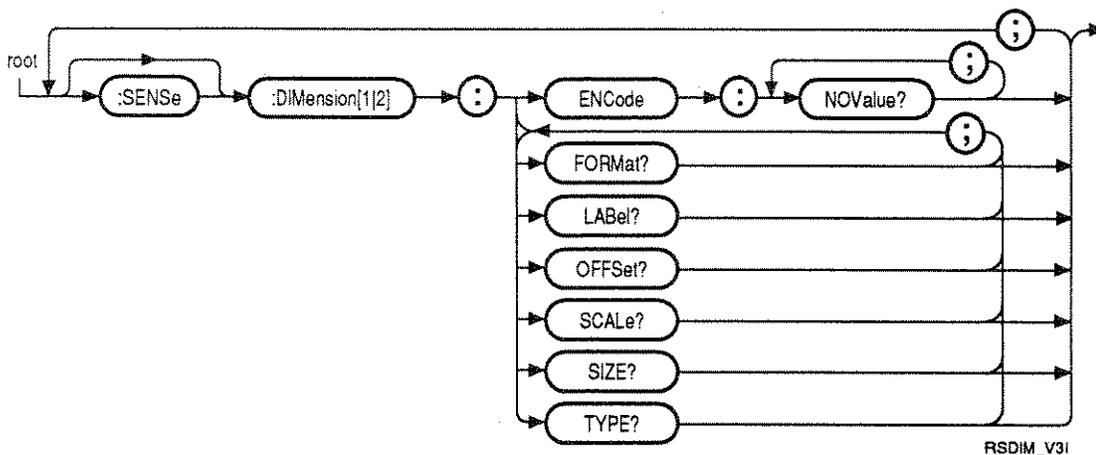


Figure 2-18. :DIMension[1|2] Subtree Commands Syntax Chart

[[:SENSe]:DIMension[1 | 2]:ENCode:NOValue?

[[:SENSe]:DIMension[1 | 2]:ENCode:NOValue?

This query returns a numeric value (-1 or 0) that is to be treated as implying “no value recorded” in the :SENS:DATA? response (vs Time or Fast Histogram data).

Once you know the value, there is no other reason to use this command because the value is static.

How this value is used is described in chapter 5, “Data Transfer.”

Syntax [:SENSe]:DIMension[1 | 2]:ENCode:NOValue?

Example OUTPUT 712;":DIM:ENC:NOV?"

- Comments**
- Query only.
 - If vs Time data is being sensed, the :SENS:DIM1:ENC:NOV? response is -1. This indicates that a value of -1 in the :SENS:DATA? response represents that no measurement was acquired for the corresponding sample position.
 - If Fast Histogram data is being sensed, the :SENS:DIM1:ENC:NOV? response is 0. This indicates that a data value of 0 in the :SENS:DATA? response represents that no measurements were recorded for the corresponding histogram bin.

Related Front-Panel Keys None

[:SENSe]:DIMension[1|2]:FORMat?

[:SENSe]:DIMension[1|2]:FORMat?

This query returns a literal with an integer that specifies the data format that is used to format the :SENS:DATA? query response.

| | |
|-----------------|--|
| Syntax | <code>[:SENSe]:DIMension[1 2]:FORMat?</code> |
| Example | <code>OUTPUT 712;":DIM:FORM?"</code> |
| Comments | <ul style="list-style-type: none">• Use the :FORMat[:DATA] ASC BLOC command to specify the desired format.• If :FORM ASC is used, this query responds with ASC,+5 for vs Time (indicating that each ASCII formatted <NR1> value has up to 5 digits) or with ASC,+15 for Fast Histogram (indicating that each ASCII formatted floating point value has 15 digits).• If :FORM BLOC is used, this query responds with INT,+16 for vs Time (indicating that the measurement data is transferred in a definite length block as signed integers of 16 bits (2 bytes) per measurement sample). Or, it responds with REAL,+64 for Fast Histogram (indicating that the measurement data is transferred in a definite length block as IEEE 754 floating point numbers of 64 bits (8 bytes) per measurement sample).• See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using this query.• Query only. |

Related Front-Panel Keys None

[[:SENSe]:DIMension[1|2]:LABel?

[[:SENSe]:DIMension[1|2]:LABel?

This query returns a literal corresponding to the :SENS:DATA? response that uniquely distinguishes the dimension. The response is shown in the following table:

| Measurement | DIM1:LAB? Response | DIM2:LAB? Response |
|------------------------------|--------------------|--------------------|
| Frequency vs Time | FREQ | TIME |
| Time Interval vs Time | TINT | TIME |
| Fast Frequency Histogram | FPR | FBIN |
| Fast Time Interval Histogram | TIPR | TIB |

How this command is used is described in chapter 5, "Data Transfer."

Syntax [[:SENSe]:DIMension[1|2]:LABel?

Example OUTPUT 712;":DIM:LAB?"

- Comments**
- See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using this query.
 - Query only.

Related Front-Panel Keys None

[[:SENSe]:DIMension[1|2]:OFFSet?

[[:SENSe]:DIMension[1|2]:OFFSet?

This query returns the offset value to be added to the dimension's values returned by the :SENS:DATA? query. The scaling and offset values are defined in the following table:

| Dimension | Equation | Where |
|---------------|--|---|
| DIM1 (Y-axis) | $Y = (\text{scale} * D) + \text{offset}$ | D is a value returned by the :SENS:DATA? query. |
| DIM2 (X-axis) | $X = (\text{scale} * S) + \text{offset}$ | S = 1 to value returned by the :SENS:DIM:SIZE? query. |

Syntax [[:SENSe]:DIMension[1|2]:OFFSet?

Example OUTPUT 712;":DIM:OFFS?"

- Comments**
- The scale value is obtained using the :SENSe:DIMension[1|2]:SCALE? query.
 - For Fast Histogram, DIM1 always returns a 0 because :SENS:DATA? returns probability values which do not need to have a Y-axis offset applied to them.
 - See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using this query.
 - Query only.

Related Front-Panel Keys None

[[:SENSe]:DIMension[1|2]:SCALe?

[[:SENSe]:DIMension[1|2]:SCALe?

This query returns the scale factor to be applied to the dimension's values returned by the :SENS:DATA? query. The scaling and offset values are defined in the following table:

| Dimension | Equation | Where |
|---------------|--|---|
| DIM1 (Y-axis) | $Y = (\text{scale} * D) + \text{offset}$ | D is a value returned by the :SENS:DATA? query. |
| DIM2 (X-axis) | $X = (\text{scale} * S) + \text{offset}$ | S = 1 to value returned by the :SENS:DIM:SIZE? query. |

Syntax [[:SENSe]:DIMension[1|2]:SCALe?

Example OUTPUT 712;":DIM:SCAL?"

- Comments**
- The offset value is obtained using the :SENSe:DIMension[1|2]:OFFSet? query.
 - For Fast Histogram, DIM1 always returns a 1 because :SENS:DATA? returns probability values which do not need to have a Y-axis scale factor applied to them.
 - See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using this query.
 - Query only.

Related Front-Panel Keys None

[[:SENSe]:DIMension[1 | 2]:SIZE?

[[:SENSe]:DIMension[1 | 2]:SIZE?

This query returns the number of values that will be in the :SENS:DATA? response as shown in the following table. The number returned is the total number of vs. Time display pixels. The number includes pixels that have no measurement value associated with them. These “no measurement” values are represented with a value of -1 in the output data.

| Condition | :DIM1:SIZE? and :DIM2:SIZE? Response |
|----------------------------|---|
| (vs Time) SWE:TIME:PAN OFF | 450 |
| (vs Time) SWE:TIME:PAN ON | 450n, where n= number of screens in Panorama (with standard memory, n= 1 to 20; with extended memory, n = 1 to 80). |
| Fast Histogram | Less than or equal to 450 |

Syntax [:SENSe]:DIMension[1 | 2]:SIZE?

Example OUTPUT 712;":DIM:SIZE?"

- Comments**
- If there is no valid data, :SIZE? returns 0.
 - This query should be used when the Analyzer is idle because the validity of the data is stable at that time.
 - See chapter 5, “Data Transfer” and appendix D, “Programming Examples” for additional information and examples of using this query.
 - Query only.

Related Front-Panel Keys None

[[:SENSe]:DIMension[1|2]:TYPE?

[[:SENSe]:DIMension[1|2]:TYPE?

This query returns a literal that indicates whether the dimension is implicit (IMPL) or explicit (EXPL) for the :SENS:DATA? response.

Syntax [:SENSe]:DIMension[1|2]:TYPE?

Example OUTPUT 712;":DIM:TYPE?"

- Comments**
- See chapter 5, "Data Transfer" and appendix D, "Programming Examples" for additional information and examples of using this query.
 - :SENS:DIM1:TYPE? (Y-dimension of vs Time or Fast Histogram) is always explicit and :SENS:DIM2:TYPE? (X-dimension of vs Time or Fast Histogram) is always implicit. This indicates that one dimension of data is always returned by :SENS:DATA? and this data corresponds to the Y-dimension of the sensed data (vs Time or Fast Histogram).
 - Query only.

Related Front-Panel Keys None

[[:SENSe]:EVENT[1|2] Subtree

[[:SENSe]:EVENT[1|2] Subtree

The :EVENT[1|2] subtree commands define the “input event” (voltage threshold). For frequency measurements, this is the event that is counted. For time interval measurements, this is the start/stop edge. EVENT1 corresponds to input A and EVENT2 corresponds to input B. There is no input control for the optional channel C.

Subtree Syntax *Figure 2-19 is the syntax diagram for the :EVENT[1|2] subtree.*

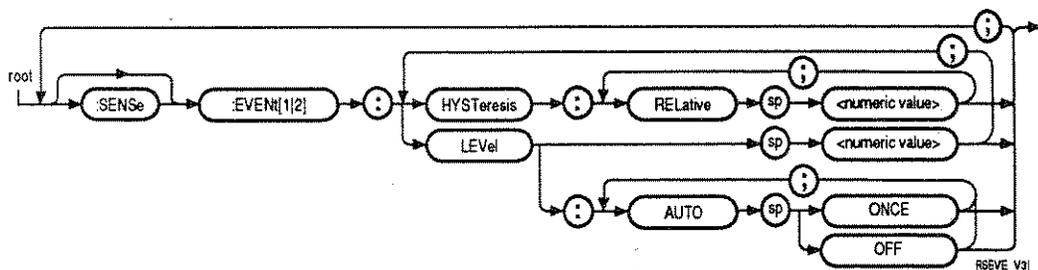


Figure 2-19. :EVENT[1\2] Subtree Commands Syntax Chart

[[:SENSe]:EVENT[1|2]:HYSTeresis:RELative

[[:SENSe]:EVENT[1|2]:HYSTeresis:RELative

This command specifies the size of the hysteresis window as a percentage of the allowable hysteresis. For example, 0% is the minimum hysteresis setting and 100% is the maximum hysteresis setting.

Syntax `[[:SENSe]:EVENT[1|2]:HYSTeresis:RELative <numeric value>`

Example `OUTPUT 712;":EVENT2:HYST:REL 30"`

- Comments**
- EVENT1 corresponds to input A and EVENT2 corresponds to input B. There is no input control for the optional channel C.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - **Units = Percent [PCT]**
 - ***RST = 0**

Related Front-Panel Keys See *figure 1-6*.

[[:SENSe]:EVENT[1|2]:LEVel

[[:SENSe]:EVENT[1|2]:LEVel

The :EVENT[1|2]:LEVel keyword acts as either a subtree or a command to define the input event level. If a numeric value is entered, it acts as a command. If it is used with the AUTO command it acts as a subtree.

When a numeric value is entered, this command sets the input event level at the center of the hysteresis window. The actual input event is at the top of the hysteresis window for the POSitive slope and at the bottom of the hysteresis window for the NEGative slope.

Syntax [:SENSe]:EVENT[1|2]:LEVel <numeric value>

Example OUTPUT 712;":EVEN2:LEV 5"

- Comments**
- EVENT1 corresponds to input A and EVENT2 corresponds to input B. There is no input control for the optional channel C.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - **Units = Volts [V]**
 - ***RST = 0**

Related Front-Panel Keys See *figure 1-6*

[[:SENSE]:EVENT[1|2]:LEVEL:AUTO

[[:SENSE]:EVENT[1|2]:LEVEL:AUTO

The :AUTO ONCE command activates the “find 50% voltage threshold” function for the specified input. On completion, AUTO is set off. The Analyzer performs only a single-shot auto-threshold detection.

Syntax [[:SENSE]:EVENT[1|2]:LEVEL:AUTO ONCE|OFF

Example OUTPUT 712;[:EVEN2:LEV:AUTO ONCE"

- Comments**
- EVENT1 corresponds to input A and EVENT2 corresponds to input B. There is no input control for the optional channel C.
 - After completion of the :AUTO ONCE command, the SENSE:EVENT[1|2]:LEVEL? query will return the corresponding voltage value.
 - Errors +2009 and +2010 are generated when no 50% voltage threshold is found on channel A and B, respectively.
 - Event; no query.
 - *RST = OFF

Related Front-Panel Keys See *figure 1-6*.

[[:SENSe]:FREQuency Subtree

[[:SENSe]:FREQuency Subtree

The [[:SENSe]:FREQuency subtree commands control the frequency measuring capabilities of the Analyzer.

Syntax *Figure 2-20* is the syntax diagram for the :FREQuency subtree.

[[:SENSe]:FREQuency Subtree

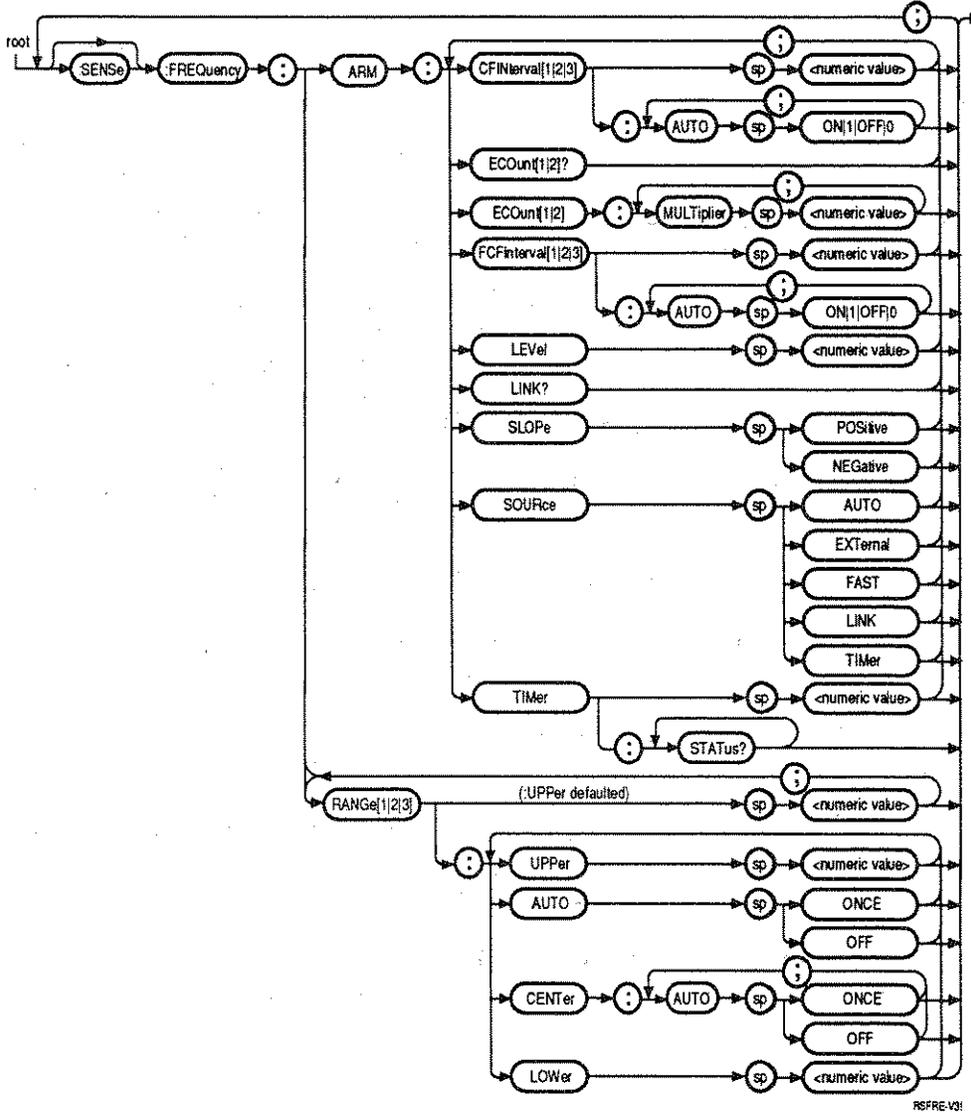


Figure 2-20. :FREQuency Subtree Commands Syntax Chart

[[:SENSe]:FREQuency:ARM:CFINterval[1|2|3]

[[:SENSe]:FREQuency:ARM:CFINterval[1|2|3]

When a numeric value is entered, this command sets the sampling interval at center frequency for :SENS:FREQ:ARM:SOUR AUTO. The sampling interval at center is the time between two adjacent samples of a measurement at the center frequency of the selected frequency range. 1 = A, 2 = B, and 3 = C.

The :FREQuency:ARM:CFINterval[1|2|3] keywords act as either a subtree or a command to define the sampling interval at center frequency. If a numeric value is entered, it acts as a keyword. If it is used with the AUTO command it acts as a subtree.

Syntax [:SENSe]:FREQuency:ARM:CFINterval[1|2|3] <numeric value>

Example OUTPUT 712;":FREQ:ARM:CFIN 4.762E-6"

- Comments**
- Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - **Units = Seconds [S]**
 - Specifying a <numeric value> causes :SENS:FREQ:ARM:CFIN:AUTO to be set to OFF.
 - ***RST = 5E-6**

Related Front-Panel Keys See *figure 1-15*.

[[:SENSe]:FREQuency:ARM:CFINterval[1|2|3]:AUTO

[[:SENSe]:FREQuency:ARM:CFINterval[1|2|3]:AUTO

This command controls the coupling of the sampling interval at center frequency to :SENS:SWEep:TIME:SPAN when :SENS:FREQ:ARM:SOUR AUTO. 1 = A, 2 = B, and 3 = C.

Syntax [:SENSe]:FREQuency:ARM:CFINterval[1|2|3]:AUTO
 ON|1|OFF|0

Example OUTPUT 712;":FREQ:ARM:CFIN:AUTO ON"

Comments • *RST = ON

**Related
Front-Panel Keys** See *figure 1-15*.

[[:SENSe]:FREQuency:ARM:ECOunt[1|2]?

[[:SENSe]:FREQuency:ARM:ECOunt[1|2]?

This query retrieves the actual event count used for sampling when sensing frequency A or B and edge sampling on current measurement channel. This value is the smallest multiple of :SENS:FREQ:ARM:ECO[1|2]:MULT that can be used within the specified frequency range (:SENS:FREQ:RANG[1|2]:UPP and :SENS:FREQ:RANG[1|2]:LOW). 1 = A and 2 = B.

Syntax [:SENSe]:FREQuency:ARM:ECOunt[1|2]?

Example OUTPUT 712;":FREQ:ARM:ECO?"

- Comments**
- This value should only be queried when sensing frequency A or B (:SENS:FUNC "FREQ[1|2]") or :SENS:FUNC "FREQ[1|2]:HIST" and edge sampling on current measurement channel (:SENS:FREQ:ARM:SOUR LINK). If this query is used with any other configuration, error -221 is generated and a value of 0 is returned.
 - Query only.

Related Front-Panel Keys See *figure 1-15*.

[[:SENSe]:FREQuency:ARM:ECOunt[1|2]:MULTiplier

[[:SENSe]:FREQuency:ARM:ECOunt[1|2]:MULTiplier

This command sets the edge count multiplier when sensing frequency A or B and edge sampling on current measurement channel. Applies only when :SENS:FREQ:ARM:SOUR LINK and frequency A or B are selected. 1 = A and 2 = B.

The edge count multiplier for channel A must be an even number. Channel B can be even or odd. The actual edge count used (:SENS:FREQ:ARM:ECO[1|2]?) will be the smallest multiple of the value specified that can be used by the current frequency range (:SENS:FREQ:RANG[1|2]:UPP and :SENS:FREQ:RANG[1|2]:LOW).

Syntax [:SENSe]:FREQuency:ARM:ECOunt[1|2]:MULTiplier
 <numeric value>

Example OUTPUT 712;":FREQ:ARM:ECO:MULT 4"

- Comments**
- ECO:MULT must be set to a positive value of 2 or greater (and must be even) for channel A and 1 or greater for channel B.
 - Can use MINimum or MAXimum instead of <numeric value>.
 - *RST= ECO1 is 2 and ECO2 is 1.

Related Front-Panel Keys See *figure 1-15*.

[[:SENSe]:FREQuency:ARM:FCFInterval[1|2|3]

[[:SENSe]:FREQuency:ARM:FCFInterval[1|2|3]

When a numeric value is entered, this command sets the sampling interval at center frequency for :SENS:FREQ:ARM:SOUR FAST. The sampling interval at center is the time between two adjacent samples of a measurement at the center frequency of the selected frequency range. 1 = A, 2 = B, and 3 = C.

The :FREQuency:ARM:FCFInterval[1|2|3] keywords act as either a subtree or a command to define the sampling interval at center frequency. If a numeric value is entered, it acts as a keyword. If it is used with the AUTO command it acts as a subtree.

Use this command when you want to control the value of interval at center so that the value is not affected by changing the center frequency, a changing input frequency, or the time/div setting. Typically, the Auto setting for Interval at Center provides the best results.

Syntax [:SENSe]:FREQuency:ARM:FCFInterval[1|2|3] <numeric value>

Example OUTPUT 712;":FREQ:ARM:FCFI3 2.5E-6"

- Comments**
- The example command switches the Interval at Center control for Fast sampling to Manual and sets the interval to 2.5 μ s.
 - **Units = Seconds [S]**
 - ***RST = 5E-6**

Related Front-Panel Keys See *figure 1-15*.

[[:SENSe]:FREQuency:ARM:FCFInterval[1|2|3]:AUTO

[[:SENSe]:FREQuency:ARM:FCFInterval[1|2|3]:AUTO

This command controls the coupling of the sampling interval at center frequency to :SENS:SWEep:TIME:SPAN when :SENS:FREQ:ARM:SOUR FAST. 1 = A, 2 = B, and 3 = C.

Syntax [:SENSe]:FREQuency:ARM:FCFInterval[1|2|3]:AUTO
ON|1|OFF|0

Example OUTPUT 712;":FREQ:ARM:FCFI3:AUTO OFF"

- Comments**
- The example command sets the Fast sampling mode Interval at Center control to Manual for channel C.
 - *RST = ON

Related Front-Panel Keys See *figure 1-15*.

[[:SENSe]:FREQuency:ARM:LEVel

[[:SENSe]:FREQuency:ARM:LEVel

This command specifies the voltage level of the external arm signal used for external edge sampling of frequency measurements when :SENS:FREQ:ARM:SOUR is EXT, or :SENS:FREQ:ARM:SOUR LINK with frequency C.

Syntax [:SENSe]:FREQuency:ARM:LEVel <numeric value>

Example OUTPUT 712;[:FREQ:ARM:LEV 0"

- Comments**
- This level is used only if the source for the event detector is external.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - Related Command is :SENS:FREQ:ARM:SOUR EXT.
 - This value is coupled to :SENS:HIST:ARM:LEV, :SENS:TINT:ARM:LEV, and :TRIG:LEV in that they all share the same value. Changing one, changes all others to the same value.
 - Allowed values are -1.3 and -0.13 (with 10:1 probe) for ECL, 0, and 1.5 and 0.15 (with 10:1 probe) for TTL.
 - Units = Volts [V]
 - *RST= 1.5

Related Front-Panel Keys See *figure 1-14*.

[[:SENSe]:FREQuency:ARM:LINK?

[[:SENSe]:FREQuency:ARM:LINK?

This query returns INT (for A), INT2 (for B), or EXT (for C). This response indicates which edge is used for edge sampling on frequency measurements when :SENS:FREQ:ARM:SOUR LINK is chosen.

Syntax [:SENSe]:FREQuency:ARM:LINK?

Example OUTPUT 712;"[:FREQ:ARM:LINK?"

Comments

- Related Command is :SENS:FREQ:ARM:SOUR LINK.
- Query only.

**Related
Front-Panel Keys** None

[[:SENSe]:FREQuency:ARM:SLOPe

[[:SENSe]:FREQuency:ARM:SLOPe

This command specifies the slope of the external arm signal used for external edge sampling of frequency measurements. Only applies when :SENS:FREQ:ARM:SOUR EXT, or :SENS:FREQ:ARM:SOUR LINK and frequency C are selected.

| Parameter Name | Description |
|----------------|--------------------------------------|
| POSitive | Selects the positive (rising) edge. |
| NEGative | Selects the negative (falling) edge. |

Syntax [:SENSe]:FREQuency:ARM:SLOPe POSitive | NEGative

Example OUTPUT 712;":FREQ:ARM:SLOP POS"

- Comments**
- Related Command is :SENS:FREQ:ARM:SOUR EXT.
 - This command is only used with external edge sampling.
 - *RST = POS

Related Front-Panel Keys See *figure 1-15*.

[[:SENSe]:FREQuency:ARM:SOURce

[[:SENSe]:FREQuency:ARM:SOURce

This command selects a source for the frequency measurement sampling.

The literal parameter choices are listed in the following table:

| Parameter Name | Description |
|----------------|---|
| AUTO | Sets measurement sampling to occur automatically. The AUTO setting is coupled to other instrument settings. Using this parameter is equivalent to selecting Auto on the Sampling menu. |
| EXTernal | Sets measurement sampling to edge sampling with the External Arm signal. Using this parameter is equivalent to selecting Edge and then Ext on the Sampling menu. |
| FAST | Sets fast measurement sampling to occur automatically. The FAST setting is coupled to other instrument settings. Using this parameter is equivalent to selecting Fast on the Sampling menu. |
| LINK | Specifies edge sampling (which is coupled to current measurement function, :SENS:FUNC). Frequency A uses input A. Frequency B uses input B. Frequency C uses External Arm Signal. |
| TIMer | Sets measurement sampling to occur repetitively in time. Each subsequent arm is referenced to the previous arm. |

Syntax [[:SENSe]:FREQuency:ARM:SOURce <parameter>

[[:SENSe]:FREQuency:ARM:SOURce

- Example** OUTPUT 712;":FREQ:ARM:SOUR EXT"
- Comments**
- Error -221 will be generated and the source will be unaffected if EXT is requested when EXT is already specified as either a trigger source (TRIG:SOUR) or as a histogram arm source (:SENS:HIST:ARM:SOUR).
 - The source is changed from EXT to LINK (frequency A or B) or AUTO (frequency C) when it is necessary to resolve a conflict with :TRIG:SOUR EXT or :SENS:HIST:ARM:SOUR EXT.
 - ***RST= AUTO**
 - The Fast sampling mode provides an effective sampling rate of up to 7.5 MHz for repetitive signals. It should normally be used for repetitive signals.
 - Fast sampling is only available for Frequency measurements.

NOTE _____

When set to Fast sampling, the HP 53310A makes consecutive acquisitions of 17 measurement samples until enough data is collected to fill the display. This can be up to 225 measurements across the vs. Time display representing 14 separate acquisitions. Each of these acquisitions requires a trigger. For example, if 225 measurements were needed and the trigger condition was an edge at the External Arm input, 14 trigger edges, along with the signal being measured, would be needed before enough measurement data would be acquired and displayed.

**Related
Front-Panel Keys**

See figure 1-15.

[[:SENSe]:FREQuency:ARM:TIMer

This command specifies the sampling time used for time sampling of frequency measurements. This value only applies when :SENS:FREQ:ARM:SOUR is TIM.

The :FREQuency:ARM:TIMer keyword acts as either a subtree or a command to define the sampling time used for time sampling. If a numeric value is entered, it acts as a command. If it is used with the :STATus? query it acts as a subtree.

Syntax [:SENSe]:FREQuency:ARM:TIMer <numeric value>

Example OUTPUT 712;":FREQ:ARM:TIM 5E-3"

- Comments**
- TIMER must be a positive value.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - Units = Seconds [S]
 - *RST = 50.0E-6

Related Front-Panel Keys See *figure 1-15*.

`[[:SENSe]:FREQuency:ARM:TIMer:STATus?`

`[[:SENSe]:FREQuency:ARM:TIMer:STATus?`

This query returns a 0 or 1.

0 = Selected frequency sampling time will be used
(`:SENS:FREQ:ARM:TIM <numeric value>`).

1 = Selected frequency sampling time is less than the approximate minimum sampling interval. The actual sampling time may be a multiple of the selected value. This can only occur for selected sampling times less than 10E-6 seconds. See chapter 9 "Sampling Menu" of the *Operating Manual* for additional information.

Syntax `[[:SENSe]:FREQuency:ARM:TIMer:STATus?`

Example `OUTPUT 712;":FREQ:ARM:TIM:STAT?"`

Comments

- This value should only be queried when sensing Frequency A, B, or C (`:SENS:FUNC "FREQ[1|2|3]"` or `:SENS:FUNC "FREQ[1|2|3]:HIST"`). If this query is used with any other configuration, error -221 is generated and a value of 0 is returned.

**Related
Front-Panel Keys** None

[[:SENSe]:FREQuency:RANGe[1|2|3][:UPPer]

[[:SENSe]:FREQuency:RANGe[1|2|3][:UPPer]

This command specifies the maximum frequency expected on the selected channel. 1 = A, 2 = B, and 3 = C.

Syntax `[[:SENSe]:FREQuency:RANGe[1|2|3][:UPPer] <numeric value>`

Example `OUTPUT 712;[:FREQ:RANG 10 MHZ"`

- Comments**
- RANG3 is available only if Option 30 (C-channel) is installed.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - Units = Hertz [HZ]
 - *RST = RANG1 is 125E+6, RANG2 is 70E+6, and RANG3 is 1.375E+9

Related Front-Panel Keys See *figure 1-7*.

[[:SENSe]:FREQuency:RANGe[1|2|3]:AUTO

[[:SENSe]:FREQuency:RANGe[1|2|3]:AUTO

When AUTO ONCE is used, this command invokes an event that analyzes the input signal and sets appropriate upper (:SENS:FREQ:RANG[1|2|3]:UPP) and lower (:SENS:FREQ:RANG[1|2|3]:LOW) values. Make sure your signal is connected to the appropriate input and the input is configured before using this command. 1 = A, 2 = B, and 3 = C.

Syntax [:SENSe]:FREQuency:RANGe[1|2|3]:AUTO ONCE|OFF

Example OUTPUT 712;":FREQ:RANG2:AUTO ONCE"

- Comments**
- RANG3 is available only if Option 30 (C-channel) is installed.
 - Event; with query.
 - AUTO ONCE is a single shot event. Therefore, after it completes, AUTO is set to OFF.
 - ***RST = OFF for all three ranges**

Related Front-Panel Keys See *figure 1-7*.

[[:SENSe]:FREQuency:RANGe[1|2|3]:CENTer:AUTO

[[:SENSe]:FREQuency:RANGe[1|2|3]:CENTer:AUTO

When AUTO ONCE is used, this command invokes an event that analyzes the input signal and sets appropriate upper (:SENS:FREQ:RANG[1|2|3]:UPP) and lower (:SENS:FREQ:RANG[1|2|3]:LOW) values but maintains the current span between upper and lower frequencies. Make sure your signal is connected to the appropriate input and the input is configured before using this command. 1 = A, 2 = B, and 3 = C.

Syntax [:SENSe]:FREQuency:RANGe[1|2|3]:CENTer:AUTO
ONCE|OFF

Example OUTPUT 712;":FREQ:RANG2:CENT:AUTO ONCE"

- Comments**
- RANG3 is available only if Option 30 (C-channel) is installed.
 - AUTO ONCE is a single shot event. Therefore, after it completes, AUTO is set to OFF.
 - Event; with query.
 - ***RST = OFF for all three ranges**

Related Front-Panel Keys See *figure 1-7*.

[[:SENSe]:FREQuency:RANGe[1|2|3]:LOWer

[[:SENSe]:FREQuency:RANGe[1|2|3]:LOWer

This command specifies the minimum frequency expected on the selected channel. 1 = A, 2 = B, and 3 = C.

Syntax `[[:SENSe]:FREQuency:RANGe[1|2|3]:LOWer <numeric value>`

Example `OUTPUT 712;[:FREQ:RANG:LOW 9 MHZ"`

- Comments**
- RANG3 is available only if Option 30 (C-channel) is installed.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - Units = Hertz [HZ]
 - *RST = RANG1 is 75E+6, RANG2 is 30E+6, and RANG3 is 625E+6.

Related Front-Panel Keys See *figure 1-7*.

[[:SENSe]:FUNction

The [[:SENSe]:FUNction <sensor function string> command selects the sensor functions used by the Analyzer. The <sensor function string> is specified as a quoted string. For example:

:SENS:FUNC "FREQ2:HIST" selects the Fast Frequency Histogram on channel B.

The query response of :SENS:FUNC? returns the function as <STRING RESPONSE DATA>. The query returns the short form mnemonics and omits any optional portions. For example:

:SENS:FUNC? returns the quoted string "FREQ2:HIST" for the Fast Frequency Histogram on channel B.

The following table lists the sensor functions available on the Analyzer:

| Function String | Description |
|------------------------------|---|
| "FREQuency[1 2 3]" | Selects the Frequency vs Time function (1 = channel A, 2 = B, and 3 = C). |
| "FREQuency[1 2 3]:HISTogram" | Selects Fast Histogram for Frequency (1 = A, 2 = B, and 3 = C). |
| "TINTerval" | Selects Time Interval A→B vs Time function. |
| "TINTerval:HISTogram" | Selects Fast Histogram for Time Interval A→B. |

[[:SENSe]:FUNction

Syntax Figure 2-21 is the syntax diagram for the :FUNction command.

Example OUTPUT 712;":FUNC ""TINT:HIST""

- Comments**
- Error -224 is generated if a string parameter is received that does not correspond to one of the allowed functions.
 - To enable/disable the Histogram from vs Time, see :CALC:HIST[:STAT] command.
 - Changing the function can result in either the :SENS:FREQ:ARM:SOUR or :SENS:TINT:ARM:SOUR being changed from EXT to LINK (frequency A or B), from EXT to AUTO (frequency C), or from EXT to INT1 (time interval). This is done if it is necessary to resolve the conflict of :TRIG:SOUR or :SENS:HIST:ARM:SOUR already using EXT.
 - *RST = "FREQ"

Related Front-Panel Keys See figure 1-5.

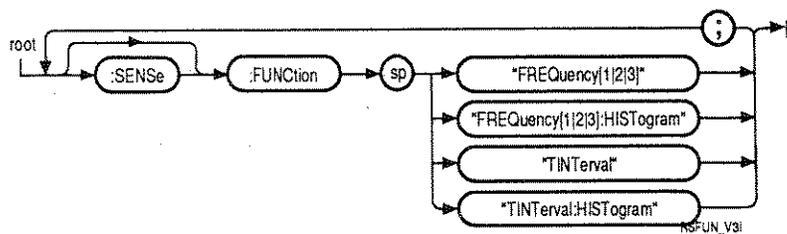


Figure 2-21. :FUNction Subtree commands Syntax chart

[[:SENSe]:HISTogram Subtree

The [[:SENSe]:HISTogram subtree commands control the Fast Histogram characteristics. See :CALC:HIST subtree commands for Histogram from vs Time capabilities.

Subtree Syntax Figure 2-22 is the syntax diagram for the :HISTogram subtree.

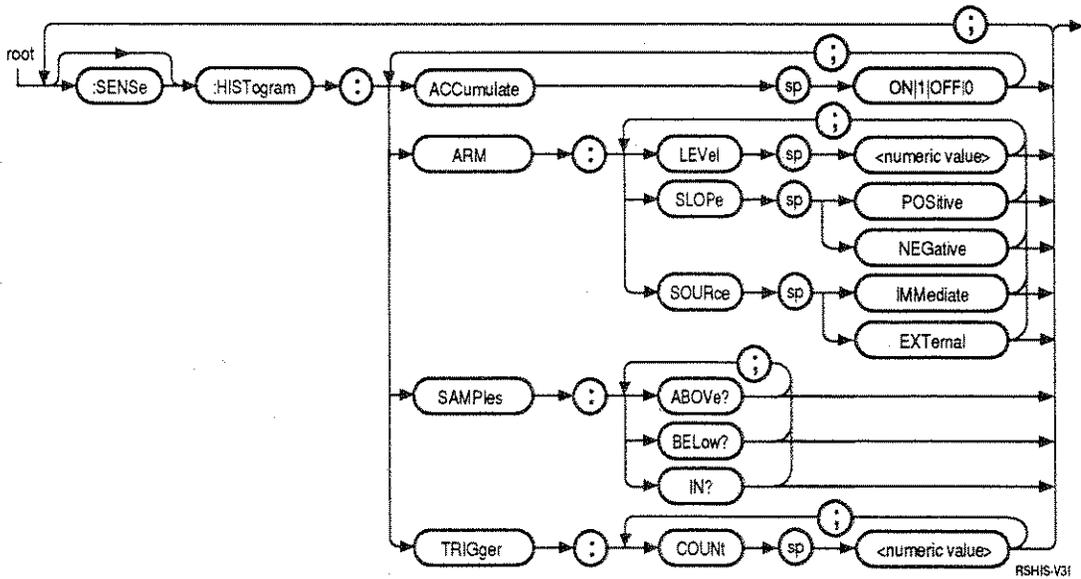


Figure 2-22. :HISTogram Subtree Commands Syntax Chart

[[:SENSe]:HISTogram:ACCumulate

[[:SENSe]:HISTogram:ACCumulate

This command specifies if the Fast Histogram data is accumulated between updates.

| | |
|---------------------------------|---|
| Syntax | <code>[[:SENSe]:HISTogram:ACCumulate ON 1 OFF 0</code> |
| Example | <code>OUTPUT 712;":HIST:ACC ON"</code> |
| Comments | <ul style="list-style-type: none">• Coupled to <code>:CALC:HIST:ACC</code> in that they share the same value (both are either ON or OFF). Changing one changes the other.• *RST = OFF |
| Related Front-Panel Keys | See <i>figure 1-14</i> . |

[[:SENSe]:HISTogram:ARM:LEVel

[[:SENSe]:HISTogram:ARM:LEVel

This command specifies the voltage level of the external arming signal used to start Fast Histogram acquisitions when :SENS:HIST:ARM:SOUR is EXT.

Syntax [:SENSe]:HISTogram:ARM:LEVel <numeric value>

Example OUTPUT 712;":HIST:ARM:LEV 0"

- Comments**
- Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - Only used when :SENS:HIST:ARM:SOUR is EXT.
 - This value is coupled to :SENS:FREQ:ARM:LEV, :SENS:TINT:ARM:LEV, and :TRIG:LEV in that they all share the same value. Changing one , changes all others to the same value.
 - Allowed values are -1.3 and -0.13 (with 10:1 probe) for ECL, 0, and 1.5 and 0.15 (with 10:1 probe) for TTL.
 - **Units = Volts [V]**
 - ***RST = 1.5**

Related Front-Panel Keys See *figure 1-6*.

[[:SENSe]:HISTogram:ARM:SLOPe

[[:SENSe]:HISTogram:ARM:SLOPe

This command specifies the slope of the external arming signal used to start Fast Histogram acquisitions when :SENS:HIST:ARM:SOUR is EXT.

| Parameter Name | Description |
|----------------|--------------------------------------|
| POSitive | Selects the positive (rising) edge. |
| NEGative | Selects the negative (falling) edge. |

Syntax [[:SENSe]:HISTogram:ARM:SLOPe POSitive | NEGative

Example OUTPUT 712;":HIST:ARM:SLOP POS"

- Comments**
- Used only when :SENS:HIST:ARM:SOUR is EXT.
 - This value is coupled to :TRIG:SLOP in that they are the same value. Changing one, changes the other to the same value.
 - *RST = POS

Related Front-Panel Keys See *figure 1-14*.

[[:SENSe]:HISTogram:ARM:SOURce

[[:SENSe]:HISTogram:ARM:SOURce

This command specifies how to synchronize the Fast Histogram acquisition.

| Parameter Name | Description |
|----------------|--|
| IMMediate | Fast Histogram measurements start immediately. |
| EXTernal | Fast Histogram measurements start when the external arming requirements are satisfied. |

Syntax `[[:SENSe]:HISTogram:ARM:SOURce IMMEDIATE | EXTernal`

Example `OUTPUT 712;":HIST:ARM:SOUR IMM"`

- Comments**
- Only applies when sensing Fast Histogram. That is, `:SENS:FUNC "FREQ[1|2|3]:HIST"` or `:SENS:FUNC "TINT:HIST"` were used.
 - The external arm signal cannot be used for Fast Histogram synchronization (`:SENS:HIST:ARM:SOUR`) and measurement sampling (`:SENS:FREQ:ARM: SOUR` or `:SENS:TINT:ARM:SOUR`) simultaneously. This conflict is resolved by allowing the `:SENS:HIST:ARM:SOUR` to be set to `EXT` and changing the measurement sampling to a source other than `EXT`.
 - ***RST = IMM**

Related Front-Panel Keys See *figure 1-14*.

[[:SENSe]:HISTogram:SAMPles:ABOVe?"]

[[:SENSe]:HISTogram:SAMPles:ABOVe?"]

When sensing a Fast Frequency Histogram, this query returns the number of samples above
:SENS:FREQ:RANG[1|2|3]:UPP.

When sensing a Fast Time Interval Histogram, this query returns the number of samples above
:SENS:TINT:RANG:UPP.

Syntax [[:SENSe]:HISTogram:SAMPles:ABOVe?

Example OUTPUT 712;[:HIST:SAMP:ABOV?"]

- Comments**
- The :HIST:SAMP:ABOV? command has the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - The response is 0 if queried when sensing vs Time (error -221) or no valid Fast Histogram data exists (error -230).
 - Query only.

Related Front-Panel Keys None.

[[:SENSe]:HISTogram:SAMPles:BELOW?

[[:SENSe]:HISTogram:SAMPles:BELOW?

When sensing a Fast Frequency Histogram, this query returns the number of samples below
:SENS:FREQ:RANG[1|2|3]:LOW.

When sensing a Fast Time Interval Histogram, this query returns the number of samples below
:SENS:TINT:RANG:LOW.

Syntax [[:SENSe]:HISTogram:SAMPles:BELOW?

Example OUTPUT 712;":HIST:SAMP:BEL?"

- Comments**
- The :HIST:SAMP:BEL? command has the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - The response is 0 if queried when sensing vs Time (error -221) or no valid Fast Histogram data exists (error -230).
 - Query only.

Related Front-Panel Keys None.

[[:SENSe]:HISTogram:SAMPles:IN?

[[:SENSe]:HISTogram:SAMPles:IN?

When sensing a Fast Frequency Histogram, this query returns the number of samples falling within :SENS:FREQ:RANG[1|2|3]:LOW and :SENS:FREQ:RANG[1|2|3]:UPP.

When sensing a Fast Time Interval Histogram, this query returns the number of samples falling within :SENS:TINT:RANG:LOW and :SENS:TINT:RANG:UPP.

Syntax [[:SENSe]:HISTogram:SAMPles:IN?

Example OUTPUT 712;":HIST:SAMP:IN?"

- Comments**
- The :HIST:SAMP:IN? command has the effect of holding off further commands from being processed until a measurement completes. This holdoff action can only be canceled by the measurement completing, device clear, or power-on.
 - The response is 0 if queried when sensing vs Time (error -221) or no valid Fast Histogram data exists (error -230).
 - Query only.

Related Front-Panel Keys None.

[[:SENSe]:HISTogram:TRIGger:COUNT

[[:SENSe]:HISTogram:TRIGger:COUNT

This command specifies the number of frequency or time intervals Time Interval A→B vs Time function.erval measurements required for a Fast Histogram update.

This number of measurements must fall within the frequency range (:SENS:FREQ:RANG[1|2|3]:LOW and :SENS:FREQ:RANG[1|2|3]:UPP) or time interval range (:SENS:TINT:RANG:LOW and :SENS:TINT:RANG:UPP).

Syntax [:SENSe]:HISTogram:TRIGger:COUNT <numeric value>

Example OUTPUT 712;":HIST:TRIG:COUN 500"

- Comments**
- Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - *RST = 1000

Related Front-Panel Keys See *figure 1-14*.

[[:SENSe]:ROSCillator:SOURce?

[[:SENSe]:ROSCillator:SOURce?

This query's response indicates the source of the timebase reference. It can be internal (INT), external (EXT), or indeterminate if an error (ERR) has been detected. This is a query only because the Analyzer auto-detects the availability of an external reference oscillator and will use the external reference oscillator if one is detected.

Syntax *Figure 2-23* is the syntax diagram for the :ROSCillator:SOURce command.

Example OUTPUT 712;"ROSC:SOUR?"

Comments

- Query only.

Related Front-Panel Keys None

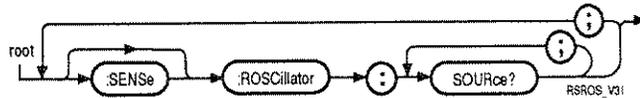


Figure 2-23. :ROSCillator Subtree Commands Syntax Chart

[[:SENSe]:SWEep Subtree

The [[:SENSe]:SWEep subtree commands control the timebase characteristics of the Analyzer.

The timebase parameters are inter-related by a rather involved set of couplings. In order to avoid running into unexpected couplings, it is advisable to set the timebase parameters in the following sequence:

1. Set the timebase reference, panorama enable, main timebase and panorama timebase in any order.
2. Set the panorama delay.
3. Set the main delay.

The order of steps 2 and 3 are only important if panorama is on.

Subtree Syntax *Figure 2-24* is the syntax diagram for the :SWEep subtree.

[[:SENSe]:SWEep Subtree

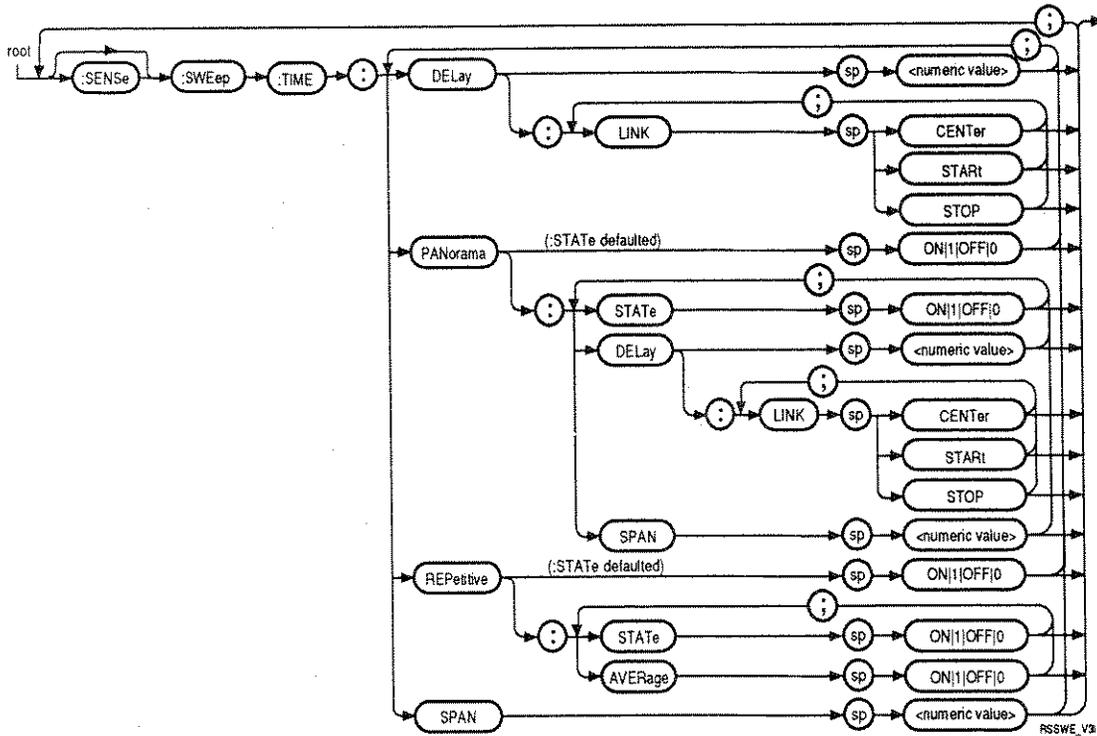


Figure 2-24. :SWEep Subtree Commands Syntax Chart

[[:SENSe]:SWEep:TIME:DElay

This command specifies the main timebase delay. This delay is the interval of time between the trigger event and the delay reference of the main display trace (:SENS:SWE:TIME:DEL:LINK).

The :SWEep:TIME:DElay keyword acts as either a subtree or a command to define the delay. If a numeric value is entered, it acts as a command. If it is used with the LINK command it acts as a subtree.

Syntax [:SENSe]:SWEep:TIME:DElay <numeric value>

Example OUTPUT 712;":SWE:TIME:DEL 5E-3"

- Comments**
- Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - When panorama is enabled, the main delay may be coupled by other timebase parameter changes to keep the main display within the panorama display.
 - When panorama is disabled, changing the main delay may cause the panorama delay to be updated to keep the panorama display around the main display.
 - Modifying this parameter when panorama disabled does not unvalidate existing data.
 - **Units = Seconds [S]**
 - ***RST = 0**

Related Front-Panel Keys See *figures 1-8 and 1-9*.

[[:SENSe]:SWEep:TIME:DELAy:LINK

[[:SENSe]:SWEep:TIME:DELAy:LINK

This command sets the delay reference to the beginning (START), center (CENTer), or end (STOP) of the main display trace (the larger of the two traces when panorama enabled).

The literal parameters (START|STOP|CENTer) correspond to the left, center, and right edge of of the main display trace.

Syntax [:SENSe]:SWEep:TIME:DELAy:LINK
 START|STOP|CENTer

Example OUTPUT 712,":SWE:TIME:DEL:LINK CENT"

- Comments**
- When panorama is disabled, changing the reference may cause the panorama delay to be updated to keep the panorama display around the main display.
 - When panorama is enabled, changing the reference may cause the main delay to be updated to keep the main display within the panorama display.
 - The main delay is specified by :SENS:SWE:TIME:DEL.
 - Coupled to SWE:TIME:PAN:DEL:LINK in that they share the same value.
 - *RST = CENT

Related Front-Panel Keys See *figures 1-8* and *1-9*.

[[:SENSe]:SWEep:TIME:PANorama:[STATe]

[[:SENSe]:SWEep:TIME:PANorama:[STATe]

This command enables or disables the panorama measurement capabilities.

Syntax `[[:SENSe]:SWEep:TIME:PANorama:[STATe] ON | 1 | OFF | 0`

Example `OUTPUT 712;[:SWE:TIME:PAN ON"`

- Comments**
- Turning this command ON affects data transfer in that the `:SENS:DATA?` command and the `:SENS:DIM[1|2]` subtree commands now correspond to Panorama data.
 - When panorama is ON, analysis results and transferred data do not reflect that the data was repetitive (`:SENS:SWE:TIME:REP ON`) or averaged (`:SENS:SWE:TIME:REP:AVER ON`).
 - ***RST =OFF**

**Related
Front-Panel Keys**

See *figure 1-9*.

[[:SENSe]:SWEep:TIME:PANorama:DElay

[[:SENSe]:SWEep:TIME:PANorama:DElay

This command specifies the panorama timebase delay. This delay is the interval of time between the trigger point and the delay reference of the panorama display.

The `:SWEep:TIME:PANorama:DElay` keyword acts as either a subtree or a command to define the delay. If a numeric value is entered, it acts as a command. If it is used with the `LINK` command it acts as a subtree.

Syntax `[[:SENSe]:SWEep:TIME:PANorama:DElay <numeric value>`

Example `OUTPUT 712;":SWE:TIME:PAN:DEL 2E-3"`

- Comments**
- When panorama is disabled, the panorama delay may be coupled by other timebase parameter changes to keep the panorama display around the main display.
 - When panorama is enabled, changing the panorama delay may cause the main delay to be updated to keep the main display within the panorama display.
 - Can use `MINimum` or `MAXimum` instead of `<numeric value> [suffix]`.
 - **Units = Seconds [S]**
 - ***RST = 0**

Related Front-Panel Keys See *figure 1-9*.

[:SENSe]:SWEep:TIME:PANorama:DELAy:LINK

[:SENSe]:SWEep:TIME:PANorama:DELAy:LINK

This command sets the delay reference to the beginning (STARt), center (CENTer), or end (STOP) of the panorama display trace.

The literal parameters (STARt|STOP|CENTer) correspond to the left, center, and right edge of the panorama display trace.

Syntax `[:SENSe]:SWEep:TIME:PANorama:DELAy:LINK
STARt|STOP|CENTer`

Example `OUTPUT 712;":SWE:TIME:PAN:DEL:LINK STAR"`

- Comments**
- ***RST = CENT**
 - When panorama is disabled, changing the reference may cause the panorama delay to be updated to keep the panorama display around the main display.
 - When panorama is enabled, changing the reference may cause the main delay to be updated to keep the main display within the panorama display.
 - The panorama delay is specified by `:SENS:SWE:TIME:PAN:DEL`.
 - Coupled to `SWE:TIME:DEL:LINK` in that they share the same value.

Related Front-Panel Keys See *figures 1-8 and 1-9.*

[[:SENSe]:SWEep:TIME:PANorama:SPAN

[[:SENSe]:SWEep:TIME:PANorama:SPAN

This command specifies the time duration of the panorama display (panorama timebase).

Syntax `[[:SENSe]:SWEep:TIME:PANorama:SPAN <numeric value>`

Example `OUTPUT 712;[:SWE:TIME:PAN:SPAN 1E-3"`

- Comments**
- The span is 10 times the setting displayed on the associated softkey because the softkey value is specified "per division" corresponding to 10 display divisions.
 - When panorama is disabled, changing the reference may cause the panorama delay to be updated to keep the panorama display around the main display.
 - When panorama is enabled, changing the reference may cause the main delay to be updated to keep the main display within the panorama display.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - **Units = Seconds [S]**
 - ***RST = 1E-3**

Related Front-Panel Keys See *figure 1-9*.

[[:SENSE]:SWEp:TIME:REPetitive[:STATe]

[[:SENSE]:SWEp:TIME:REPetitive[:STATe]

When REPetitive is ON, repetitive measurement points are shown on the display and AVERage can be enabled or disabled. When REPetitive is OFF, the measurement points shown are real-time.

Syntax `[[:SENSE]:SWEp:TIME:REPetitive[:STATe] ON|1|OFF|0`

Example `OUTPUT 712;":SWE:TIME:REP:STAT ON"`

- Comments**
- To have this command impact analysis functions or data transferred out of the Analyzer, Panorama must be turned off (`:SENS:SWE:TIME:PAN OFF`).
 - ***RST = OFF**

Related Front-Panel Keys See *figure 1-11*.

[[:SENSE]:SWEep:TIME:REPetitive:AVERage

[[:SENSE]:SWEep:TIME:REPetitive:AVERage

This command enables or disables the averaging capability when repetitive is enabled (:SENS:SWE:TIME:REP:STAT ON).

Syntax [:SENSE]:SWEep:TIME:REPetitive:AVERage
ON|1|OFF|0

Example OUTPUT 712;":SWE:TIME:REP:AVER ON"

- Comments**
- This command applies only if :SENS:SWE:TIME:REP:STAT ON.
 - To have this command impact analysis functions or data transferred out of the Analyzer, Panorama must be turned off (:SENS:SWE:TIME:PAN OFF).
 - *RST = OFF

Related Front-Panel Keys See figure 1-11.

[[:SENSe]:SWEep:TIME:SPAN

[[:SENSe]:SWEep:TIME:SPAN

This command specifies the time duration of the main display (main timebase).

Syntax `[[:SENSe]:SWEep:TIME:SPAN <numeric value>`

Example `OUTPUT 712;":SWE:TIME:SPAN 2E-1"`

- Comments**
- The span is 10 times the setting displayed on the associated softkey because the softkey value is specified "per division" corresponding to 10 display divisions.
 - When panorama is disabled, changing the reference may cause the panorama delay to be updated to keep the panorama display around the main display.
 - When panorama is enabled, changing the reference may cause the main delay to be updated to keep the main display within the panorama display.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - ***Units = Seconds [S]**
 - ***RST = 1E-3**

Related Front-Panel Keys See *figures 1-8 and 1-9.*

[[:SENSe]:TINTerval Subtree

[[:SENSe]:TINTerval Subtree

The [[:SENSe]: TINTerval subtree commands control the time interval measurement characteristics.

Subtree Syntax *Figure 2-25* is the syntax diagram for the :TINTerval subtree.

[[:SENSe]:TINTerval Subtree

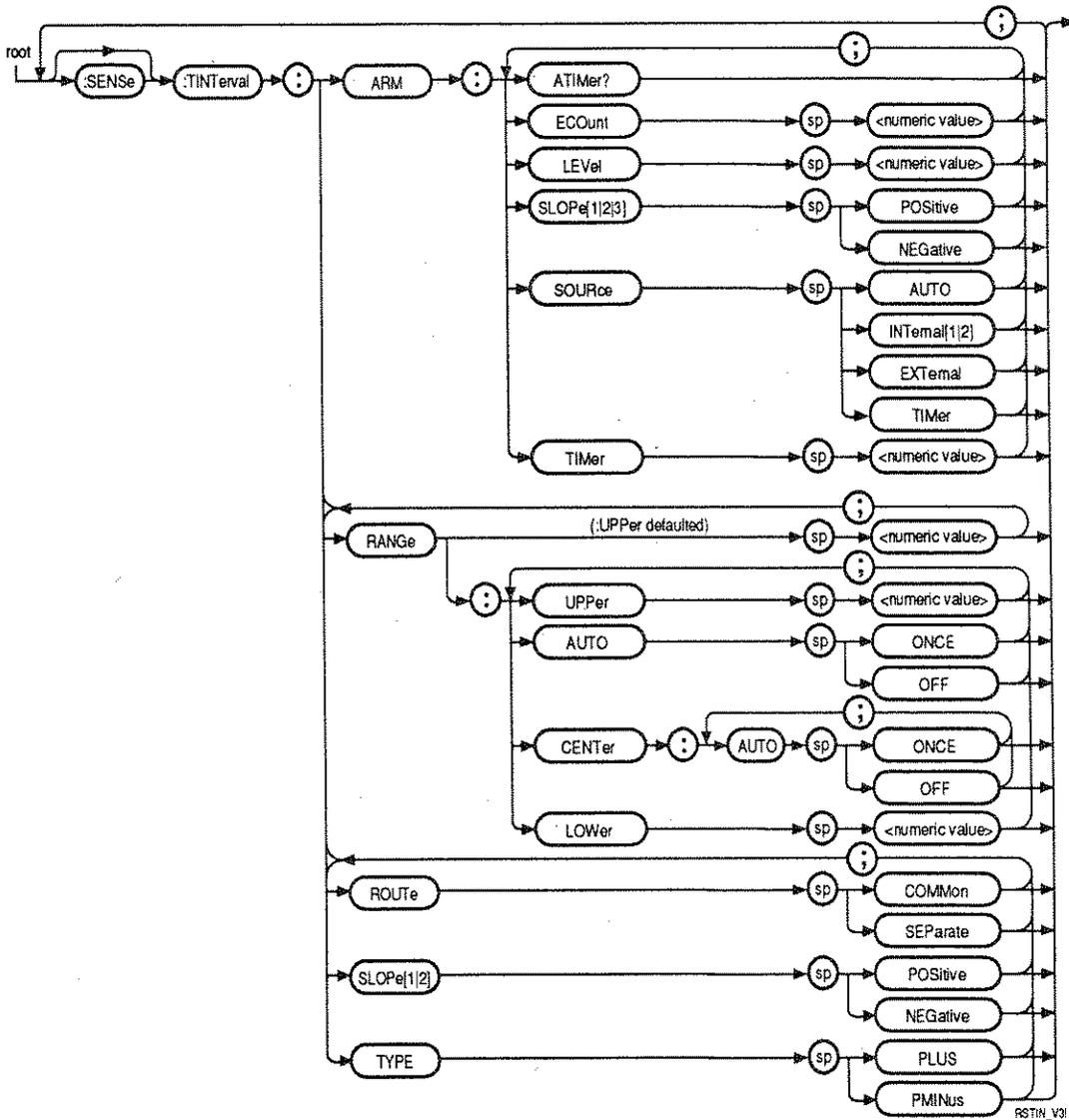


Figure 2-25. :TINTerval Subtree Commands Syntax Chart

`[:SENSe]:TINterval:ARM:ATIMer?`

`[:SENSe]:TINterval:ARM:ATIMer?`

This query's response indicates the sampling time for Time Interval vs Time or Time Interval Histogram from vs Time when `:SENS:TINT:ARM:SOUR` is AUTO.

Syntax `[:SENSe]:TINterval:ARM:ATIMer?`

Example `OUTPUT 712;":TINT:ARM:ATIM?"`

- Comments**
- This value is coupled to the main timebase (`:SENS:SWE:TIME:SPAN`) in that changes to the main timebase modify the auto sampling time.
 - Query only.

Related Front-Panel Keys See *figure 1-15*.

[[:SENSe]:TINTerval:ARM:ECOunt

[[:SENSe]:TINTerval:ARM:ECOunt

This command specifies the event count used for edge sampling of time interval measurements when :SENS:TINT:ARM: SOUR is INT[1|2].

Syntax [:SENSe]:TINTerval:ARM:ECOunt <numeric value>

Example OUTPUT 712;":TINT:ARM:ECO 2"

- Comments**
- ECOunt must be set to a positive value of 1 or greater.
 - Can use MINimum or MAXimum instead of <numeric value>.
 - *RST= 1

Related Front-Panel Keys See *figure 1-15*.

[[:SENSe]:TINTerval:ARM:LEVel

[[:SENSe]:TINTerval:ARM:LEVel

This command specifies the voltage level of the external arm signal used for external edge sampling of time interval measurements when :SENS:TINT:ARM:SOUR is EXT.

Syntax [:SENSe]:TINTerval:ARM:LEVel <numeric value>

Example OUTPUT 712;[:TINT:ARM:LEV 0"

- Comments**
- LEVel is effective only if the source for the event detector is external.
 - This value is coupled to :SENS:FREQ:ARM:LEV, :SENS:HIST:ARM:LEV, and :TRIG:LEV in that they all share the same value. Changing one , changes all others to the same value.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - Allowed values are -1.3 and -0.13 (10:1 probe) for ECL, 0, and 1.5 and 0.15 (10:1 probe) for TTL.
 - **Units = Volts [V]**
 - ***RST = 1.5**

Related Front-Panel Keys See *figure 1-15*.

[[:SENSE]:TINTerval:ARM:SLOPe[1|2|3]]

[[:SENSE]:TINTerval:ARM:SLOPe[1|2|3]]

This command specifies the slope used for edge sampling time interval measurements when :SENS:TINT:ARM:SOUR is INT[1|2] or EXT.

SLOPe1 = Slope of edge sampling on channel A.

SLOPe2 = Slope of edge sampling on channel B.

SLOPe3 = Slope of edge sampling on External Arm signal.

| Parameter Name | Description |
|----------------|--------------------------------------|
| POSitive | Selects the positive (rising) edge. |
| NEGative | Selects the negative (falling) edge. |

Syntax [[:SENSE]:TINTerval:ARM:SLOPe[1|2|3] <parameter>

Example OUTPUT 712;":TINT:ARM:SLOP POS"

Comments • *RST = POS

Related Front-Panel Keys See *figure 1-15*.

[[:SENSe]:TINTerval:ARM:SOURce

[[:SENSe]:TINTerval:ARM:SOURce

This command selects a source for the event detector (that is, for the measurement sampling).

The literal parameter choices are listed in the following table:

| Parameter Name | Description |
|----------------|---|
| INTernal[1 2] | Sets measurement sampling to channel A or B edge sampling. |
| EXTernal | Sets measurement sampling to External Arm edge. |
| TIMer | Sets measurement sampling to occur repetitively in time. Each subsequent arm is referenced to the previous arm. |
| AUTO | Sets measurement sampling to occur automatically. |

Syntax `[[:SENSe]:TINTerval:ARM:SOURce <parameter>`

Example `OUTPUT 712;":TINT:ARM:SOUR EXT"`

- Comments**
- Error -221 is generated and the source is not affected if EXT is requested when EXT is already specified as :TRIG:SOUR or :SENS:HIST:ARM: SOUR.
 - The source is changed from EXT to INT1 when it is necessary to resolve a conflict with :TRIG:SOUR EXT or :SENS:HIST:ARM:SOUR EXT.
 - *RST= AUTO

Related Front-Panel Keys See figure 1-15.

[[:SENSe]:TINTerval:ARM:TIMer

[[:SENSe]:TINTerval:ARM:TIMer

This command specifies the sampling time used for time interval measurements when :SENS:TINT:ARM:SOUR is TIMer.

Syntax [:SENSe]:TINTerval:ARM:TIMer <numeric value>

Example OUTPUT 712;":TINT:ARM:TIM 500 MS"

- Comments**
- TIMer must be a positive value.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - **Units = Seconds [S]**
 - ***RST = 10E-6**

Related Front-Panel Keys See *figure 1-15*.

[[:SENSe]:TINTerval:RANGe[:UPPer]

[[:SENSe]:TINTerval:RANGe[:UPPer]

This command specifies the maximum time interval expected.

Syntax `[[:SENSe]:TINTerval:RANGe[:UPPer] <numeric value>`

Example `OUTPUT 712;":TINT:RANG 3E-3"`

- Comments**
- Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - Units = Seconds [S]
 - *RST = 1.4E-3

Related Front-Panel Keys See *figure 1-7*.

[[:SENSe]:TINTerval:RANGe:AUTO

[[:SENSe]:TINTerval:RANGe:AUTO

When AUTO ONCE is used, it invokes an event that analyzes the input signal(s) and sets appropriate upper (:SENS:TINT:RANG:UPP) and lower (:SENS:TINT:RANG:LOW) values.

Make sure your signal(s) are connected to the input(s) and that the input(s) are configured before using this command.

Syntax [:SENSe]:TINTerval:RANGe:AUTO ONCE | OFF

Example OUTPUT 712;":TINT:RANG:AUTO ONCE"

- Comments**
- AUTO ONCE is a single-shot event, therefore after it is complete, AUTO is set to OFF.
 - Error +2008 is generated when the "auto-configure" (ONCE) fails.
 - Event; no query.
 - *RST = OFF

Related Front-Panel Keys See *figure 1-7*.

[[:SENSE]:TINTerval:RANGe:CENTer:AUTO

[[:SENSE]:TINTerval:RANGe:CENTer:AUTO

When AUTO ONCE is used, it invokes an event that analyzes the input signal(s) and sets appropriate upper (:SENS:TINT:RANG:UPP) and lower(:SENS:TINT:RANG:LOW) values while maintaining the current span between the upper and lower values.

Make sure your signal(s) are connected to the input(s) and that the input(s) are configured before using this command.

Syntax `[[:SENSE]:TINTerval:RANGe:CENTer:AUTO ONCE | OFF`

Example `OUTPUT 712;":TINT:RANG:CENT:AUTO ONCE"`

- Comments**
- AUTO ONCE is a single-shot event, therefore after it is complete, AUTO is set to OFF.
 - Error +2008 is generated when the "auto-configure" (ONCE) fails.
 - Event with query.
 - ***RST = OFF**

Related Front-Panel Keys See *figure 1-7*.

[[:SENSe]:TINTerval:RANGe:LOWer

[[:SENSe]:TINTerval:RANGe:LOWer

This command specifies the minimum time interval expected when measuring time interval.

Syntax `[[:SENSe]:TINTerval:RANGe:LOWer <numeric value>`

Example `OUTPUT 712;[:TINT:RANG:LOW 1E-3"`

- Comments**
- Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - **Units = Seconds [S]**
 - ***RST = 600E-6**

Related Front-Panel Keys See *figure 1-7*.

[[:SENSe]:TINTerval:ROUTe

[[:SENSe]:TINTerval:ROUTe

This command specifies whether the A and B inputs and controls operate independently. If SEParate is specified, the A and B inputs and controls are independent of each other for time interval applications when the signals applied to the A and B inputs are from different sources. If COMMon is specified, the A and B input amplifiers are both connected to the A input. The channel A impedance control (:INP:IMP) is used to condition the input signal to both channel A and B input amplifiers.

Syntax [[:SENSe]:TINTerval:ROUTe COMMon | SEParate

Example OUTPUT 712;":TINT:ROUT COMM"

Comments • *RST = SEP

Related Front-Panel Keys See *figure 1-5*.

[[:SENSe]:TINTerval:SLOPe[1|2]

[[:SENSe]:TINTerval:SLOPe[1|2]

This command specifies the slopes used to make a time interval measurement.

:SENS:TINT:SLOP1 sets the time interval start event's slope (on channel A) and :SENS:TINT:SLOP2 sets the time interval stop event's slope (on channel B).

For example, positive pulse width time intervals could be measured using :SENS:TINT:ROUT COM with :SENS:TINT:SLOP1 POS and :SENS:TINT:SLOP2 NEG.

| Parameter Name | Description |
|----------------|--------------------------------------|
| POSitive | Selects the positive (rising) edge. |
| NEGative | Selects the negative (falling) edge. |

Syntax [:SENSe]:TINTerval:SLOPe[1|2] POSitive | NEGative

Example OUTPUT 712;":TINT:SLOP POS"

Comments • *RST = SLOP1 is POS and SLOP2 is POS

Related Front-Panel Keys See *figure 1-5*.

[[:SENSe]:TINInterval:TYPE

[[:SENSe]:TINInterval:TYPE

This command specifies **+TI (PLUS)** or **± TI (PMin)**.

Syntax `[[:SENSe]:TINInterval:TYPE PLUS|PMinus`

Example `OUTPUT 712;":TINT:TYPE PLUS"`

Comments . *RST = PLUS

Related Front-Panel Keys See *figure 1-5*.

:STATus Subsystem

The :STATus subsystem commands allow you to specify or examine the status of the Operation Status and the Questionable Data/Signal Register groups.

Because the entire status reporting system in the Analyzer is both important and complex, chapter 4, "Status Reporting" contains a detailed explanation (with examples) of how to use this capability efficiently. If you are not familiar with the operation of the status reporting system, read chapter 4 before trying to program these commands.

Figure 4-4 shows all of the status system register groups and queues in the Analyzer. This is a high level drawing that does not show all of the registers that are contained in each group. It is intended as a guide to the bits used in each of these register groups to monitor the Analyzer's status. Note that in addition to the Operation Status and the Questionable Data/Signal Register groups, a summary of the Status Structure Registers (defined by IEEE 488.2-1987) is shown.

:STATus:OPERation Subtree

The :STATus:OPERation subtree commands allow you to examine the status of the Analyzer monitored by the Operation Status Register group. The Operation Status Register group consists of a condition register, two transition registers, an event register, and an enable register. The commands in this subtree allow you to control and monitor these registers.

The Operation Status Register group monitors operations currently being performed. The Analyzer implements bits 0, 4, 5, 8, and 9 (see *table 2-1*). The bit numbers are identical for all of the registers in the Operation Status Register group.

:STATus Subsystem

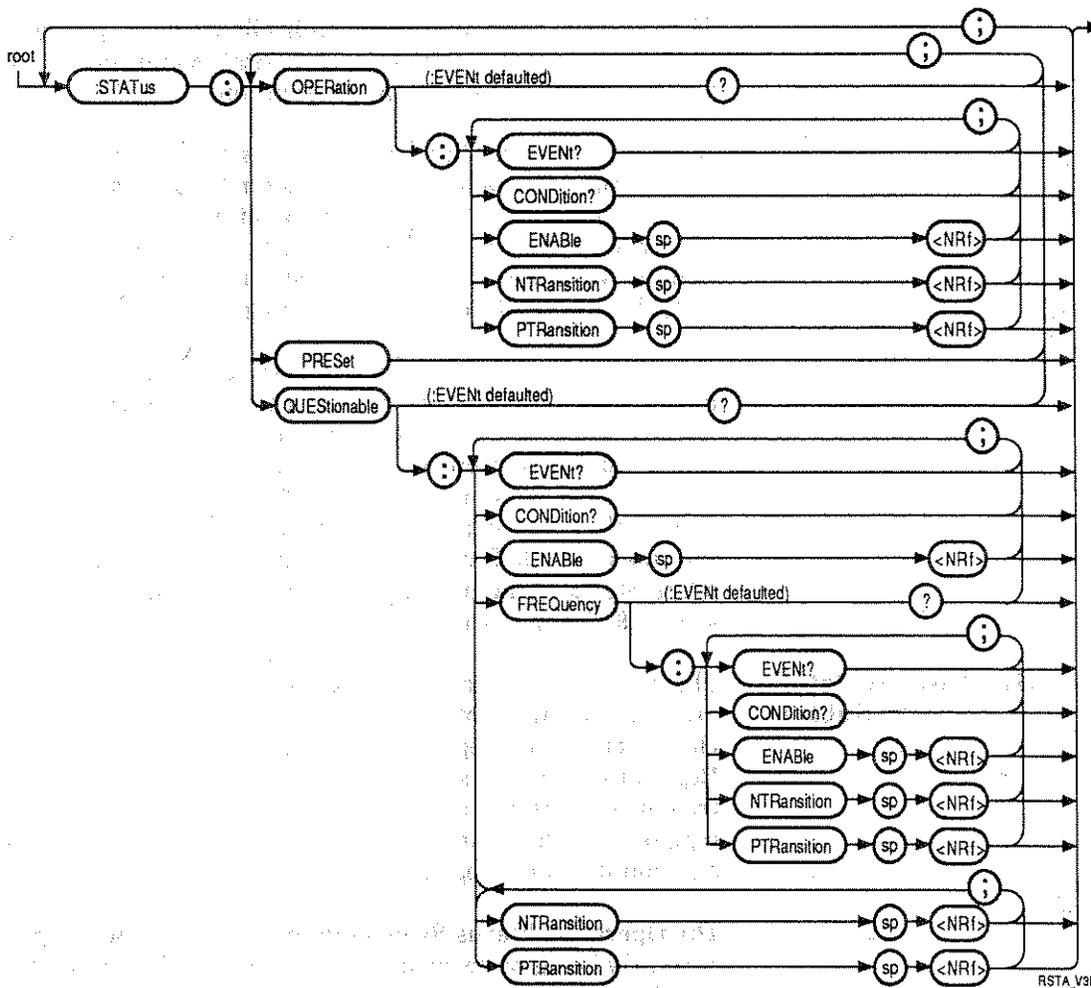


Figure 2-26. :STATus Subsystem Commands Syntax Chart

Table 2-1. Operation Status Register Bits

| BIT | WEIGHT | DESCRIPTION |
|------------|--------|------------------------------|
| 10 thru 15 | NA | Not Used |
| 9 | 512 | Using the internal reference |
| 8 | 256 | Printing |
| 6 and 7 | NA | Not Used |
| 5 | 32 | Waiting for trigger |
| 4 | 16 | Measuring |
| 1 thru 3 | NA | Not Used |
| 0 | 1 | Calibrating |

**:STATus:QUESTionable
Subtree**

The :STATus:QUESTionable subtree commands allow you to examine the status of the Analyzer monitored by the Questionable Status Register group. The Questionable Status Register group consists of a condition register, two transition registers, an event register, and an enable register. The commands in this subtree allow you to control and monitor these registers.

The Questionable Status Register group monitors marginal conditions in the Analyzer. The Analyzer implements bits 2, 4, and 7 (see *table 2-2*). The bit numbers are identical for all of the registers in the Questionable Status Register group.

Bit 4 (Questionable frequency measurements) is used in conjunction with the Questionable Frequency Status Register group to extend the frequency monitoring capabilities. Bit 4 indicates that frequency measurements are of questionable quality due to a timebase reference problem, the selected frequency range including frequencies outside of the specification limits, or possible spurious modulation.

:STATus Subsystem

Bit 2 (Questionable time interval measurements) indicates that time interval measurements are of questionable quality due to a timebase reference problem.

Table 2-2. Questionable Status Register Bits

| BIT | WEIGHT | DESCRIPTION |
|-----------|--------|--|
| 8 thru 15 | NA | Not Used |
| 7 | 512 | Calibration failure detected |
| 5 and 6 | NA | Not Used |
| 4 | 16 | Questionable frequency measurements summary bit — one or more events in the Questionable Frequency Stats Reg have occurred |
| 3 | NA | Not Used |
| 2 | 4 | Questionable time interval measurements |
| 0 and 1 | NA | Not Used |

:STATus:QUEStionable: FREQuency Subtree

The :STATus:QUEStionable:FREQuency subtree commands allow you to examine the status of the Analyzer monitored by the Questionable Frequency Status Register group. The Questionable Frequency Status Register group consists of a condition register, positive and negative transition registers, an event register, and an enable register. The commands in this subtree allow you to control and monitor these registers.

The Questionable Frequency Status Register group monitors marginal frequency conditions in the Analyzer. The Analyzer implements bits 0, 1 and 2 (see *table 2-3*). The bit numbers are identical for all the registers in the Questionable Frequency Status Register group. These bits are used in conjunction with bit 4 (Frequency) of the Questionable Status Register group to extend the frequency monitoring capabilities.

:STATus Subsystem

Table 2-3. Questionable Frequency Status Register Bits

| BIT | WEIGHT | DESCRIPTION |
|-----------|--------|---|
| 3 thru 15 | NA | Not Used |
| 2 | 4 | Possible spurious modulation (see appendix C, "Specifications", in the Operating Reference Manual). |
| 1 | 2 | Selected frequency range out of specification |
| 0 | 1 | Timebase reference problem |

Subsystem Syntax

Figure 2-26 is the syntax diagram for the STATus subsystem.

Comments

- Related commands are *CLS, *SRE <NRf>, *ESE <NRf>, *ESR?, and *STB?.

NOTE

Each of the :OPERation subtree commands are described on separate pages. The :QUEStionable and the :QUEStionable:FREQuency subtree commands operate in the same way except for the registers they affect.

:STATus:OPERation[:EVENT]?

:STATus:OPERation[:EVENT]?

:STATus:QUESTionable[:EVENT]?

:STATus:QUESTionable:FREQuency[:EVENT]?

NOTE

All three commands operate in the same way except for the registers they affect.

This command returns a decimal-weighted number representing the bits set in the corresponding Status Register's event register. This command clears all bits in the event register.

Using this query (or the *CLS Common Command) clears the event register.

Syntax :STATus:OPERation[:EVENT]?

:STATus:QUESTionable[:EVENT]?

:STATus:QUESTionable:FREQuency[:EVENT]?

Example OUTPUT 712;".STAT:OPER?"

Comments

Related Front-Panel Keys None

:STATus:OPERation:CONDition?

:STATus:OPERation:CONDition?

:STATus;;QUEStionable:CONDition?

:STATus:QUEStionable:FREQuency:CONDition?

NOTE

All three commands operate in the same way except for the registers they affect.

This command returns a decimal-weighted number representing the bits set in the corresponding Status Register's condition register. Reading the condition register does not destroy its contents. This is a live register that changes as the conditions in the Analyzer change.

Syntax :STATus:OPERation:CONDition?
:STATus;;QUEStionable:CONDition?
:STATus;;QUEStionable:FREQuency:CONDition?

Example OUTPUT 712;":STAT:OPER:COND?"

Comments

Related Front-Panel Keys None

:STATus:OPERation:ENABLE

:STATus:OPERation:ENABLE

:STATus:QUEStionable:ENABLE

:STATus:QUEStionable:FREQuency:ENABLE

NOTE

All three commands operate in the same way except for the registers they affect.

This command sets bits in the corresponding Status Register's enable register that enable bits in the event register to generate the summary bit (setting Status Byte Register bit 7 true). The event register bits are not reported in the Status Byte Register unless specifically enabled.

If any bits are enabled in the enable register, the corresponding bits set in the Operation Status Register's event register are reported to the Status Byte Register.

The :STAT:OPER:ENAB? query returns a decimal-weighted number representing the bits enabled in the Operation Status Register's enable register.

This is a user-settable register that can only be changed by using :STAT:OPER:ENAB <NRf> to set it to a different value or :STAT:PRES to set it to its preset value.

Syntax :STATus:OPERation:ENABLE <NRf>
 :STATus:QUEStionable:ENABLE <NRf>
 :STATus:QUEStionable:FREQuency:ENABLE <NRf>

:STATus:QUEStionable:FREQuency:ENABle

Example OUTPUT 712;":STAT:OPER:ENABle 1"

- Comments**
- Can also use a <non-decimal numeric> (numeric specified in binary, octal, or hexadecimal) in place of the <NRf> parameter. Query always returns a <NR1> parameter.
 - Related command is :STAT:PRES.

Related Front-Panel Keys None

:STATus:OPERation:NTRansition

:STATus:OPERation:NTRansition

:STATus:QUEStionable:NTRansition

:STATus:QUEStionable:FREQuency:NTRansition

NOTE

All three commands operate in the same way except for the registers they affect.

This command sets the specified bits in the corresponding negative transition filter. Setting a bit in the negative transition filter enables a negative transition (1 to 0) in the corresponding bit of the operation condition register to set a 1 in the associated bit of the operation event register.

This is a user-settable register that can only be changed by using :STAT:OPER:NTR <NRf> to set it to a different value or :STAT:PRES to set it to its preset value.

Syntax :STATus:OPERation:NTRansition <NRf>
:STATus:QUEStionable:NTRansition <NRf>
:STATus:QUEStionable:FREQuency:NTRansition <NRf>

Example OUTPUT 712;":STAT:OPER:NTR 32"

Comments

- Can also use a <non-decimal numeric> (numeric specified in binary, octal, or hexadecimal) in place of the <NRf> parameter. Query always returns a <NR1> parameter.
- Related command is :STAT:PRES.

Related Front-Panel Keys None

:STATus:OPERation:PTRansition

:STATus:OPERation:PTRansition

:STATus:QUEStionable:PTRansition

:STATus:QUEStionable:FREQuency:PTRansition

NOTE

All three commands operate in the same way except for the registers they affect.

This command sets the specified bits in the operation positive transition filter. Setting a bit in the positive transition filter enables a positive transition (0 to 1) in the corresponding bit of the operation condition register to set a 1 in the associated bit of the operation event register.

This is a user-settable register that can only be changed by using the :STAT:OPER:PTR <NRf> to set it to a different value or :STAT:PRES to set it to its preset value.

Syntax :STATus:OPERation:PTRansition <NRf>
:STATus:QUEStionable:PTRansition <NRf>
:STATus:QUEStionable:FREQuency:PTRansition <NRf>

Example OUTPUT 712;":STAT:OPER:PTR 32"

- Comments**
- Can also use a <non-decimal numeric> (numeric specified in binary, octal, or hexadecimal) in place of the <NRf> parameter. Query always returns a <NR1> parameter.
 - Related command is :STAT:PRES.

Related Front-Panel Keys None

:STATus:PRESet

:STATus:PRESet

This event command presets the Operation and Questionable data structures. Additional information on data structures and status reporting is contained in chapter 4.

The :STATus:PRESet command presets the operation and questionable data structures as shown in the following table.

| Register Group | Enable/Transition Registers | :STATus:PRESet value = |
|-------------------------|-----------------------------|------------------------|
| :OPERation | :ENABle | 0's |
| :OPERation | :NTRansition | 0's |
| :OPERation | :PTRansition | 1's |
| :QUESTionable | :ENABle | 0's |
| :QUESTionable | :NTRansition | 0's |
| :QUESTionable | :PTRansition | 1's |
| :QUESTionable:FREQuency | :ENABle | 1's |
| :QUESTionable:FREQuency | :NTRansition | 0's |
| :QUESTionable:FREQuency | :PTRansition | 1's |

Syntax :STATus:PRESet

:STATus:PRESet

Example To initialize status reporting:

```
OUTPUT 712;"*CLS"  
!Clears all Event Registers and the Error Queue  
OUTPUT 712;"*SRE 0"  
!Clears Service Request Enable register  
OUTPUT 712;"*ESE 0"  
!Clears Event Status Enable register  
OUTPUT 712;" :STATus:PRESet"  
!Presents Operation and Questionable enable registers and  
!transition filters
```

- Comments**
- Refer to your controller manual and programming language reference manual for information on initializing your interface.
 - Event; no query.

Related Front-Panel Keys None

:STATus:PRESet

:SYSTEM Subsystem

The :SYSTEM subsystem commands control functions that are not directly related to instrument performance.

Subsystem Syntax *Figure 2-27 is the syntax diagram for the SYSTEM subsystem.*

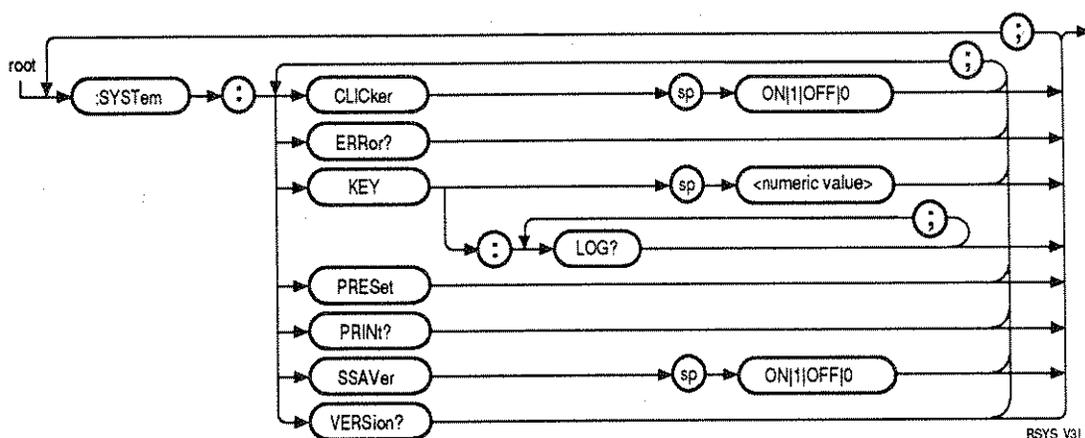


Figure 2-27. :SYSTEM Subsystem Commands Syntax Chart

:SYSTem:CLICker

:SYSTem:CLICker

This command enables or disables the audible clicking sound when the front panel keys are pressed.

Syntax :SYSTem:CLICker ON|1|OFF|0

Example OUTPUT 712;":SYST:CLIC OFF"

Comments • *RST = ON

Related Front-Panel Keys See *figure 1-16*.

:SYSTem:ERRor?

:SYSTem:ERRor?

This query is a request for the first error in the Analyzer's error queue. The error number part of the response is an integer in the range of -32768 to 32767. Negative error numbers are reserved by the SCPI standard. Positive error numbers are instrument dependent. An error number of 0 indicates no error has occurred and one of -350 indicates a full queue. See appendix F, Error Messages, for additional information on specific error messages and the manipulation of the error queue.

Syntax :SYSTem:ERRor?

Example OUTPUT 712;":SYST:ERR?"

Comments

- *CLS clears the error queue.
- Query only.

**Related
Front-Panel Keys** None

:SYSTEM:KEY

:SYSTEM:KEY

This event command acts as a command when a numeric value is specified but also acts as a subtree when the :LOG? query is used.

Keys are referred to numerically. If the :SYSTEM:KEY is sent with a numeric value, it simulates the pressing of that key. The :KEY command (as well as pressing a front-panel key) places an entry into the key queue.

The key query returns the key code for the last key pressed. The Analyzer has a 500 entry key queue and returns a -1 to indicate an empty queue. The queue is a "last-in, first-out" queue.

The following table lists the keys and their corresponding number:

| KEY | NO. | KEY | NO. | KEY | NO. |
|--------------------|-----|-----------|-----|------------------|-----|
| Softkey 7 (bottom) | 17 | 5 | 67 | Softkey 1 (top) | 113 |
| Shift | 18 | 6 | 68 | Function & Input | 114 |
| Clear | 19 | ms | 69 | Vertical | 115 |
| Softkey 6 | 33 | Softkey 3 | 81 | Timebase | 116 |
| 0 | 34 | 7 | 82 | Trigger | 117 |
| . | 35 | 8 | 83 | Display | 118 |
| +/- | 36 | 9 | 84 | Recall/Save | 119 |
| ns | 37 | s | 85 | Run | 130 |
| Softkey 5 | 49 | Fine | 87 | Stop/Single | 131 |
| 1 | 50 | Softkey 2 | 97 | Clear display | 132 |
| 2 | 51 | Markers | 99 | Local | 133 |
| 3 | 52 | Histogram | 100 | Print | 134 |
| us | 53 | Sampling | 101 | Preset | 135 |
| Softkey 4 | 65 | Utility | 102 | Autoscale | 136 |
| 4 | 66 | Status | 103 | | |

Syntax :SYSTEM:KEY <numeric value>

Example OUTPUT 712;":SYST:KEY 114"

:SYSTem:KEY

- Comments**
- It is not recommended that you use this command as a substitute for the other programming commands because the simulated key presses do not receive the synchronization protection the programming commands do.
 - Can use MINimum or MAXimum instead of <numeric value>.
 - ***RST = Key queue cleared**

Related Front-Panel Keys None

:SYSTem:KEY:LOG?

:SYSTem:KEY:LOG?

This command returns all of the entries in the key queue. The entries are read out in a "last in, first out" fashion. A "-1" is returned to indicate an empty queue.

Syntax :SYSTem:KEY:LOG?

Example OUTPUT 712;":SYST:KEY:LOG?"

Comments

- Query only.
- ***RST = Key queue cleared.**

Related Front-Panel Keys None

:SYSTem:PRESet

:SYSTem:PRESet

This event command presets the Analyzer to the same state as the front-panel preset function.

Syntax :SYSTem:PRESet

Example OUTPUT 712;":SYST:PRES"

- Comments**
- One difference in the instrument configuration between :SYSTem:PRESet and *RST is that the Analyzer is idle (that is, :INIT:CONT is OFF) after using *RST. See *table 1-1*, "Subsystem Commands Summary" in chapter for a complete list of the reset conditions.
 - The *RST common command is recommended for remote initialization of the Analyzer's reset state.
 - Event; no query.

Related Front-Panel Keys See *figure 1-3*.

:SYSTem:PRINt?

:SYSTem:PRINT?

This query causes the Analyzer to put PCL-formatted display image data in the Output Queue.

Syntax :SYSTem:PRINT?

Example OUTPUT 712;":SYST:PRIN?"

- Comments**
- See appendix D, "Programming Examples" for an example of how to use the :SYST:PRIN? command.
 - Query only.

Related Front-Panel Keys See *figure 1-3*.

:SYSTem:SSAVer

:SYSTem:SSAVer

This event command enables or disables the screen saver function.

Syntax :SYSTem:SSAVer ON|1|OFF|0

Example OUTPUT 712;":SYST:SSAV"

- Comments**
- Event; no query.
 - *RST = OFF

Related Front-Panel Keys See *figure 1-3*.

:SYSTem:VERSion?

:SYSTEM:VERSion?

This query returns an <NR2> formatted numeric value corresponding to the SCPI version number for which the Analyzer complies. The response has the form YYYY.V. The Ys represent the year (for example, 1990) and the V represents an approved version for that year (for example, 0). The Analyzer complies with the SCPI standard 1990.0 and, therefore, returns that number for this query.

Syntax :SYSTem:VERSion?

Example OUTPUT 712;":SYST:VERS?"

Comments • Query only.

**Related
Front-Panel Keys** None

:TRIGger Subsystem

:TRIGger Subsystem

The :TRIGger subsystem commands synchronize the Analyzer's actions with the specified events for vs Time measurements.

Fast Histogram measurements are synchronized by using the :SENS:HIST:ARM and :SENS:HIST:TRIG subtree commands.

Syntax *Figure 2-28* is the syntax diagram for the TRIGger subsystem.

:TRIGger Subsystem

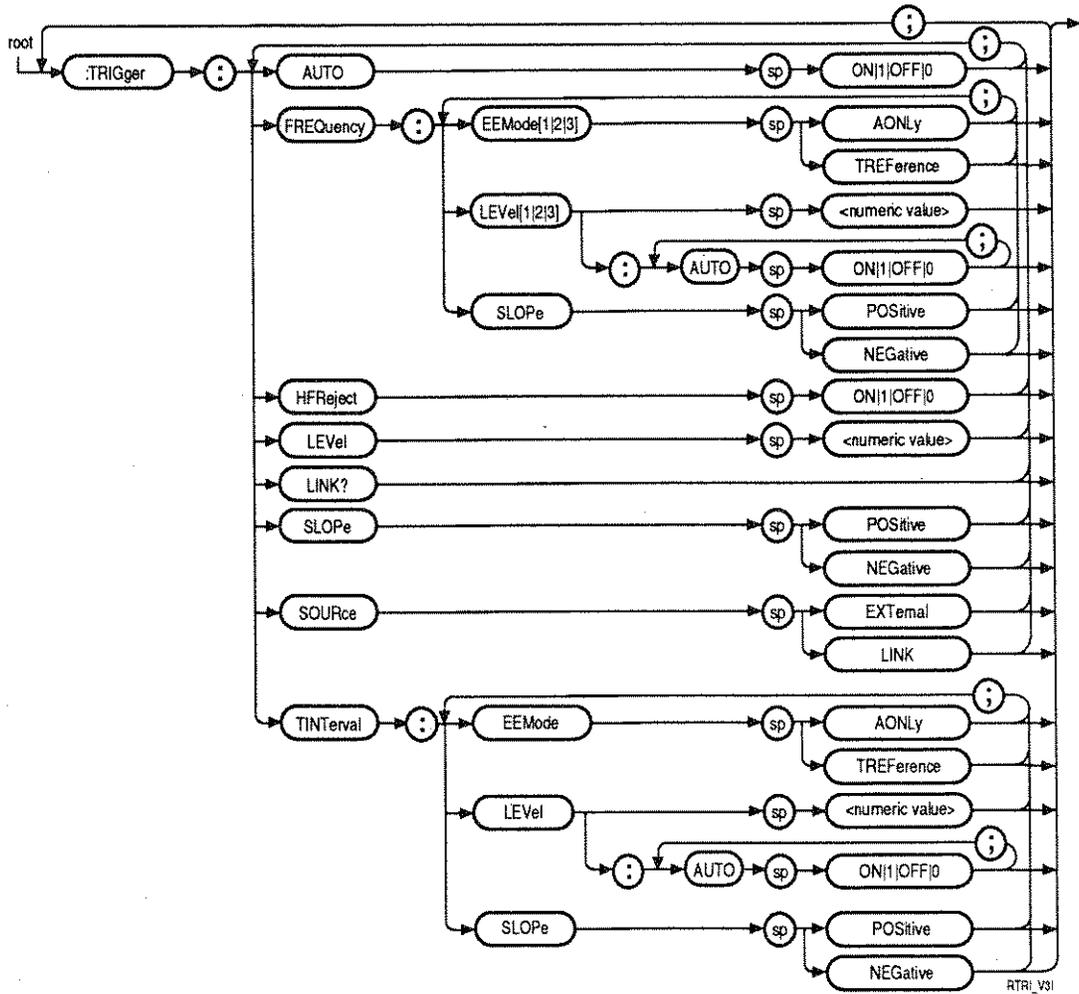


Figure 2-28. :TRIGger Subsystem Commands Syntax Chart

:TRIGger:AUTO

:TRIGger:AUTO

If ON is specified, the Analyzer auto-triggers if necessary. If OFF is specified, the Analyzer waits indefinitely for the trigger specified by :TRIG:SOUR.

Syntax :TRIG:AUTO ON|1|OFF|0

Example OUTPUT 712;":TRIG:AUTO ON"

Comments • *RST = ON

Related Front-Panel Keys See *figure 1-10*.

:TRIGger:FREQUency:EEMode[1|2|3]

A signal at the External Arm input can be set to simply arm the measurement (AONL) or to provide a time record for the measurement (TREF). This time record is referenced to the instant at which the External Arm input signal occurred.

Syntax :TRIGger:FREQUency:EEMode[1|2|3]
AONLy|TREFerence

Example OUTPUT 712;":TRIG:FREQ:EEM3 TREF"

Comments

- The example command sets the external edge trigger mode to Time Reference for a frequency measurement on channel C.
- Use the command, :TRIG:SOUR EXT, to enable the external edge trigger mode.
- Use "AONL" for Arm Only or "TREF" for Time Reference.
- RST = AONLy

Related Front-Panel Keys See *figure 1-10*.

:TRIGger:FREQuency:LEVel[1|2|3]

:TRIGger:FREQuency:LEVel[1|2|3]

This command specifies the frequency to “value trigger” on when sensing Frequency vs Time with :TRIG:SOUR LINK. 1 = frequency on A, 2 = frequency on B, and 3 = frequency on C.

Acts as a command when a numeric value is specified but also acts as a subtree when the :AUTO command is used. Can use MINimum or MAXimum instead of <numeric value> [suffix].

Syntax :TRIG:FREQuency:LEVel <numeric value>

Example OUTPUT 712;":TRIG:FREQ:LEV 9.5 MHZ"

- Comments**
- Only applies when sensing Frequency vs Time (:SENS:FUNC "FREQ[1|2|3]") and value triggering (:TRIG:SOUR LINK).
 - Specifying a <numeric value> turns :TRIG:FREQ:LEV:AUTO to OFF and disables the “auto-centering” of the “value trigger” level.
 - **Units = Hertz [HZ]**
 - ***RST = LEV1 is 100E+6, LEV2 is 50E+6, and LEV3 is 1E+9.**

Related Front-Panel Keys See *figure 1-10*.

:TRIGger:FREQuency:LEVel[1|2|3]:AUTO

:TRIGger:FREQuency:LEVel[1|2|3]:AUTO

When ON is specified, this command couples :TRIG:FREQ:LEV (value trigger level) to the center of the frequency range. Whenever the frequency range changes (:SENS:FREQ:RANG[1|2|3] UPP and :SENS:FREQ:RANG[1|2|3] LOW), the frequency value trigger level is set to the center of the new frequency range.

Syntax :TRIG:FREQuency:LEVel[1|2|3]:AUTO ON|1|OFF|0

Example OUTPUT 712;" :TRIG:FREQ:LEV:AUTO ON"

- Comments**
- Only applies when sensing Frequency vs Time (:SENS:FUNC "FREQ[1|2|3]") and value triggering (:TRIG:SOUR LINK).
 - Related command are :SENS:FREQ:RANG[1|2|3] LOW and :SENS:FREQ:RANG[1|2|3] UPP.
 - **Units = Hertz [HZ]**
 - ***RST = ON**

Related Front-Panel Keys See figure 1-10.

:TRIGger:FREQuency:SLOPe

:TRIGger:FREQuency:SLOPe

This command specifies which edge to “value trigger” on when sensing Frequency vs Time with :TRIG:SOUR LINK. POSitive triggers on the rising edge of the frequency modulation and NEGative triggers on the falling edge.

Syntax :TRIG:FREQ:SLOPe POSitive | NEGative

Example OUTPUT 712;":TRIG:FREQ:SLOP POS"

- Comments**
- Only applies when sensing Frequency vs Time (:SENS:FUNC "FREQ[1|2|3]") and value triggering (:TRIG:SOUR LINK).
 - *RST = POS

Related Front-Panel Keys See *figure 1-10*.

:TRIGger:HFReject

:TRIGger:HFReject

When ON, the Analyzer rejects spurious high-frequency transitions on the "value trigger" for vs Time measurements (frequency or time interval) when :TRIG:SOUR LINK is used.

Syntax :TRIG:HFReject ON|1|OFF|0

Example OUTPUT 712;":TRIG:HFR ON"

- Comments**
- Only applies when :TRIG:SOUR LINK is used.
 - ***RST = OFF**

Related Front-Panel Keys See *figure 1-10*.

:TRIGger:LEVel

:TRIGger:LEVel

Specifies the voltage value of the external signal used to trigger a vs Time measurement when :TRIG:SOUR is EXT.

Syntax :TRIGger:LEVel <numeric value>

Example OUTPUT 712;":TRIG:LEV 0"

- Comments**
- Only used when :TRIG:SOUR is set to EXT.
 - This value is coupled to :SENS:HIST:ARM:LEV, :SENS:FREQ:ARM:LEV, and :SENS:TINT:ARM:LEV in that they share the same value. Changing one, changes all the others to the same value.
 - Allowed values are -1.3 and -0.13 (10:1 probe) for ECL, 0, and 1.5 and 0.15 (10:1 probe) for TTL.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - **Units = Volts [V]**
 - ***RST = 1.5**

**Related
Front-Panel Keys**

See *figure 1-10*.

:TRIGger:LINK?

:TRIGger:LINK?

This query always returns the string "SENS:FUNC" and is used to indicate that when :TRIG:SOUR is LINK, the source of the trigger event is coupled to the current function being sensed. That is, when :TRIG:SOUR is LINK, the Analyzer "value triggers" on the frequency or time interval (when sensing vs Time).

Syntax :TRIGger:LINK?

Example OUTPUT 712;":TRIG:LINK?"

Comments • Query only

Related Front-Panel Keys None.

:TRIGger:SLOPe

:TRIGger:SLOPe

This command specifies the slope of the edge of the external signal used to trigger a vs Time measurement when :TRIG:SOUR is EXT. POSitive triggers on the rising edge of the external signal and NEGative triggers on the falling edge.

Syntax :TRIGger:SLOPe POSitive | NEGative

Example OUTPUT 712;":TRIG:SLOP POS"

- Comments**
- Only used when :TRIG:SOUR is set to EXT.
 - This value is coupled to :SENS:HIST:ARM:SLOP in that they share the same value. Changing one, changes the other to the same value.
 - *RST = POS

Related Front-Panel Keys See *figure 1-10*.

:TRIGger:SOURce

This command selects the source for the vs Time trigger. (that is, for the sweep trigger event). The available source choices are described in the following table:

| PARAMETER NAME | DESCRIPTION |
|----------------|--|
| EXTernal | Sets sweep trigger to external input edge. |
| LINK | Sets sweep to "value trigger" on a time interval measurement or frequency measurement. |

Syntax :TRIGger:SOURce <parameter>

Example OUTPUT 712;":TRIG:SOUR EXT"

- Comments**
- The external arm signal cannot be used for sweep triggering (:TRIG:SOUR EXT) and measurement sampling (:SENS:FREQ:ARM:SOUR EXT and :SENS:TINT:ARM:SOUR EXT) simultaneously. This conflict is resolved by allowing the :TRIG:SOUR to be set to EXT and changing the measurement sampling to a source other than EXT.
 - *RST =LINK

Related Front-Panel Keys See figure 1-10.

:TRIGger:TINTerval:EEMode

:TRIGger:TINTerval:EEMode

A signal at the External Arm input can be set to simply arm the measurement (AONL) or to provide a time record for the measurement (TREF). This time record is referenced to the instant at which the External Arm input signal occurred.

Syntax :TRIGger:TINTerval:EEMode AONLy | TREFerence

Example OUTPUT 712;"TRIG:TINT:EEM TREF"

- Comments**
- The example command sets the external edge trigger mode to Time Reference for a time interval measurement.
 - Use the command, :TRIG:SOUR EXT, to enable the external edge trigger mode.
 - Use "AONL" for Arm Only or "TREF" for Time Reference.
 - *RST = AONLy

Related Front-Panel Keys See *figure 1-10*.

:TRIGger:TINTerval:LEVel

:TRIGger:TINTerval:LEVel

This command specifies which time interval to “value trigger” on when sensing Time Interval vs Time with :TRIG:SOUR LINK.

Acts as a command when a numeric value is specified, but also acts as a subtree when the AUTO command is used.

Syntax :TRIG:TINT:LEVel <numeric value>

Example OUTPUT 712;":TRIG:TINT:LEVel 5E-1"

- Comments**
- Only applies when sensing Time Interval vs Time (:SENS:FUNC “TINT”) and value triggering (:TRIG:SOUR LINK).
 - Specifying a <numeric value> turns :TRIG:TINT:LEV:AUTO to OFF and disables the “auto centering” of the “value trigger” level.
 - Can use MINimum or MAXimum instead of <numeric value> [suffix].
 - **Default units = Seconds [S]**
 - ***RST = 1E-3**

**Related
Front-Panel Keys**

See *figure 1-10*.

:TRIGger:TINterval:LEVel:AUTO

:TRIGger:TINterval:LEVel:AUTO

When ON is specified, this command couples :TRIG:TINT:LEV (the value trigger level) to the center of the time interval range. Whenever the time interval range changes (SENS:TINT:RANG:UPP and SENS:TINT:RANG:LOW), the time interval value trigger level is set to the center of the new time interval.

Syntax :TRIG:TINterval:LEVel:AUTO ON|1|OFF|0

Example OUTPUT 712;":TRIG:TINT:LEV:AUTO ON"

- Comments**
- Only applies when sensing Time Interval vs Time (:SENS:FUNC "TINT") and value triggering (:TRIG:SOUR LINK).
 - *RST = ON

Related Front-Panel Keys See *figure 1-10*.

:TRIGger:TINterval:SLOPe

This command specifies the slope to "value trigger" on when sensing Time Interval vs Time. POSitive triggers on the rising edge of the time interval modulation and NEGative triggers on the falling edge.

Syntax :TRIG:TINT:SLOPe POSitive|NEGative

Example OUTPUT 712;":TRIG:TINT:SLOP POS"

- Comments**
- Only applies when sensing Time Interval vs Time (:SENS:FUNC "TINT") and value triggering (:TRIG:SOUR LINK).
 - *RST = POS

Related Front-Panel Keys See *figure 1-10*.

3

COMMON COMMANDS REFERENCE DICTIONARY

In This Chapter

This chapter describes the Common Commands defined by IEEE Standard 488.2-1987 that are used in the Analyzer. These Common Commands control some of the basic Analyzer functions, such as reset and how the Status Data Structures are read and cleared.

Common Commands can be issued at any time, either alone in a program message or with other commands in a program message.

Examples:

| PROGRAM STATEMENT | DESCRIPTION |
|-------------------------------------|--|
| OUTPUT 712;"CLS" | alone in a program message |
| OUTPUT 712;"INP:COUP AC;"CLS";INIT" | with other commands in a program message |

If a subsystem is selected and a Common Command is received by the Analyzer, the Analyzer remains in the selected subsystem.

Many of the common commands are used for status reporting. See chapter 4 "Status Reporting" for a complete discussion of how to read the status registers and how to use status information available from the Analyzer.

Table 3-1 lists the Common Commands and *figure 3-1* shows the Common Commands syntax diagram.

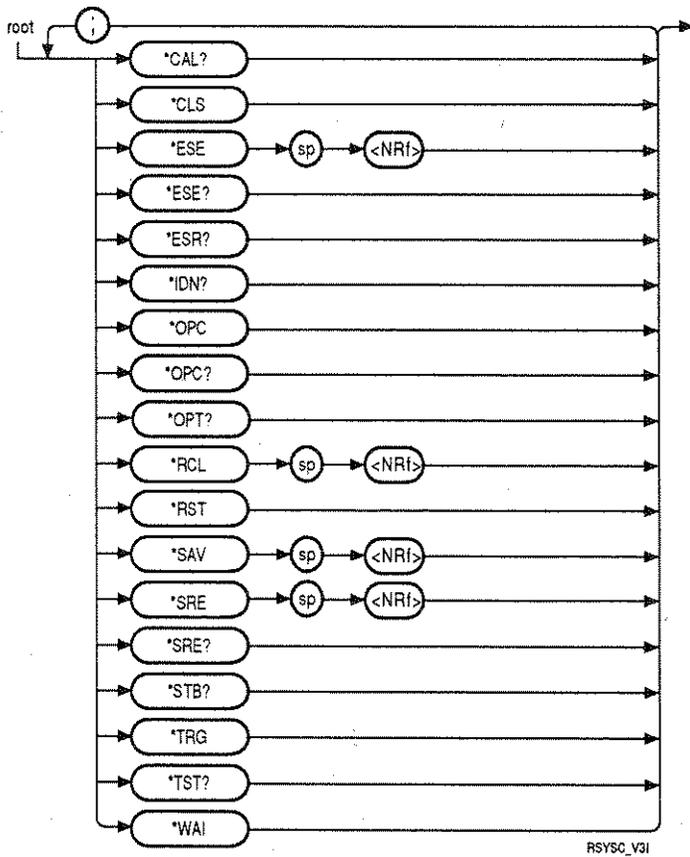


Figure 3-1. Common Commands Syntax Chart

Table 3-1. Common Commands Summary

| COMMAND | DESCRIPTION |
|------------|--|
| *CAL? | Performs an internal interpolator self-calibration. |
| *CLS | Clears Status data structures (Event Registers and Error Queue). |
| *ESE <NRf> | Sets the Standard Event Status Enable Register. |
| *ESE? | Queries the Standard Event Status Enable Register. |
| *ESR? | Queries the Standard Event Status Register. |
| *IDN? | Queries the Analyzer identification. |
| *OPC | Causes Analyzer to set the operation complete bit in the Standard Event Status Register when all pending operations (see note) are finished. |
| *OPC? | Places an ASCII "1" in the Output Queue when all pending operations (see note) are completed. |
| *OPT? | Identifies the options installed in the Analyzer. |
| *RCL <NRf> | Restores state of the Analyzer from a copy stored in local non-volatile memory (0 through 9 are valid memory registers). |
| *RST | Resets the Analyzer to a known state. |
| *SAV <NRf> | Stores the current state of the Analyzer in local non-volatile memory (1 through 9 are valid memory registers). |
| *SRE <NRf> | Sets the Service Request Enable Register. |
| *SRE? | Queries the Service Request Enable Register. |
| *STB? | Queries the Status Byte and Master Summary Status bit. |
| *TRG | This trigger command is the device-specific analog of the IEEE 488.1-defined GET. It initiates a single measurement. |
| *TST? | Executes an internal self-test and reports the results. |
| *WAI | Makes Analyzer wait until all pending operations (see note) are complete before executing commands following *WAI command. |

Note: Pending operations include measurements in progress.

****CAL? (Calibration)***

***CAL? (Calibration)**

The *CAL? command directs the Analyzer to perform an internal interpolator self-calibration. A response of 0 (zero) indicates that the calibration executed successfully. A response of 1 indicates the calibration failed.

Syntax *CAL?

Example To perform an internal interpolator self-calibration:

OUTPUT 712; "*CAL?"

- Comments**
- Error +2004 is generated if the interpolator calibration fails.
 - The Operation Status Register has a "Calibrating" bit and the Questionable Status Register has a "Calibration Error" bit.

Related Commands

****CLS (Clear Status)***

***CLS (Clear Status)**

The *CLS command clears all Event Registers summarized in the status byte (Standard Event Status Register, Operation Event Status Register, Questionable Data Event Status Register, and Questionable Frequency Event Status Register), and clears the error queue.

It also places the Analyzer in "Operation Complete Idle State" and "Operation Complete Query Idle State" (IEEE 488.2). This results in disabling any prior *OPC command.

The *CLS command does not affect any of the Enable Registers, Transition Filter Registers, or Condition Registers.

If the *CLS command immediately follows a program message terminator, the Output Queue and the MAV bit are cleared because any new program message after a program message terminator clears the output queue.

Syntax *CLS

Example To clear all of the Event Registers and the Error Queue:

OUTPUT 712; "*CLS"

- Comments**
- The *CLS command does not clear data memories or any settings other than the status data structures.
 - See Status Reporting in chapter 4 for additional information about status data structures.

Related Commands *ESR?
*STB?
STATus subsystem commands

**ESE (Event Status Enable)*

***ESE (Event Status Enable)**

The *ESE command sets the Standard Event Status Enable Register bits. Setting a bit to 1 in the Standard Event Status Enable Register enables the corresponding bit in the Standard Event Status Register to generate a summary bit to the Status Byte Register.

When a bit is set to 1 in the Standard Event Status Enable Register and its corresponding bit in the Standard Event Status Register is set to 1, the Event Status bit (ESB) in the Status Byte Register (bit 5) is set.

In response to *ESE?, the Analyzer returns a number from 0 to 255. This number represents the total weight of all bits in the Standard Event Status Enable Register that are set to 1.

Syntax *ESE <NRf>

*ESE?

| ITEM | DESCRIPTION | RANGE/RESTRICTIONS |
|------|--|--------------------|
| NRf | Weight of bits to be enabled in the Standard Event Status Register | 0 to 255 |

Example You enable an event by specifying its decimal weight (as shown in the table in Comments). To enable more than one event, specify the sum of the decimal weights.

To enable the command, execution, device-specific, and query errors to set bit 5 (ESB, Event Status Bit) of the Status Byte Register:

OUTPUT 712; "*ESE 60"

Comments

- The weight of each bit in the Standard Event Status Enable Register is shown in the following table.

***ESE (Event Status Enable)**

Standard Event Status Enable Register

| BIT | WEIGHT | DESCRIPTION |
|-----|--------|---|
| 7 | 128 | Power on. Power has been applied to the Analyzer. |
| 6 | NA | Not used |
| 5 | 32 | Command Error (-100 thru -199). Indicates a syntax error has been detected by the Analyzer's parser. |
| 4 | 16 | Execution Error (-200 thru -299). Indicates the current command cannot be executed because of an out of range parameter or conflicting settings. |
| 3 | 8 | Device Specific Error (-300 thru -399 or +2000 thru +2011). Indicates the Analyzer has detected a device-specific error (one that is not a command error, query error, or execution error). |
| 2 | 4 | Query Error (-400 thru -499). Indicates either an attempt was made to read the Output Queue when it was empty or output data has been lost. |
| 1 | NA | Not used. |
| 0 | 1 | Operation Complete. Indicates the Analyzer has completed any pending operations and is ready to accept new commands. This bit is set only in response to the *OPC command. |

- When power to the Analyzer is turned on, the Standard Event Status Enable Register is set to 0.
- See Status Reporting in chapter 4 for additional information about the Standard Event Status Register.

Related Commands

*ESR?
*OPC
*SRE
*STB?

****ESR? (Event Status Register Query)***

***ESR? (Event Status Register Query)**

The *ESR? query returns the contents of the Standard Event Status Register. This register is an 8-bit register. It is constantly updated as events occur. Once a bit is set to 1, it remains set until the register is read with the *ESR? query or cleared with the *CLS command.

Syntax *ESR?

Example To read the current integer value of the Event Status Register:

```
10  OUTPUT 712; “*ESR?”
20  ENTER 712; A
30  PRINT “EVENT STATUS REGISTER =”; A
40  END
```

Comments

- The following table shows each bit in the Standard Event Status Register and the bit weight. When you read the Standard Event Status Register, the value returned is the total bit weights of all bits that are set to 1 at the time you read the byte.

***ESR? (Event Status Register Query)**

Standard Event Status Register

| BIT | WEIGHT | DESCRIPTION |
|-----|--------|---|
| 7 | 128 | Power on. Power has been applied to the Analyzer. |
| 6 | NA | Not used. |
| 5 | 32 | Command Error (-100 thru -199). Indicates a syntax error has been detected by the Analyzer's parser. |
| 4 | 16 | Execution Error (-200 thru -299). Indicates the current command cannot be executed because of an out of range parameter or conflicting settings. |
| 3 | 8 | Device Specific Error (-300 thru -399 or +2000 thru +2011). Indicates the Analyzer has detected a device-specific error (one that is not a command error, query error, or execution error). |
| 2 | 4 | Query Error (-400 thru -499). Indicates either an attempt was made to read the Output Queue when it was empty or output data has been lost. |
| 1 | NA | Not used. |
| 0 | 1 | Operation Complete. Indicates the Analyzer has completed any pending operations and is ready to accept new commands. This bit is set only in response to the *OPC command. |

Related Commands

- *CLS
- *ESE
- *OPC
- *SRE
- *STB?

**IDN? (Identification Query)*

***IDN? (Identification Query)**

This query causes the Analyzer to identify itself over the bus. In response to *IDN?, the Analyzer sends the following ascii response:

HEWLETT-PACKARD,53310A,0,xxxx

- **HEWLETT-PACKARD** is the manufacturer.
- **53310A** is the model number.
- **0** is returned instead of the Analyzer serial number.
- **xxxx** is the firmware version.

Syntax *IDN?

Example To determine the Analyzer's version of firmware:

```
10  OUTPUT 712; "*IDN?"
20  DIM A$(40)
30  ENTER 712; A$
40  PRINT "ANALYZER IS"; A$
50  END
```

Comments

Related Commands None

****OPC (Operation Complete)***

***OPC (Operation Complete)**

The *OPC command sets the operation complete bit (bit 0) in the Standard Event Status Register when the Analyzer has completed all pending operations (which includes measurements in progress).

Syntax *OPC

Example To initiate a measurement and print a message when the Analyzer has completed it.

```
10  OUTPUT 712;"*CLS;*SRE 32;*ESE 1"  
20  ON INTR 7 GOTO Meas_done  
30  ENABLE INTR 7;2  
40  OUTPUT 712;"*OPC;;INIT"  
50  Wait: GOTO Wait  
60  Meas_done: !  
70  PRINT "Measurement Complete"  
80  END
```

Comments • The effect of this command is disabled by *CLS and *RST. in the Standard Event Status Register.

Related Commands *ESE
 *ESR?
 *SRE
 *STB?
 *WAI
 *OPC?

**OPC? (Operation Complete)*

***OPC? (Operation Complete)**

The *OPC? query places an ASCII "1" in the Output Queue when all pending operations (which include measurements in progress) are completed.

Syntax *OPC?

Example To initiate a measurement and synchronize on the Analyzer's completing the measurement:

```
10 OUTPUT 712;"INIT;*OPC?"
20 ENTER 712; A
30 PRINT "MEASUREMENT COMPLETE"
40 END
```

- Comments**
- The *OPC? query's holdoff action is canceled by power on and device clear.
 - The *OPC? query does NOT affect the Operation Complete bit in the Standard Event Status Register (see *OPC).

Related Commands *OPC
*WAI

**OPT? (Option Identification Query)*

***OPT? (Option Identification Query)**

This query causes the Analyzer to identify any installed options. The following options can be installed in the Analyzer:

- Option 001 Extended Memory — expands maximum memory size to 64k measurements (identified in the first field of the response).
- Option 030 Channel C — adds a third channel for high frequency measurements (identified in the second field of the response).
- Option 010 Oven Timebase — improves long term stability (identified in the third field of the response).
- Option 031 RF Communications — adds digital RF communications analysis and high resolution channel C (identified in the second and fourth fields of the response).

If all options are installed, the Analyzer sends a message with the following fields:

MEMORY , C , OVEN , RF

Empty fields are not allowed. A missing option is identified by an ASCII 0 (zero). Therefore, if only the extended memory option is installed, the Analyzer sends the following message:

MEMORY , 0 , 0 , 0

Syntax *OPT?

Example To determine the Analyzer's installed options:

```
10 OUTPUT 712; "*OPT?"
20 DIM A$[40]
30 ENTER 712; A$
40 PRINT "INSTALLED OPTIONS ARE"; A$
50 END
```

***RCL (Recall)**

***RCL (Recall)**

This command restores the state of the Analyzer from a copy stored in local non-volatile memory.

Syntax *RCL <NRf>

| ITEM | DESCRIPTION | RANGE/RESTRICTIONS |
|------|-----------------|--------------------|
| NRf | Recall Register | 0 to 9 |

Example

Comments

- The current state of the Analyzer is saved in recall register 0 when one of the following events occur:
 - a. A *RST or *RCL 1-9 command is executed.
 - b. A :SYSTem:PRESet command is executed.
 - c. The front-panel Preset function is activated.
 - d. Power to the Analyzer is turned off.
 - e. The front-panel Autoscale is executed.
 - f. A :CONFigure or :MEASure subsystem command is executed.
- The *RCL command has the same scope as the *RST command.
- RCL 0 does not cause current state to be saved.
- Error +2007 is generated when a recall fails because no save has occurred since the non-volatile RAM was cleared at the factory or since new ROMs were installed.

Related Commands *RST

***RST (Reset)**

***RST (Reset)**

This command resets the Analyzer's settings to their reset values. Reset values are shown in *table 1-1* "Subsystem Commands Summary" in chapter 1 as well as on the appropriate commands described in chapter 2.

It also places the Analyzer in "Operation Complete Idle State" and "Operation Complete Query Idle State" (IEEE 488.2). This results in disabling any prior *OPC command.

Syntax *RST

Example To reset the Analyzer to default parameters:

OUTPUT 712; "*RST"

Comments

- Using the *RST command is not the same as pressing the Preset key on the front panel.

Related Commands

***SAV (Save)**

*SAV (Save)

This command stores the current state of the Analyzer in local non-volatile memory.

Syntax *SAV <NRf>

| ITEM | DESCRIPTION | RANGE/RESTRICTIONS |
|------|---------------|--------------------|
| NRf | Save Register | 1 to 9 |

Example

Comments

- The current state of the instrument is saved in recall register 0 when one of the following events occur:
 - a. A *RST or *RCL 1-9 command is executed.
 - b. A :SYSTem:PRESet command is executed.
 - c. The front-panel Preset function is activated.
 - d. Power to the Analyzer is turned off.
 - e. The front-panel Autoscale is executed.
 - f. A :CONFigure or :MEASure subsystem command is executed.
- The *SAV command has the same scope as the *RST command.

Related Commands *RST

***SRE (Service Request Enable)**

***SRE (Service Request Enable)**

This command sets the Service Request Enable Register bits. This register determines which bits in the Status Byte can generate a service request interrupt (SRQ). Setting a bit to 1 in the Service Request Enable Register enables the corresponding bit in the Status Byte Register to generate an SRQ. Setting a bit to 0 disables the corresponding bit in the Status Byte Register from generating an SRQ.

Syntax *SRE <NRf>

*SRE?

| ITEM | DESCRIPTION | RANGE/RESTRICTIONS |
|------|---|--------------------|
| NRf | Sum of the weights of bits set in the Service Request Enable Register | 0 to 255 |

Example To enable the Message Available and Event Status bits to generate SRQ:

OUTPUT 712; **"*ESE 255; *SRE 48"**

(*ESE 255 enables all of the bits in the Standard Event Status Register to set the Event Status Bit in the Status Byte Register.)

Comments The weight of each bit in the Service Request Enable Register is shown in the following table.

***SRE (Service Request Enable)**

Service Request Enable Register

| BIT | WEIGHT | DESCRIPTION |
|-----|--------|---|
| 7 | 128 | OPR - Operation Status Summary Bit - one or more events in the Operation Status Register have occurred. |
| 6 | 64 | Not used. |
| 5 | 32 | ESB - Event Status Bit - one or more events in the Standard Event Status Register have occurred. |
| 4 | 16 | MAV - Message Available Bit - the Output Queue contains information. |
| 3 | 8 | QUE - Questionable Data/Signal Status Summary Bit - one or more events in the Questionable Data/Signal Status Register have occurred. |
| 2 | NA | Not Used |
| 1 | NA | Not Used |
| 0 | NA | Not Used |

- In response to *SRE?, the Analyzer returns a number between 0 to 63 or 128 to 191. (Bit 6, the RQS bit, cannot be set.) This number represents the total weight of all bits in the Service Request Enable Register that are set to 1.
- When power to the Analyzer is turned on, the Service Request Enable Register is set to 0.
- See Status Reporting in chapter 4 for additional information about the Status Byte Register.

Related Commands

- *ESE
- *ESR?
- *STB?

****STB? (Status Byte Query)***

***STB? (Status Byte Query)**

The *STB? query returns the current value of the Analyzer's Status Byte Register. The MSS (Master Summary Status) bit and not the RQS bit is reported on bit 6. The MSS bit indicates if the Analyzer has at least one reason for requesting service. If any bit is set to 1, the corresponding register or queue that sets that bit should be read to obtain additional information.

Syntax *STB?

Example. To read the Status Byte Register:

```
10  OUTPUT 712; "*STB?"
20  ENTER 712; A
30  DISP "STATUS BYTE ="; A
40  END
```

***STB? (Status Byte Query)**

Comments The weight of each bit in the Status Byte Register is shown in the following table.

Status Byte Register

| BIT | WEIGHT | DESCRIPTION |
|-----|--------|---|
| 7 | 128 | OPR - Operation Status Summary Bit - one or more events in the Operation Status Register have occurred. |
| 6 | 64 | RQS/MSS - Request Service/Master Summary Status Bit - the MSS bit indicates whether or not the device has at least one reason to request service. The MSS bit is sent in bit 6 of the STB? response and NOT in response to a serial poll. The RQS bit is contained in the serial poll response and is cleared by a serial poll. |
| 5 | 32 | ESB - Event Status Bit - one or more events in the Standard Event Status Register have occurred. |
| 4 | 16 | MAV - Message Available Bit - the Output Queue is not empty. |
| 3 | 8 | QUE - Questionable Data/Signal Status Summary Bit - one or more events in the Questionable Data/Signal Status Register have occurred. |
| 2 | NA | Not Used |
| 1 | NA | Not Used |
| 0 | NA | Not Used |

- *STB? does not directly alter the status byte, MSS/RQS bit or anything related to the generation of SRQ.

Related Commands *CLS
*ESE
*ESR?
*SRE

****TRG (Trigger)***

***TRG (Trigger)**

The *TRG command is the device-specific analog of the IEEE 488.2 defined Group Execute Trigger (GET) interface message. It has exactly the same effect as GET when received, parsed, and executed by the Analyzer.

The *TRG command initiates a measurement. If it is sent while a measurement is in progress, it is ignored and an error -211 is generated.

Syntax *TRG

Example To initiate a measurement and synchronize on the Analyzer's completing the measurement:

```
10  OUTPUT 712; "*TRG;*OPC?"
20  ENTER 712; A
30  PRINT "MEASUREMENT COMPLETE"
40  END
```

Comments

Related Commands :INIT:IMM
Group Execute Trigger (GET)

**TST? (Self-Test Query)*

***TST? (Self-Test Query)**

The *TST? query causes the Analyzer to execute an internal self-test and report if it detected any errors. A 0 (zero) response indicates that the test completed without detecting any errors. Any non-zero response indicates that the Analyzer detected an error. If any non-zero response is received, refer to the Service Manual or qualified service personnel for repair.

Syntax *TST?

Example To do a self-test query:

```
10  OUTPUT 712; "*TST?"
20  ENTER 712; A
30  PRINT A
40  END
```

- Comments**
- The Analyzer takes approximately 90 seconds to execute the self-test and respond to this query.
 - Error -330 is generated when the self-test fails.

Related Commands None

****WAI (Wait to Continue)***

***WAI (Wait to Continue)**

This command prevents the Analyzer from executing any further commands or queries until all pending operations (including measurements in progress) are completed.

Syntax *WAI

Example To initiate a measurement and synchronize reading out an error on the completion of the measurement:

```
10  OUTPUT 712;"INIT"  
20  OUTPUT 712;"*WAI"  
30  OUTPUT 712;"SYST:ERR?"  
40  ENTER 712;A  
50  PRINT "ERROR=";A  
60  END
```

Comments

- The *WAI command's holdoff action is canceled by power on and device clear. (*RST and *CLS have no affect on *WAI operation.)

Related Commands *OPC
 *OPC?

***WAI (Wait to Continue)**

3-24 HP 53310A Programming Reference Manual

STATUS REPORTING

In This Chapter

This chapter contains the following information:

- An overview of how (and why) the status of specified events in the Analyzer is monitored and how this information can be reported to a computer.
- How the status reporting structure is organized. The Analyzer conforms to the status reporting structure required by the SCPI standard. An important feature of all instruments that conform to the SCPI standard is that they all implement the required status reporting structure in the same way. Therefore, once you learn to use the SCPI standard status reporting structure, you can use the status reporting structure of any instrument that conforms to the SCPI standard.
- How to initialize and enable the status registers to monitor specific events and generate a service request (SRQ) interrupt.
- How to determine if a monitored event has occurred.
- How to interrupt an external computer.
- How to determine which monitored event has occurred.

Status Reporting Overview

An overview of the Analyzer's status reporting structure is shown in *figure 4-1*. This is the status reporting structure required by the SCPI standard. It monitors specified events in the Analyzer (such as when an error occurs or a message is available). The ability to monitor and report these events allows you to determine such things as the availability and the reliability of the measured data.

Status Reporting Overview

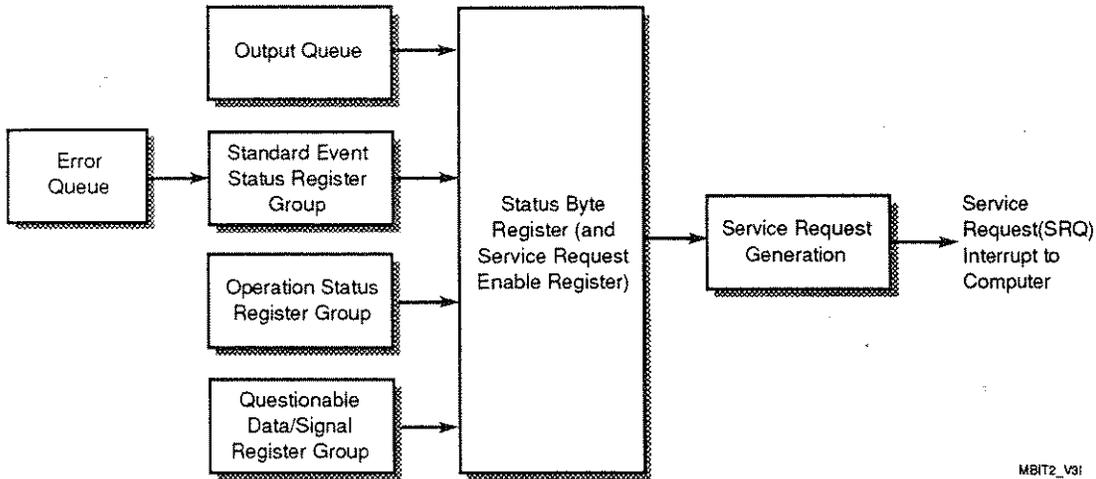


Figure 4-1. Status Reporting Overview Block Diagram

The status reporting structure consists of the following:

- The Questionable Data/Signal Status Register Group.
- The Operation Status Register Group.
- The Standard Event Status Register Group (including the Error Queue).
- The Output Queue.
- The Status Byte Register and the Service Request Enable Register.

Status Reporting Overview

Each of the register groups consists of more than one register. How all the components of the status reporting structure interact with each other and how the individual registers in each group function are described in this chapter.

NOTE

Not all bits in all registers are used. As you learn more about the Status Data Structures, you will also learn exactly what events the Analyzer can monitor and report.

Table 4-1 alphabetically lists and briefly defines all the acronyms, commands, and terms used in this chapter. The :STATus subsystem commands and Common Commands are described in more detail in chapters 2 and 3. If you need additional information about these commands, read the detailed explanations. However, because the terminology required by the SCPI standard can be confusing, you may find this short list convenient for quick reference.

Status Reporting Overview

Table 4-1. Acronyms and Terms

| Acronyms | Definition |
|------------|---|
| ESB | Event Status Summary Bit (bit 5 of the Status Byte Register). |
| MSS | Master Status Summary bit (bit 6 of the Status Byte Register). See RQS (this relationship is explained later in this chapter). |
| MAV | Message Available bit (bit 4 of the Status Byte Register). |
| <NRf> | Flexible number representation. The common and status commands use only positive integers. |
| OPR | Operation Status Summary bit (bit 7 of the Status Byte Register). |
| QUE | Questionable Data/Signal summary bit (bit 3 of the Status Byte Register). |
| RQS | Request Service bit (bit 6 of the Status Byte Register). See MSS (this relationship is explained later in this chapter). |
| Commands | Description |
| *ESE <NRf> | Common Command that sets bits in the Standard Event Enable Register. These bits enable the corresponding bits in the Standard Event Status Register to generate the summary bit ESB. This interaction is explained in this chapter. |
| *ESE? | Common Command that reads the bits that are set in the Standard Event Enable Register. |
| *ESR? | Common Command that reads the bits that are set in the Standard Event Status Register. |
| *RST | Common Command that resets the Analyzer to a known state (see Table 1-1 for summary of the reset conditions). |
| *CLS | Common Command that clears all Event Registers and the Error Queue. It does not affect any Enable registers. |
| *SRE <NRf> | Common Command that sets bits in the Service Request Enable Register. These bits enable the corresponding bits in the Status Byte Register to generate the SRQ interrupt. |
| *SRE? | Common Command that reads the bits that are set in the Service Request Enable Register. |
| :STATus | Keyword for the :STATus subsystem. A brief explanation and examples of how to use this subsystem's commands are contained in this chapter. Detailed descriptions of the individual commands are contained in chapter 2. |

Table 4-1. Acronyms and Terms (Continued)

| Commands | Description |
|-------------------------------|---|
| *STB? | Common Command that reads the bits that are set in the Status Byte Register. |
| :SYSTem:ERRor? | :SYSTem subsystem command that reads the oldest message from the error queue and removes that error from the error queue. |
| Terms | Description |
| Condition Registers | The :STATus Subsystem registers that monitor events in real-time. |
| Clear | Set register to all 0's. |
| Enable | This term indicates that a bit has been set in one of the enable registers. Most bits enable a corresponding bit in the associated event register to generate a summary bit. The Service Enable Register bits enable the corresponding bits in the Status Byte Register to generate an SRQ interrupt. |
| Event Registers | Registers that latch when an event occurs. |
| Set | This term indicates that a bit in a register has been set to a defined state. |
| Summary bit | This term indicates either a bit generated by an enabled bit in an event register or the resulting bit that is set in the Status Byte Register. |
| Transition Registers (Filter) | The :STATus Subsystem registers that determine which transition (rising or falling) of the condition register bit sets the corresponding event register bit. |

Determining How to Report Status Events

Before you can determine how to program the Analyzer for status reporting, you must first determine your reporting requirements. *Figure 4-2* shows the decisions that you must make to use the Analyzer's status reporting capability effectively. This illustration provides an introduction to status reporting and a quick reference guide when you use the status reporting system in applications.

NOTE

Figures 4-2 and 4-4 should be used in two ways. When you are first learning about status reporting, they provide an overview and an introduction to the material discussed in this chapter. After you know how to use status reporting, they provide a quick reference for programming status reporting.

The first step in programming the status reporting system is to select the events you want to monitor. As shown in figure 4-2, you can enable these specific bits in the appropriate event register. (The notes on the figure show how to enable any available bits in each event register.) A detailed description of each register group, its capabilities, and how to enable each event is contained in this chapter.

If you want the Analyzer to report events as soon as they occur, you must enable the related Status Byte Register bits to generate a service request (SRQ) to the computer.

If you do not want to use SRQ, you can periodically read the status information by using the appropriate :STATUS subsystem commands and Common Commands. This is the simplest method and requires the least amount of programming.

Once you have decided what you want to monitor and how you want to report this information to the computer, you must then decide what you are going to do with the information. You must write the subroutines to handle the service request interrupt, read the appropriate registers, and decode the decimal data returned from the Analyzer. When you have determined which events have occurred, you can decide on the appropriate response to the event.

Status Reporting Overview

Figure 4-2.
STATUS REPORTING FLOWCHART

(See Page 4-7)

4-8 HP 53310A Programming Reference Manual

Standard Status Data Structure

It is very important to understand the following two points:

- To monitor an event, you must first clear then enable that event. You clear all events when you perform initialization.
- To generate an SRQ interrupt to an external computer, you must enable at least one bit in the Status Byte Register.

How to accomplish these objectives is described in the rest of this chapter.

Evolution of the Status Data Structures

The Status Data Structures evolved in the following two major steps:

- The Status Byte Register, the Standard Event Status Register group, and the Output Queue are defined as the Standard Status Data Structure Model in IEEE 488.2-1987.
- The Operation Status Register group and the Questionable Data/Signal Status Register group are required by the SCPI standard 1.0 (1990).

Standard Status Data Structure

Figure 4-3 shows the Standard Status Data Structure Register Model which contains the following:

- The Status Byte Register.
- The Service Request Enable Register.
- The Standard Event Status Register.
- The Standard Event Enable Register.
- The Output Queue.
- The logic describing the relationship of the registers to each other and to the computer.

Standard Status Data Structure

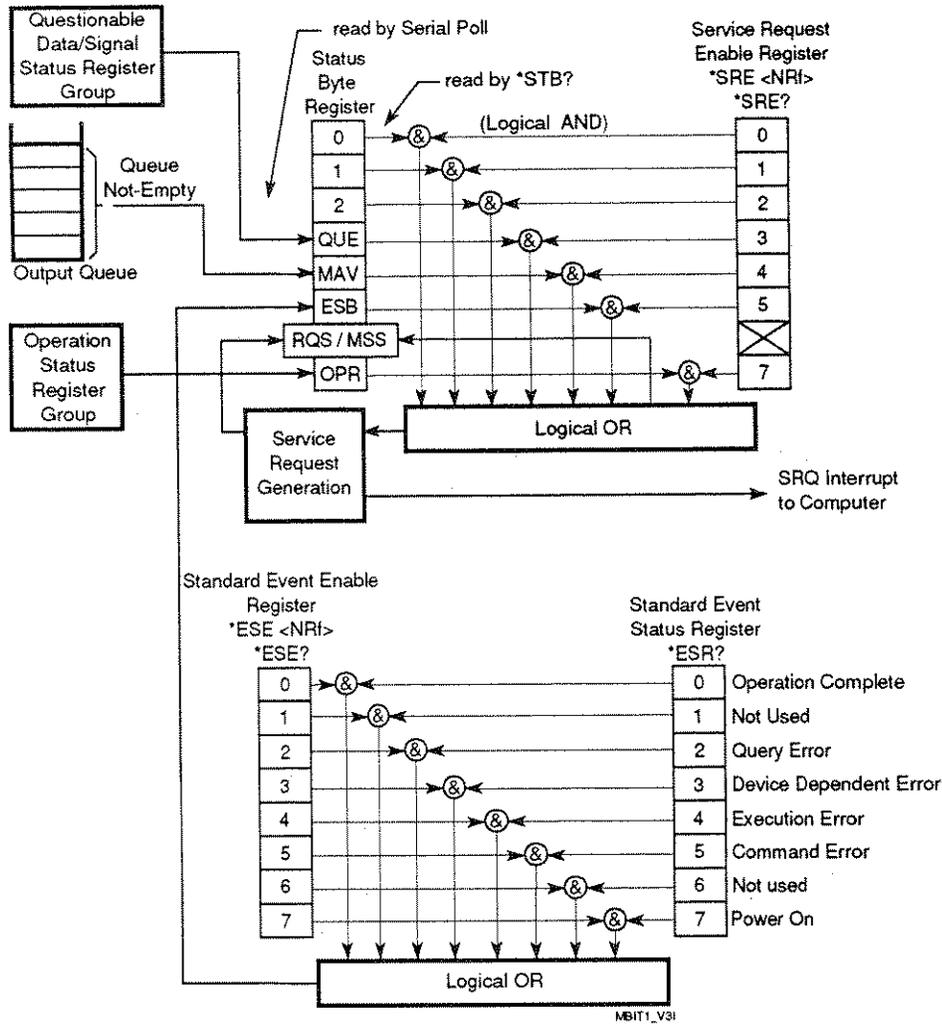


Figure 4-3. Standard Status Data Structure Functional Diagram

Standard Status Data Structure

You can enable and read these registers using Common Commands. Common Commands are easy to recognize because they all begin with an "*" (for example, *RST, *SRE, *STB, and so on). If you have programmed other HP-IB instruments, you are probably already familiar with this structure.

NOTE

The inputs from the Operation Status Register group and Questionable Data/Signal Status Register group are added only for completeness and clarity. These two register groups (which were added by the SCPI standard) are described separately.

Status Byte Register

The Status Byte Register is the summary-level register in the status reporting structure. It contains summary bits that monitor activity in the other status registers and queues. The Status Byte Register is a live register. That is, its summary bits are set and cleared by the presence or absence of summary bit from other event registers or queues.

If the Status Byte Register is to be used with the Service Request Enable Register to set bit 6 (RQS/MSS) and to generate an SRQ, at least one of the summary bits must be enabled and then set. *Table 4-2* lists the Status Byte Register bits and describes the event that sets each bit.

NOTE

Event bits in all the other status registers must be specifically enabled to generate the summary bit that sets the associated summary bit in the Status Byte Register. How to enable bits in all event registers is described when each event enable register is described.

Standard Status Data Structure

Table 4-2. Status Byte Register

| BIT | WEIGHT | DESCRIPTION |
|----------|--------|---|
| 7 | 128 | OPR - Operation Status Summary Bit - one or more events in the Operation Status Register occurred and set enabled bits in that register. |
| 6 | 64 | RQS/MSS - Request Service/Master Status Summary Bit - the MSS bit indicates whether or not the device has at least one reason to request service. The MSS bit is sent in bit 6 of the STB? response and NOT in response to a serial poll. The RQS bit is contained in the serial poll response and is cleared by a serial poll. |
| 5 | 32 | ESB - Event Status Bit - one or more events in the Standard Event Status Register occurred and set enabled bits in that register. |
| 4 | 16 | MAV - Message Available Bit - the Output Queue is not empty. |
| 3 | 8 | QUE - Questionable Data/Signal Status Summary Bit - one or more events in the Questionable Data/Signal Status Register occurred and set enabled bits in that register. |
| 0 thru 2 | NA | Not Used |

How to Read the Status Byte Register

You can read the Status Byte Register using either the *STB? Common Command or the HP-IB serial poll command. Both commands return the decimal-weighted sum of all the set bits in the register. The difference between the two methods is that the serial poll command reads bit 6 as the RQS (Request Service) bit and clears the bit which clears the SRQ interrupt. The *STB? command reads bit 6 as the MSS (Master Summary Status) and does not clear the bit or have any affect on the SRQ interrupt.

The use of bit 6 can be confusing. This bit was defined to cover all possible computer interfaces, including a computer that could not do a serial poll. The important point to remember is that, if you are using an SRQ interrupt to an external computer, the serial poll command clears bit 6. Clearing bit 6 allows the Analyzer to generate another SRQ interrupt when another enabled event occurs.

No other bits in the Status Byte Register are cleared by either method. A possible exception is Message Available bit (bit 4). If there are no other messages in the Output Queue, bit 4 (MAV) can be cleared as a result of reading the response to the *STB? command.

Standard Status Data Structure

Service Request Enable Register

The Service Request Enable Register bits enable corresponding bits in the Status Byte Register to request service. They also allow the enabled bits to set RQS and MSS (bit 6) in the Status Byte Register. For example, use the following command to set bit 4 (MAV) and bit 5 (ESB) in the Service Request Enable Register:

```
OUTPUT 712; *SRE 48"
```

This command uses the parameter (48) to enable the Analyzer to generate an SRQ interrupt under the following conditions:

- One or more bytes in the Output Queue set bits 4 (MAV).
- An enabled event in the Standard Event Status Register generates a summary bit that sets bit 5 (ESB).

You set other bits in the Service Request Enable Register using the *SRE command and read the bits that are set by using the *SRE? query.

Examples of Reading the Status Byte Register

The following two programs show how to read the Status Byte Register using the *STB? Common Command and the HP BASIC serial poll command.

Example 1 (Using *STB?)

The following program uses the *STB? command to read the contents of the Analyzer's Status Byte Register when none of the register's summary bits are enabled to generate an SRQ interrupt:

```
10 OUTPUT 712; *STB?"  
20 ENTER 712;A  
30 PRINT A  
40 END
```

Standard Status Data Structure

For example, assume bit 4 (weight = 16) and bit 5 (weight = 32) are set. The program prints the sum of the two weights (48). Because these bits were not enabled to generate an SRQ, bit 6 (weight = 64) is not set.

Example 2 (Using Serial Poll)

The following program uses the HP BASIC serial poll (SPOLL) command to do the same thing when either (or both) bits are enabled to generate an SRQ interrupt:

```
10P=SPOLL(712)
20 PRINT P
30 END
```

Besides printing 112 (48 + 64 because bit 6 is read), this program also clears bit 6 (RQS) of the Status Byte Register. The difference in the decimal value returned by the two programs (64) is the value of bit 6. Bit 6 is set when the first enabled summary bit is set and is cleared when the Status Byte Register is read by the serial poll command.

Most programmers use serial polling inside interrupt service routines because it resets bit 6 and allows the next enabled event that occurs to generate a new SRQ interrupt. There is an example of how to do this later in the chapter.

Notice in both examples that the decimal value of the bits set in the Status Byte Register are read by the computer. However, you must then write a subroutine to evaluate the value read and decide what additional action should be taken.

Standard Status Data Structure

Standard Event Status Register

The Standard Event Status Register monitors the Analyzer status events listed in *Table 4-3*. When one of these events occurs, it sets the corresponding bit in the register. The bits set in this register generate a summary bit to set bit 5 (ESB) in the Status Byte Register only if they are specifically enabled.

Table 4-3. Standard Event Status Register

| BIT | WEIGHT | DESCRIPTION |
|-----|--------|---|
| 7 | 128 | Power on. Power has been applied to the Analyzer. |
| 6 | NA | Not used |
| 5 | 32 | Command Error (-100 thru -199). Indicates a syntax error has been detected by the Analyzer's parser. |
| 4 | 16 | Execution Error (-200 thru -299). Indicates the current command cannot be executed because of an out of range parameter or conflicting settings. |
| 3 | 8 | Device Specific Error (-300 thru -399 or +2000 thru +2011). Indicates the Analyzer has detected a device-specific error (one that is not a command error, query error, or execution error). |
| 2 | 4 | Query Error (-400 thru -499). Indicates either an attempt was made to read the Output Queue when it was empty or output data has been lost. |
| 1 | NA | Not used. |
| 0 | 1 | Operation Complete. Indicates the Analyzer has completed any pending operations and is ready to accept new commands. This bit is set only in response to the *OPC command. |

Enabling Standard Event Status Register Bits

To make it possible for any of the Standard Event Status Register bits to be able to generate a summary bit, you must first enable that bit. You enable a bit by using the *ESE Common Command to set the corresponding bit in the Standard Event Enable Register.

For example, suppose your application requires an interrupt whenever any type of error occurs. The error related bits in the Standard Event Status Register are bits 2 through 5. The sum of the decimal weights of these bits is 60. Therefore, you can enable any of these bits to generate the summary bit by sending:

OUTPUT 712; *ESE 60'

Standard Status Data Structure

Whenever an error occurs, it sets one of these bits in the Standard Event Status Register. Because the bits are all enabled, a summary bit is generated to set bit 5 (ESB) in the Status Byte Register.

If bit 5 (ESB) in the Status Byte Register is enabled (via *SRE), an SRQ service request interrupt is sent to the external computer. See "Interrupting an External Computer" later in this chapter for an example program.

Note that the Standard Event Status Register bits that are not enabled still respond to their corresponding conditions (that is, they are set if the corresponding event occurs). However, because they are not enabled, they do not generate a summary bit to the Status Byte Register.

NOTE

*You can read the Standard Event Status Register using the *ESR? query (this also clears the register). You can determine which bits are enabled by using the *ESE? query. Both of these queries return the decimal weighted sum of all set and enabled bits, respectively.*

Error Queue

As errors are detected, they are placed in an error queue. This queue is first in, first out. If the error queue overflows, the last error in the queue is replaced with error -350, "Too many errors". Any time the queue overflows, the least recent errors remain in the queue, and the most recent error is discarded. The length of the Analyzer's error queue is 30 (29 positions for the error messages, and 1 position for the "Too many errors" message). Reading an error from the head of the queue removes that error from the queue, and opens a position at the tail of the queue for a new error, if one is subsequently detected.

When all errors have been read from the queue, subsequent error queries return +0, "No error".

Standard Status Data Structure

The error queue is cleared when any of the following occur:

- Upon power up.
- Upon receipt of a *CLS command.
- Upon reading the last item from the queue.
- Upon transition from talk only to addressed mode on the front panel.

Reading the Error Queue

Executing the `SYSTEM:ERRor?` command reads the oldest message from the error queue and removes that error from the error queue. The `SYST:ERR?` response has the following format:

`<error number>,"<error description string>"`

An example of an error message is:

`-113,"Undefined header"`

Positive error numbers are specific to the Analyzer. Negative error numbers are command language-related. These messages are described in appendix F, "Error Messages." Errors also set a corresponding bit in the Standard Event Status Register

The following short program reads all error messages (one message at a time, oldest to newest) from the error queue. After each error message is read, it is automatically removed from the error queue. When the error queue is empty, the program returns the `+0,"No Error"` message.

```
10 ASSIGN @Anlz TO 712
20 !Assign path name
30 DIM Message$(255)
40 !Creates array for error message
50 REPEAT
60 !Repeats until error = 0
70 OUTPUT @Anlz;"SYST:ERR?"
80 !Read error number and message
90 ENTER @Anlz;Code,Message$
100 !Enter error number and message
110 PRINT Code,Message$
```

SCPI Status Reporting

```
120 !Print error number and message
130 UNTIL Code = 0
140 END
```

NOTE

An example of how this program is incorporated into an interrupt-driven application program is shown later in this chapter.

Output Queue

The Output Queue stores all Analyzer-to-computer messages (error messages are routed through the Output Queue). The Output Queue generates the Message Available summary bit when the Output Queue contains one or more bytes. This summary bit sets the MAV bit (bit 4) in the Status Byte Register.

One way the Output Queue can be read is by using the HP BASIC ENTER statement.

Reading the Output Queue

The following short program queries the Analyzer to identify itself and then reads and prints the message from the Output Queue:

```
10 OUTPUT712;"*IDN?"
20 DIM A$(40)
30 ENTER 712;A$
40 PRINT "ANALYZER IS";A$
50 END
```

SCPI Status Reporting

Figure 4-4 shows all the status system register groups and queues in the Analyzer. This is a high level drawing that does not show all the registers that are contained in each group. It is intended as a guide to the bits used in each of these register groups to monitor the Analyzer's status. Note that besides the Operation Status and the Questionable Data/Signal Register groups, a summary of the Standard Status Structure Registers (defined by IEEE 488.2-1987) is shown.

SCPI Status Reporting

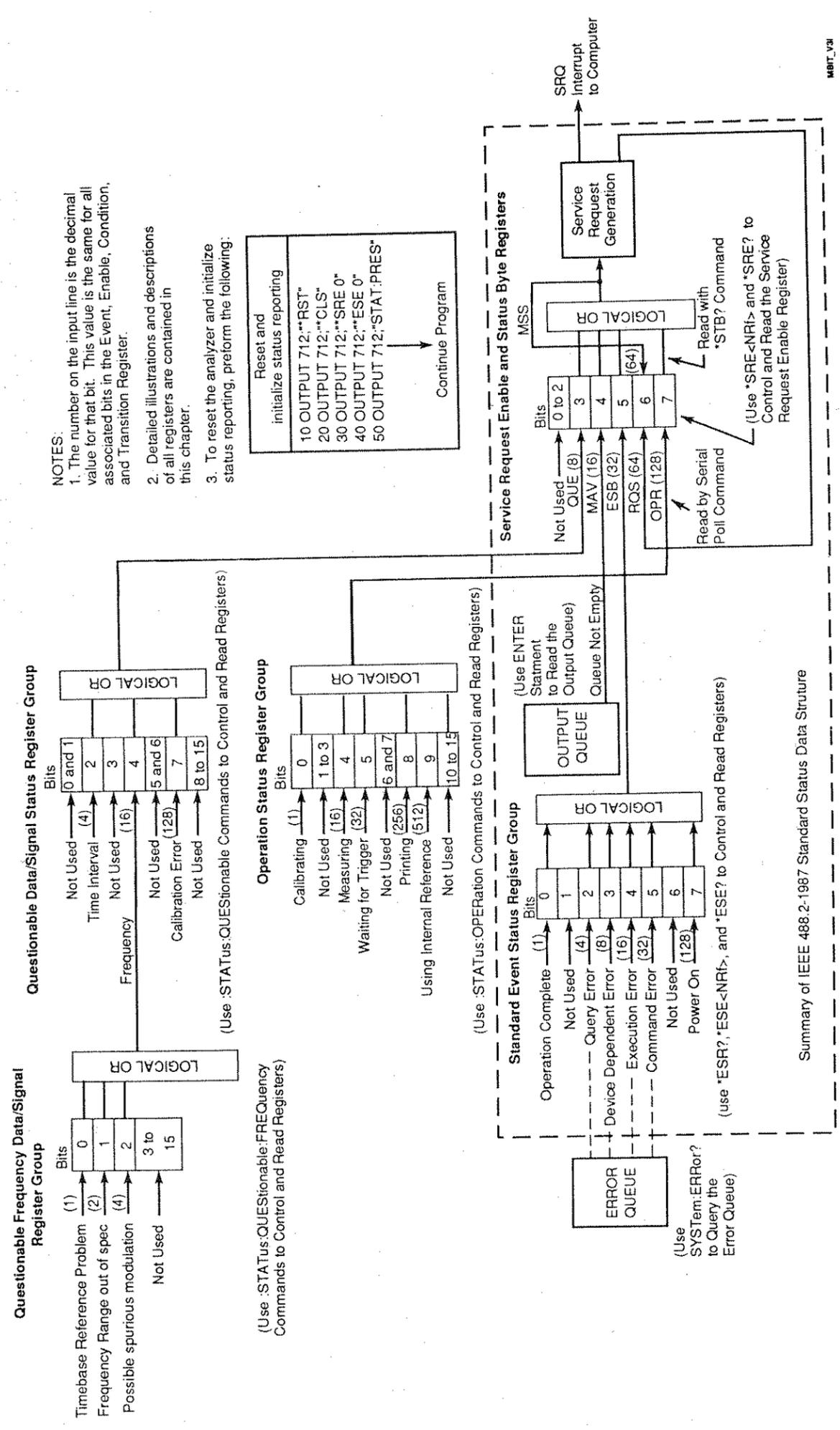


Figure 4-4. SCPI Status Reporting Summary Functional diagram

SCPI Status Reporting

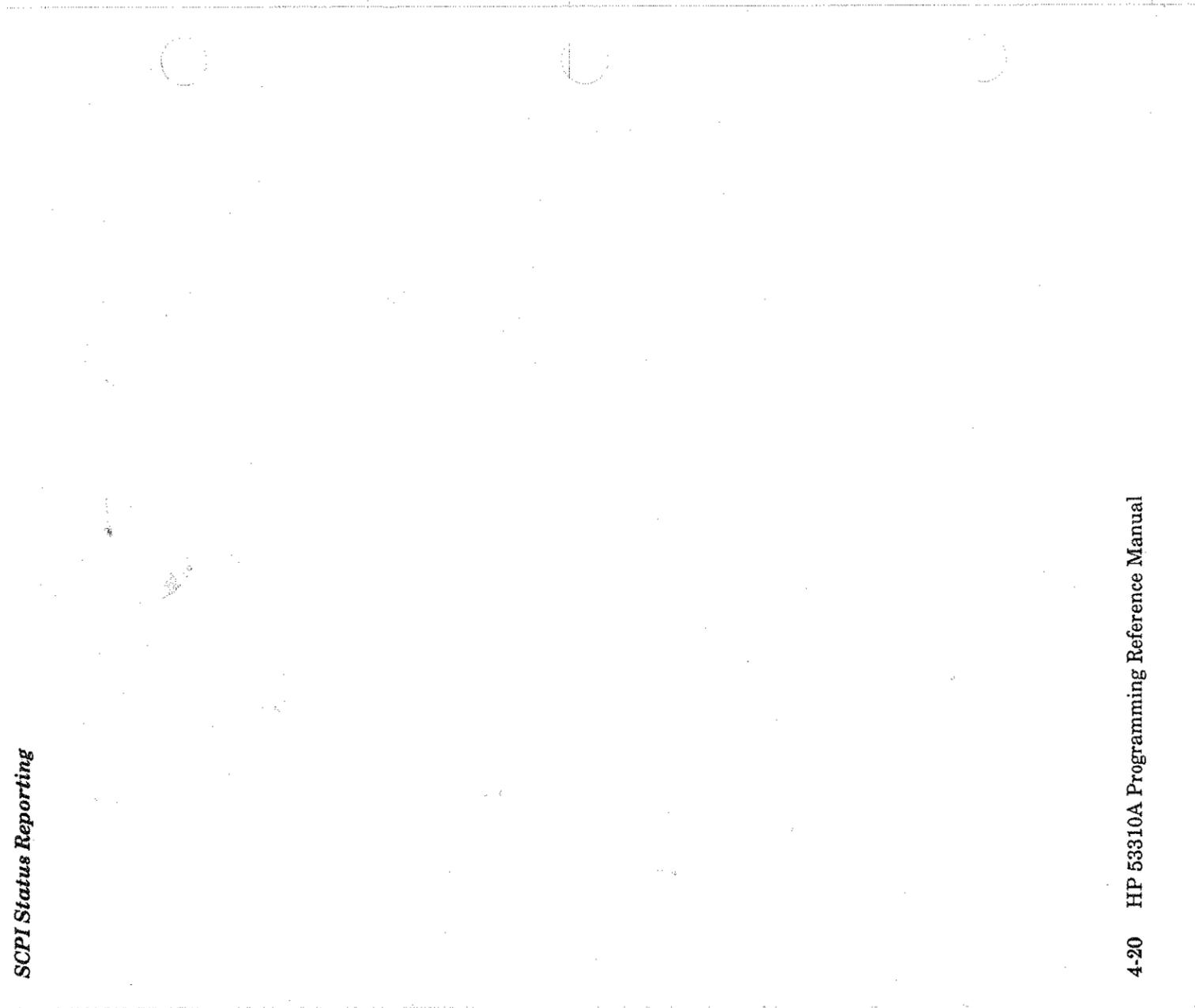


Figure 4-4.
SCPI STATUS REPORTING SUMMARY FUNCTIONAL DIAGRAM

(See Page 4-19)

4-20 HP 53310A Programming Reference Manual

The drawing also shows that summary bits from the Operation and Questionable register groups are sent to the Status Byte Register. Chapter 2 “Subsystem Commands Reference Dictionary” and chapter 3 “Common Commands Reference Dictionary” contain additional information on how to obtain status information.

SCPI Status Register Model

The Operation Status Register group and the Questionable Data/Signal Status Register group are the additional status registers required by the SCPI standard. *Figure 4-5* shows the structure of either register group. Each group has a complete set of registers that consist of the following:

- A Condition Register.
- A Transition Filter (that contains a Negative Transition Register and a Positive Transition Register).
- An Event Register.
- An Event Enable Register.
- The logic necessary to enable and generate a summary bit.

How the individual registers work is described first. Then how you can configure and read these registers using the :STATus subsystem commands is briefly explained.

SCPI Status Reporting

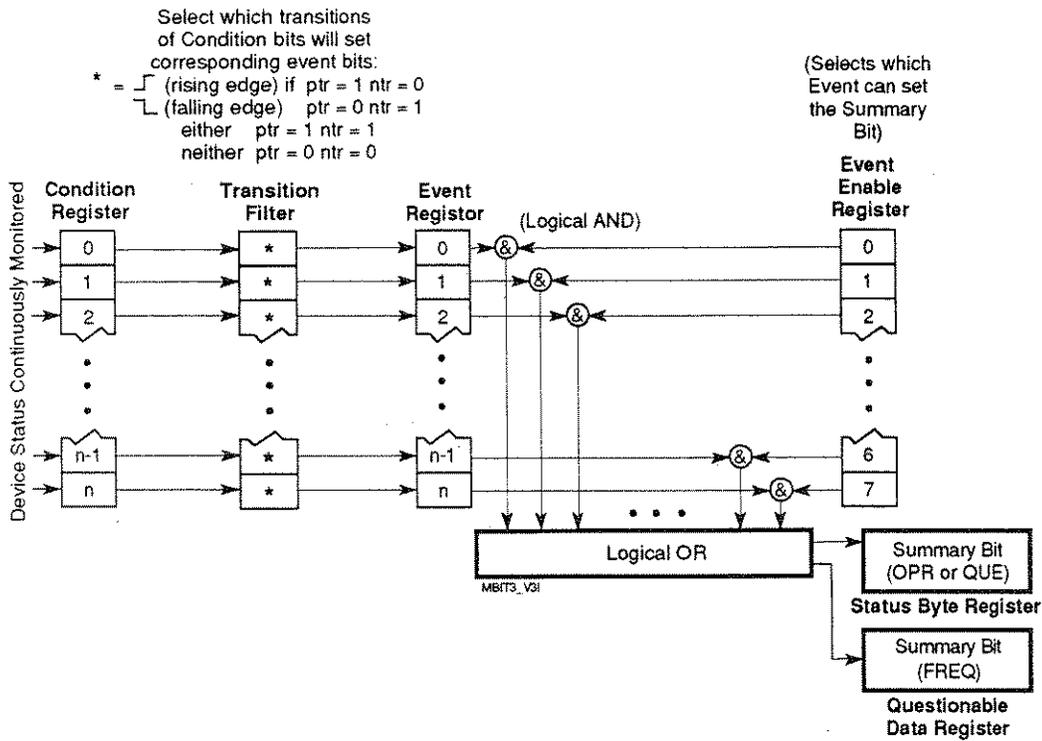


Figure 4-5. SCPI Status Reporting Model Functional Diagram

Condition Register

The condition register continuously monitors the hardware and firmware status of the Analyzer. There is no latching or buffering for this register, it is updated in real time. Condition registers can be read using the appropriate CONDITION? query. Reading a condition register does not change its contents.

Transition Filter

The transition filter controls the type of bit state changes in the condition register that set the corresponding bits in the event register. The transition filter contains two registers (a positive transition register and a negative transition register).

The bits in these registers can be set for positive transitions (PTR), negative transitions (NTR), both, or none. When a positive transition register bit is set and the corresponding condition bit changes from 0 to 1, the associated bit in the event register is set. When a negative transition register bit is set and the corresponding condition bit changes from 1 to 0, the associated bit in the event register is set.

When both a negative transition register bit and the associated positive transition bit are set and the corresponding condition bit changes from 1 to 0 or from 0 to 1, the associated bit in the event register is set. When both the negative transition register bit and the associated positive transition bit are 0, the associated bit in the event register is never set.

Transition filters can be read or set using the PTRansition and NTRansition commands. Transition filters are unaffected by *CLS (clear status) or queries. :STATus:PRESet and power-on sets them to their default values.

Event Register

The event register latches transition events from the condition register as specified by the transition filter. Bits in the event register are latched and, once set, they remain set until cleared by an event register query or *CLS command. There is no buffering. Therefore, once an event bit is set, subsequent events corresponding to that bit are ignored.

SCPI Status Reporting

Event Enable Register

The event enable register specifies which bits in the event register can generate a summary bit. The Analyzer ANDs corresponding bits in the enable and event registers and ORs all the resulting bits to generate a summary bit. Summary bits set corresponding bits in the Status Byte Register (OPR or QUE) or Questionable Register (FREQ).

:STATus Subsystem Commands

The :STATus subsystem commands allow you to specify or examine the status data structures of the Operation Status and the Questionable Data/Signal Register groups. Subsystem commands use a hierarchy relationship between commands that is indicated by a ":" (colon). For example, STAT:OPER:COND? uses the ":" between STAT and OPER to show that OPER is subordinate to STAT.

:STATus Subsystem Syntax

Figure 4-6 is a syntax diagram of the STATus subsystem.

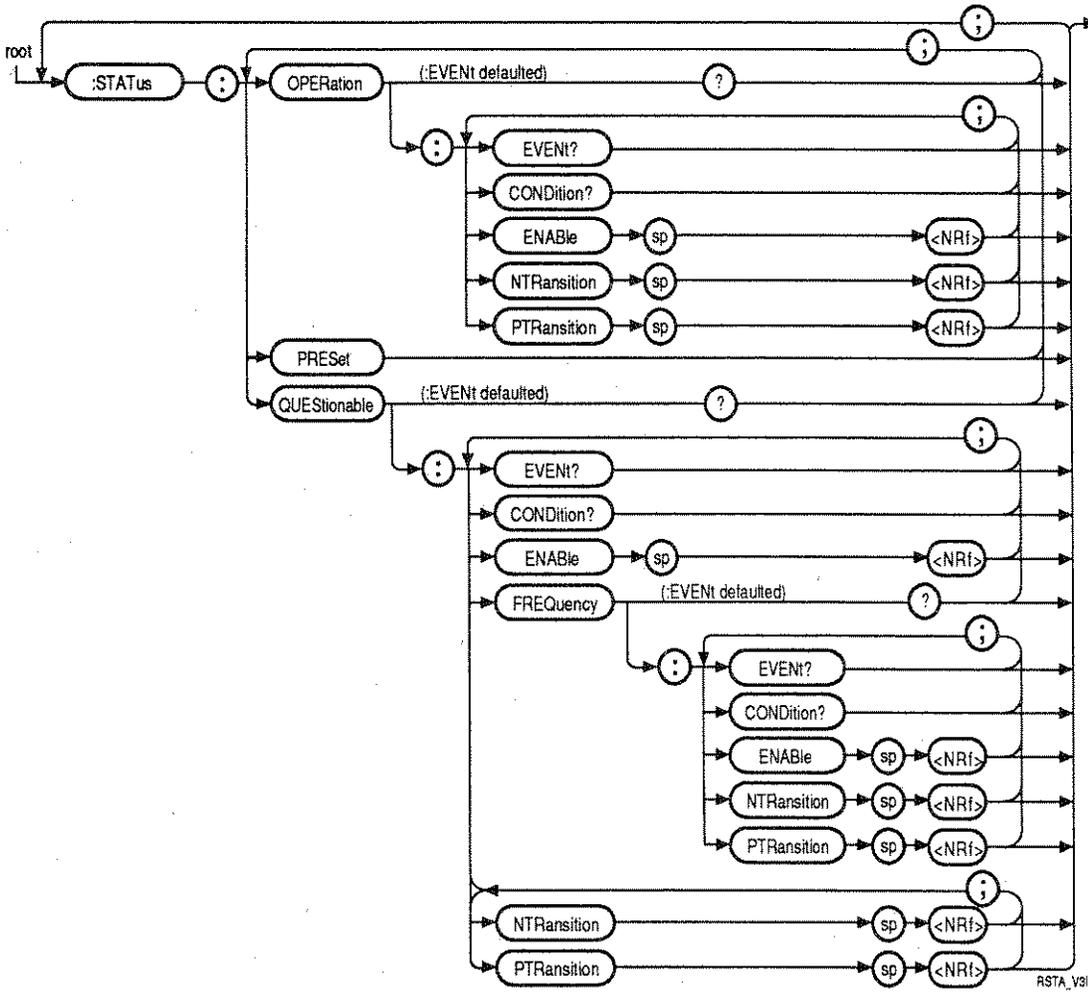


Figure 4-6. :STATus Subsystem Syntax Diagram

SCPI Status Reporting

:STATUS:PRESet Command This command presets the Operation and Questionable Data/Signal Register groups to the values shown in the Table 4-4. It has no query form.

Table 4-4. Status Reporting Preset Conditions

| Register Group | Enable/Transition Registers | :STATUS:PRESet value = |
|-------------------------|-----------------------------|------------------------|
| :OPERation | :ENABle | 0's |
| :OPERation | :NTRansition | 0's |
| :OPERation | :PTRansition | 1's |
| :QUESTionable | :ENABle | 0's |
| :QUESTionable | :NTRansition | 0's |
| :QUESTionable | :PTRansition | 1's |
| :QUESTionable:FREQuency | :ENABle | 1's |
| :QUESTionable:FREQuency | :NTRansition | 0's |
| :QUESTionable:FREQuency | :PTRansition | 1's |

Example of Initializing Status Reporting

Use the following program to initialize status reporting:

```

OUTPUT712;"*CLS"
!Clears all Event Registers and the Error Queue
OUTPUT 712;"*SRE 0"
!Clears Service Request Enable register
OUTPUT 712;"*ESE 0"
!Clears Event Status Enable register
OUTPUT 712;"*STATUS:PRESet"
!Presets Operation and Questionable enable
registers and transition filters to values shown in
table 4-4.
    
```

:STATUS:OPERation Subtree

The :STATUS:OPERation subtree commands allow you to examine the status of the Analyzer monitored by the Operation Status Register group. The Operation Status Register group consists of a condition register, positive and

SCPI Status Reporting

negative transition registers, an event register, and an enable register (see *figure 4-5*). The commands in this subtree allow you to control and monitor these registers.

The Operation Status Register group monitors operations currently being performed. The Analyzer implements bits 0, 4, 5, 8, and 9 (see *table 4-5*). The bit numbers are identical for all the registers in the Operation Status Register group.

Table 4-5. Operation Status Register Bits

| BIT | WEIGHT | DESCRIPTION |
|------------|--------|--------------------------|
| 10 thru 15 | NA | Not Used |
| 9 | 512 | Using Internal Reference |
| 8 | 256 | Printing |
| 6 and 7 | NA | Not Used |
| 5 | 32 | Waiting for Trigger |
| 4 | 16 | Measuring |
| 1 thru 3 | NA | Not Used |
| 0 | 1 | Calibrating |

:STATus:QUEStionable Subtree

The :STATus:QUEStionable subtree commands allow you to examine the status of the Analyzer monitored by the Questionable Status Register group. The Questionable Status Register group consists of a condition register, positive and negative transition registers, an event register, and an enable register (see *figure 4-5*). The commands in this subtree allow you to control and monitor these registers.

The Questionable Status Register group monitors marginal conditions in the Analyzer. The Analyzer implements bits 2, 4, and 7 (see *table 4-6*). The bit numbers are identical for all the registers in the Questionable Status Register group.

Bit 4 (Questionable frequency measurements) is used in conjunction with the Questionable Frequency Status Register group to extend the questionable frequency

SCPI Status Reporting

monitoring capabilities. Bit 4 indicates that frequency measurements are of questionable quality due to a timebase reference problem, the selected frequency range including frequencies outside of the specification limits, or possible spurious modulation.

Bit 2 (Questionable time interval measurements) indicates that time interval measurements are of questionable quality due to a timebase reference problem.

Table 4-6. Questionable Status Register Bits

| BIT | WEIGHT | DESCRIPTION |
|-----------|--------|----------------------------|
| 8 thru 15 | NA | Not Used |
| 7 | 128 | Calibration error detected |
| 5 and 6 | NA | Not Used |
| 4 | 16 | Frequency (Summary Bit) |
| 3 | NA | Not Used |
| 2 | 4 | Time Interval |
| 0 and 1 | NA | Not Used |

:STATus:QUEStionable :FREQuency Subtree

The :STATus:QUEStionable:FREQuency subtree commands allow you to examine the status of the Analyzer monitored by the Questionable Frequency Status Register group. The Questionable Frequency Status Register group consists of a condition register, positive and negative transition registers, an event register, and an enable register (see *figure 4-5*). The commands in this subtree allow you to control and monitor these registers.

The Questionable Frequency Status Register group monitors conditions under which the frequency measurements are of questionable quality. The Analyzer implements bits 0, 1 and 2 (see *table 4-7*). The bit numbers are identical for all the registers in the Questionable Frequency Status Register group. These bits are used in conjunction with bit 4 (Frequency) of the Questionable Status Register group to extend the frequency monitoring capabilities.

SCPI Status Reporting

Table 4-7. Questionable Frequency Status Register Bits

| BIT | WEIGHT | DESCRIPTION |
|-----------|--------|--|
| 3 thru 15 | NA | Not Used |
| 2 | 4 | Possible spurious modulation (see appendix C of the Operating Reference Manual). |
| 1 | 2 | Selected frequency range out of specification |
| 0 | 1 | Timebase reference error |

Using :STATus Subsystem Commands

Using the :STATus Subsystem commands to monitor the defined Analyzer capabilities is similar to using the Standard Event Status Register group in that you must first clear and enable the specific events you want to monitor.

The biggest difference is in the use of the transition filter's (NTRansition and PTRansition) registers to control which edge of the condition registers signal sets the event register. If you merely use the default conditions established by the :SYSTem:PRESet command, all events are set by the positive (rising) transition of the condition register bits.

However, to illustrate the transition filter's capabilities, the following example shows how to set the :STATus:OPERation register group to monitor when the Analyzer is waiting for a trigger (positive transition) and when the measurement is complete (negative transition).

NOTE

This example does not show a complete program example.

Has a Monitored Event Occurred?

```
10 OUTPUT 712;":STAT:PRES"  
20 OUTPUT 712;":STAT:OPER:PTR 32"  
30 OUTPUT 712;":STAT:OPER:NTR 16"  
40 END
```

Has a Monitored Event Occurred?

You can determine if a monitored event has occurred by enabling a service request (SRQ) to interrupt the computer and using the HP BASIC serial poll (SPOLL) or by periodically reading the Status Byte Register using a serial poll or *STB? query. There is a slight distinction between the information that is returned by these two commands. This difference is described in the definition of the Status Byte Register.

Enabling a Service Request (SRQ)

Typically, you program the Analyzer to send a Service Request (SRQ) to the computer to indicate that a monitored event has occurred. It can mean that data is ready, that an abnormal condition exists, or it can be used as a means of synchronizing the program or other devices to the Analyzer. Figure 4-7 shows the service request generation function in more detail. Once you have set the required bits in the Service Request Enable Register, the generation of an SRQ interrupt is automatic. How to use this interrupt with an external computer is described next.

Has a Monitored Event Occurred?

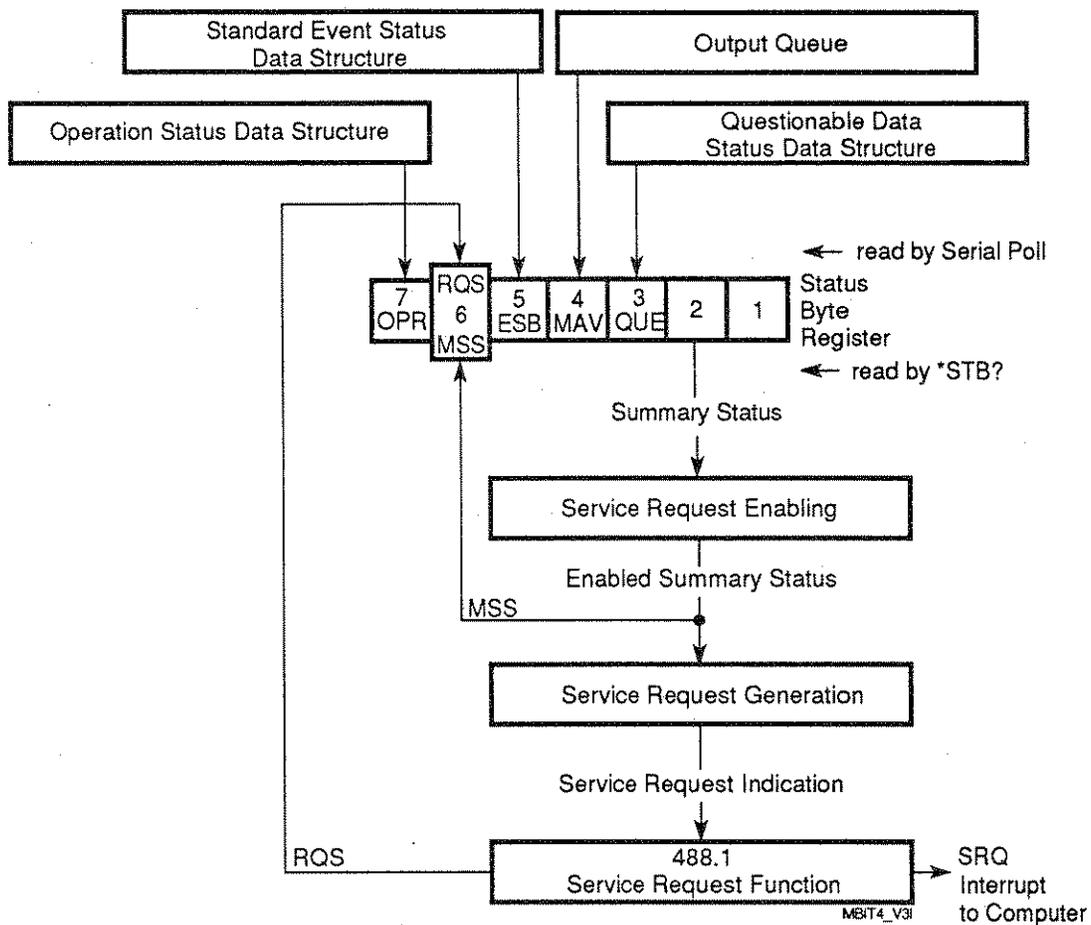


Figure 4-7. SRQ Interrupt Functional Diagram

Interrupting an External Computer

NOTE

*As mentioned earlier, you are not required to enable SRQ in order to monitor status. The Status Byte Register can be read at anytime using either the serial poll command or the *STB? Common Command.*

Interrupting an External Computer

When an enabled bit in the Status Byte Register is set, the Analyzer generates the SRQ interrupt. This line can be used as an interrupt to alert an external computer to suspend its present operation and determine what service the Analyzer requires. (Refer to your computer/language manuals for information on how to program your computer to respond to the interrupt.)

To allow any of the Status Byte Register bits to generate the SRQ interrupt, you must first enable those bits using the *SRE Common Command. For example, suppose your application requires an interrupt whenever a message is available in the Output Queue. Because Status Byte Register bit 4 (weight = 16) monitors the Output Queue, you can enable bit 4 by sending:

```
OUTPUT 712;"*SRE 16"
```

NOTE

*If in doubt, you can determine which of the Status Byte Register bits are enabled by using the *SRE? (Service Request Enable query). This query returns the decimal weight of the enabled bits.*

Interrupting an External Computer

The following example shows how to interrupt an external computer whenever an error occurs in the Analyzer:

```
10   OUTPUT 712;"*RST;*CLS;*SRE 32;*ESE 60"
20   ON INTR 7 CALL Errmsg
30   ENABLE INTR 7;2
40   OUTPUT 712;"TYPO" !COMMAND ERROR
50   OUTPUT 712;"INP:COUP &" !COMMAND ERROR
60   Wait:GOTO Wait
70   END
80   SUB Errmsg
90     DIM Message$(255)
100    Status_byte=SPOLL(712)
110    PRINT "STATUS BYTE ",Status_byte
120    OUTPUT 712;"*ESR?"
130    ENTER 712;Esr
140    PRINT "ESR ',Esr
150    REPEAT
160      OUTPUT 712;"SYST:ERR?"
170      ENTER 712;Code,Message$
180      PRINT Code,Message$
190    UNTIL Code=0
200    STOP
210  SUBEND
```

What Monitored Event Has Occurred?

What Monitored Event Has Occurred?

Determining which monitored event has occurred can be simple if you only have one event enabled or more complex if you have all (or several) events enabled. You must read the Status Byte Register and analyze the returned value to determine which event register set the summary bit and then read and analyze that event register. Sometimes reading the Error Queue or Output Queue provides all the information you need.

DATA TRANSFER

In This Chapter

This chapter describes how to transfer data between the Analyzer and an external computer. Explanations and examples of different types of data transfer are provided. The information is presented in the following order:

- An overview of the Analyzer's data transfer capabilities is provided.
- The commands used when transferring measured data values are described.

Data Transfer Overview

When discussing data transfer, it is important to understand that data transfer consists of several levels of complexity. At its simplest level, to understand how your Analyzer and computer communicate and transfer data, you must understand the concept of SCPI program and response messages (see figure 5-1). These messages are the formatted data sent between the computer and the Analyzer. Program messages contain one or more commands and response messages contain one or more responses.

The computer is in control of the HP-IB. The computer can send commands (program messages) at any time. The Analyzer only sends response messages when specifically queried to do so. All SCPI query commands end in a ? (question mark). Queries return either measured data values or instrument status information. Any internal setting that you can program can also be queried.

In the first part of this chapter, you will learn the basic mechanics of transferring data. This includes most of the queries used to obtain instrument status information and analysis results. In the second part of the chapter, how to

Data Transfer Overview

transfer measured data values is described. This type of data transfer is more complicated and needs a greater understanding of the many options that are available.

NOTE

*The programming hints shown are for HP BASIC.
The commands may be slightly different for other programming languages.*

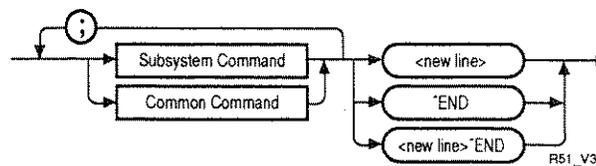
Forgiving Listening and Precise Talking

SCPI uses the concept of forgiving listening and precise talking that is defined in IEEE 488.2-1987. Forgiving listening means that the Analyzer is very flexible in accepting various command and parameter formats. For example, to set auto-triggering to ON, the Analyzer accepts either :TRIGger:AUTO ON or :TRIG:AUTO 1.

Precise talking means the response message format for a particular query is always the same. For example, if you query the auto-triggering status when it is ON by sending the query :TRIGger:AUTO?, the response is always a 1.

Program Message Syntax

Figure 5-1 shows the simplified syntax of a program message. You can send Common Commands and Subsystem commands in the same program message. If you send more than one command in one message, you must separate the commands with a semicolon.



NOTE:
<new line> = ASCII character decimal 10
^END = EOI asserted concurrent with last byte

Figure 5-1. Simplified Program Message Syntax Diagram

Data Transfer Overview

You must always end a program message with one of the terminators shown in *figure 5-1*. The word ^END means that EOI (End-or-Identify) is asserted on the HP-IB interface at the same time the last data byte is sent.

Most programming languages send these terminators automatically. For example, if you use the HP BASIC OUTPUT statement, <new line> is automatically sent after your last data byte. If you are using a PC with a language other than HP BASIC, you can usually configure your system to send whatever terminator you specify.

Subsystem Command Syntax

Figure 5-2 shows the simplified syntax of a Subsystem command. You must use a space (SP) between the last command mnemonic and the first parameter in a Subsystem Command. Note that if you send more than one parameter with a single command, you must separate adjacent parameters with a comma.

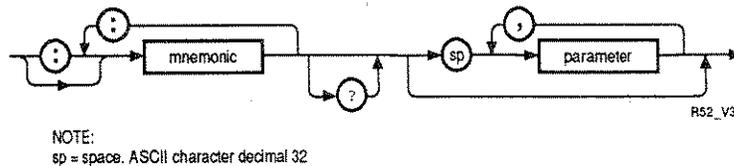
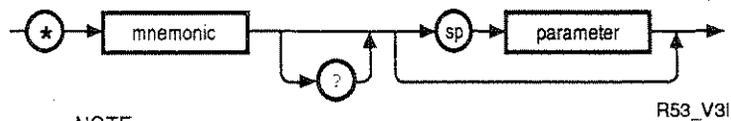


Figure 5-2. Simplified Program Command Syntax Diagram

Data Transfer Overview

Common Command Syntax

Figure 5-3 shows the simplified syntax of a Common Command. You must use a space (SP) between the command mnemonic and the parameter in a Common Command.



NOTE:
sp = space. ASCII character decimal 32

Figure 5-3. Simplified Common Command Syntax Diagram

Program Message Parameter Types

Table 5-1 contains explanations and examples of program message parameter types. Parameter types may be numeric value, Boolean, literal, string gap, NRf, or non-decimal numeric.

Table 5-1. Program Message Parameter Types

| TYPE | EXPLANATIONS AND EXAMPLES |
|-----------------------|---|
| <numeric value> | <p>Accepts all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation: 123, 123E2, -123, -1.23E2, .123, 1.23E-2, 1.23000E-01.</p> <p>Special cases include MINimum and MAXimum (also DEFault in the :CONFigure and :MEASure Subsystems only).</p> <p>MINimum selects minimum value available. MAXimum selects maximum value available. DEFault selects the default value. Queries using MINimum or MAXimum return the associated numeric value.</p> |
| <Boolean> | <p>Represents a single binary condition that is either true or false: 1 or ON, 0 or OFF (Query response returns only 1 or 0.)</p> |
| <literal> | <p>Selects from a finite number of choices. These parameters use mnemonics to represent each valid setting. An example is the INPut:COUpling AC DC command parameters (AC DC).</p> |
| <NRf> | <p>Flexible numeric representation. Only positive integers are used for NRf parameters in the Analyzer.</p> |
| <string> | <p>A string parameter is ASCII characters enclosed in double or single quotes ("FREQ", 'FREQ'). For example, OUTPUT 712;"FUNC ""FREQ"" is a valid program statement using a string parameter.</p> |
| <non-decimal numeric> | <p>Format for specifying hexadecimal (#H1F), octal (#Q1077), and binary (#B10101011) numbers using ASCII characters. Used in :STATus subsystem commands.</p> |

Response Message Syntax

Figure 5-4 shows the simplified syntax of a Response Message. Response Messages may contain both commas and semicolon separators. When a single query command returns multiple values, a comma is used to separate each data item. When multiple queries are sent in the same program message, the groups of data corresponding to each query are separated by a semicolon. Note that a <new line> ^END is always sent as a response message terminator.

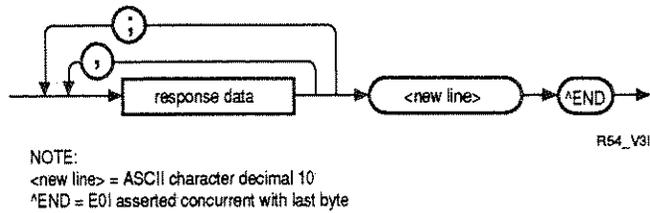


Figure 5-4. Simplified Response Message Syntax Diagram

Data Transfer Overview

Response Message Data Types

Table 5-2 contains explanations of response data types.

Table 5-2. Response Message Data Types

| Type | Description |
|-----------------------|--|
| ASCIi | Numeric data is transferred as ASCII bytes in <NR1>, <NR2>, or <NR3> format, as appropriate. The numbers are separated by commas as specified in IEEE 488.2-1897. The number of significant digits that are returned is automatically set by the Analyzer to ensure the necessary resolution. 9.91E+37 is defined by IEEE 754 to represent "Not a Number." The Analyzer responds with this value when queried for a floating point number it cannot provide. |
| BLOCK | When BLOCK data is specified it is transferred as either INTeger or REAL data. |
| INTeger | Numeric data is transferred in a definite length block as 16-bit signed integers. |
| REAL | Numeric data is transferred in a definite length block as 64-bit IEEE 754 floating point numbers. |
| Literal | Only the short form of the literal parameter is returned. |
| String | A string response is ASCII characters enclosed by double quotes. For example, string data is used for the "<error description string>" portion of error messages and for :SENS:FUNC? response. |
| Numeric Boolean | Only a 0 or a 1 is returned for the status of commands that use Boolean parameters. |
| Definite Length Block | Definite Length Blocks are used when transferring Integer and Real data. Additional information on how to use these blocks is contained in this chapter. |

Transferring Data to the Computer

After receiving a query (command header followed by a question mark), the Analyzer interrogates the requested function and places the answer in the Output Queue. The answer remains in the Output Queue until it is read or another command is issued. When read, the answer is transmitted over the HP-IB to the computer. For example, to read the result of the query :DISPlay:TRACe:CONNect? you would execute the program statement:

Data Transfer Overview

ENTER 712;Setting\$

This statement enters the current setting for the :DISPlay:TRACe:CONNect in the string variable Setting\$. In this case, a 1 is returned if the screen data points are connected and a 0 if they are not connected.

All results for queries sent in a program must be read before another program message is sent. For example, when you send the query :DISPlay:TRACe:CONNect?, you must follow that query with the program statement ENTER 712;Setting\$ to read the results of the query and place the results in a variable (Setting\$).

Sending another command before reading the result of the query causes the output buffer to be cleared and the current response to be lost. This also causes an error message to be placed in the error queue.

Requesting a response before sending a query can cause the computer to wait indefinitely unless a time out has been set up.

The format specification for handling the response message is dependent on both the computer and the programming language.

Multiple Queries

You can send multiple queries to the Analyzer within a single program message, but you must also read them back as a single response message. This can be accomplished by either reading them back into a string variable or into multiple numeric variables. For example, you can read the result of the query:

OUTPUT 712; ":FREQuency:ARM:ECOunt?;LEVel?"

into the string variable Setting\$ with the program statement:

ENTER 712;Setting\$

Commands Used to Transfer Data

When you read the result of multiple queries into string variables, each response is separated by a semicolon. For example, the response to this query is:

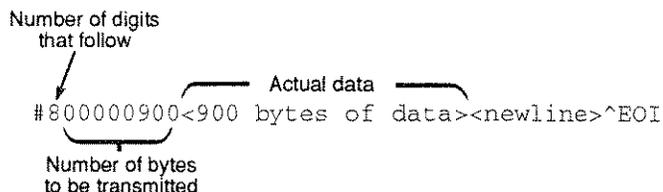
```
<edge_count>;<level>
```

To read the results into numeric values use the program statement:

```
ENTER 712;Setting1,Setting2
```

Definite-Length Block Response Data

Definite-length block response data allows any type of device-dependent data to be transmitted over the HP-IB interface as a series of 8-bit binary data bytes. This is particularly useful for sending large quantities of data or 8-bit extended ASCII codes. The syntax is a pound sign (#) followed by a non-zero digit representing the number of digits in the decimal integer. After the non-zero digit is the decimal integer that specifies the number of 8-bit data bytes being sent. This is followed by the actual data. The



terminator is a linefeed asserted with EOI.

For example, for transmitting 900 bytes of data, the syntax is:

The “8” states the number of digits that follow and the eight digits “00000900” state the number of data bytes to be transmitted.

Commands Used to Transfer Data

All Common Command and Subsystem queries transfer data to the computer. In most cases this is specific information on a single result or setting. How to transfer single results or strings was described previously.

Commands Used to Transfer Measured Data Values

Commands Used to Transfer Measured Data Values

The following Subsystems are used to format, describe and place large quantities of histogram or vs Time data in the Output Queue.

- :FORMat
- :CALCulate
- :SENSe
- :FETCh
- :MEASure
- :READ

How to transfer measured data values using these subsystems is described in the following paragraphs.

- :SENS:DATA?
- :CALC:DATA?
- :CALC:ANAL:<analy_fcn>:RES?
- :READ subsystem commands
- :FETC subsystem commands
- :MEAS subsystem commands
- :HIST:SAMP subtree commands
- :CALC:HIST:SAMP subtree commands

All of these commands have the effect of **holding off** further commands from being processed **until** a measurement completes if they are executed while a single measurement is in progress or continuous measurements are being made.

What Each Subsystem Does

The :FORMat subsystem selects the format of the data to be transferred. The selected format is used by all of the other subsystems. If you change format for one subsystem, you have changed it for all subsystems. Your choices are ASCii and BLOCk.

The :CALCulate subsystem performs post-acquisition data processing (in this case, calculates the Histogram from vs Time measurement data) and places the results in the Output Queue.

Commands Used to Transfer Measured Data Values

The :DATA? command in both the :CALCulate and [:SENSe] subsystems is used to transfer the data. The :SENS:DIM[1|2] subtree commands are used to interpret the :SENS:DATA? results (vs Time and Fast Histogram). The :CALC:DIM[1|2] subtree commands are used to interpret the :CALC:DATA? results (Histogram from vs Time).

The :CONFigure, :FETCh, :MEASure, and :READ subsystems are part of the Autoscale Group.

NOTE

It is easier to understand the relationship of the measurement data to the commands used when transferring the data if you use the Autoscale commands (:FETC, :MEAS, or :READ) when transferring data obtained using Autoscale commands and the :CALC:DATA? or :SENS:DATA? when transferring data obtained using :SENSe and :CALC subsystem commands. This is not mandatory, but if you mix and match commands, you need to pay very careful attention to be sure that you understand how to request the desired data.

A typical sequence of data transfer is as follows:

- Create the data using :SENS, :CALC, :INIT, or the Autoscale Group commands.
- Use the :FORMat command to select the data format you want to use.
- Use the appropriate :CALC:DIM or :SENS:DIM queries to obtain information about the data.

Commands Used to Transfer Measured Data Values

- Use this information to determine the format of data (:FORM?), dimension arrays (:SIZE?), and later convert the data (:SCAL? and :OFFS?) to a form that you can use.
- Use either :SENS:DATA?, :CALC:DATA? or the Autoscale Group commands to place the data in the Output Queue.
- Use a program statement like the HP BASIC ENTER statement to transfer the contents of the Output Queue to the computer. If you have used the ASSIGN statement to transfer BLOCK data, be sure to set the FORMAT attribute of the HP BASIC ASSIGN statement to OFF.
- Use the information about offset and scaling to convert the data as required for your use.

All of these commands are described in chapter 2, "Subsystem Commands Reference Dictionary." The following paragraphs explain the relationship between these commands. The examples in appendix D, "Programming Examples" show how these commands are used when transferring different types of data.

Description of Transferred Data

Two different arrays of data are transferred: Histogram or vs Time. Histogram data can be divided into Histogram from vs Time data and Fast Histogram data.

The Histogram data is the actual measured data values stored in memory and is represented as an 8 byte floating point number.

The measurement vs Time data is the pixel value of the measured data displayed on the front panel and is represented as a 2 byte signed integer.

Commands Used to Transfer Measured Data Values

Format of the Data Transferred

You can select one of the following types of data:

- The ASCII format is used to transfer the responses of all Analyzer instrument-state queries. It can also be used to transfer both vs Time and Histogram measurement data. It is the default format for the measurement data transfers.
- The block format can only be used to transfer measurement data. Integer data is automatically selected when transferring vs Time measurement data. Real data is automatically selected when transferring histogram data. Transferring block formatted data is faster than transferring ASCII formatted data.

NOTE

When first learning how to transfer measured data, use the examples contained in appendix D, "Programming Examples," plus the summary of :SENS:DIM[1\2] in table 5-3, :CALC:DIM[1\2] in table 5-4, and figures 5-5 and 5-6 to make data transfer easier to understand. Appendix D contains examples written in both HP BASIC and Microsoft QuickBASIC.

When a query for measurement data fails (for example, error -230) in block format, a definite length block with 0 measurement data bytes is returned. When a query for measurement data fails in ASCII format, no response is created. When a query for an analysis result fails (error +2011), 9.91E+37 is returned indicating that the Analysis has no analysis results.

Commands Used to Transfer Measured Data Values

How to Interpret Transferred Data

Use :SENS:DIM[1|2] subtree queries to interpret :SENS:DATA? (sensed data, vs Time or Fast Histogram) and :MEASure data. Use :CALC:DIM[1|2] subtree queries to interpret :CALC:DATA? (post-processed Histogram from vs Time) data. When :READ or :FETCh is used for vs Time or Fast Histogram data, use :SENS:DIM queries to interpret the data. When :READ or :FETCh is used for Histogram from vs Time data, use :CALC:DIM queries to interpret the data.

Table 5-3 is a summary of the information returned when using the :SENS:DIM[1|2] subtree queries. The information returned when using :CALC:DIM[1|2] subtree queries are shown in table 5-4. To relate the information in the table to how you use this information when transferring measured data values, each of the returned values is defined.

The :SENS:DATA? and :CALC:DATA? queries, or the :FETCh, :MEASure, and READ subsystem commands place the measured data values in the Output Queue. Before you use any of these commands, you need to have established how to store and interpret the data.

NOTE

In the following discussion, unless otherwise noted, the commands refer to both the :CALC:DIM[1|2] and :SENS:DIM[1|2] subtree commands.

Commands Used to Transfer Measured Data Values

Table 5-3. :SENS:DIM[1|2] Subtree Commands

| | vs Time | | Fast Histogram | |
|-----------|--|---|--|--|
| | DIM1 (Y-axis) | DIM2 (X-axis) | DIM1 (Y-axis) | DIM2 (X-axis) |
| :SENSe | | | | |
| :DIM[1 2] | frequency or time interval | time | frequency or time interval probability | frequency or time interval bins |
| :LABel? | FREQ or TINT | TIME | FPR or TIPR | FBIN or TIB |
| :TYPE? | EXPL | IMPL | EXPL | IMPL |
| :SCALE? | $\frac{\text{freq or ti span}}{\text{YPIXELS} - 1}$ (panorama off) YPIXELS=257 (panorama on) YPIXELS=225 | (panorama off) $\frac{\text{main time span}}{\text{XPIXELS} - 1}$ (panorama on) $\frac{\text{pan time span}}{\text{XPIXELS} - 1}$ where XPIXELS= 450n + 1 n is number of screens acquired n= 1 (pan off) n= 1 to 20 (pan on), no extended memory n= 1 to 80 (pan on), with extended memory | 1.0 | $\frac{\text{freq or ti span}}{\text{number of bins} - 1}$ |
| :OFFSet? | minimum frequency or time interval | (pan off) time at left edge of screen minus DIM2:SCAL (pan on) time at left edge of panorama minus DIM2:SCAL | 0.0 | minimum frequency or time interval minus :DIM2:SCAL |

Commands Used to Transfer Measured Data Values

Table 5-3. :SENS:DIM[1|2] Subtree Commands (Continued)

| | vs Time | | Fast Histogram | |
|--------------------|--|---------------|---|---------------|
| | DIM1 (Y-axis) | DIM2 (X-axis) | DIM1 (Y-axis) | DIM2 (X-axis) |
| :SIZE? | 450n where n is number of screens acquired n= 1 (pan off) n= 1 to 20 (pan on), no extended memory n= 1 to 80 (pan on), with extended memory | Same as DIM1 | number of bins is ≤450 | Same as DIM1 |
| :ENCode | | | | |
| :NOValue? | -1 | N/A | 0.0 | N/A |
| :FORMat? | Block= INT,+16 (2 bytes/meas) ASCII =ASC,+5 | N/A | Block= REAL,+64 (8 bytes/meas) ASCII= ASC,+15 | N/A |
| maximum data value | pan on 224 pan off 256 | N/A | 1.0 | N/A |
| minimum data value | 0 (actual data) -1 (no value) | N/A | 0.0 | N/A |

Commands Used to Transfer Measured Data Values

Table 5-4. :CALC:DIM[1|2] Subtree Commands

| | Histogram From vs Time | |
|--------------------|---|--|
| | DIM1 (Y-axis) | DIM2 (X-axis) |
| :CALCulate | | |
| :DIM[1 2] | frequency or time interval probability | frequency or time interval bins |
| :LABet? | FPR or TIPP | FBIN or TIB |
| :TYPE? | EXPL | IMPL |
| :SCALe? | 1.0 | $\frac{\text{freq or ti span}}{\text{number of bins} - 1}$ |
| :OFFSet? | 0.0 | minimum frequency or time interval minus :DIM2:SCAL |
| :SIZE? | number of bins is: (panorama off)= 257 (panorama on)= 225 | Same as DIM1 |
| :ENCode | | |
| :NOValue? | 0.0 | N/A |
| :FORMat? | Block= REAL,+64 (8 bytes/meas) ASCII= ASC,+15 | N/A |
| maximum data value | 1.0 | N/A |
| minimum data value | 0.0 | N/A |

For vs Time data, the :ENCode:NOValue? query tells you that a -1 is sent if no Y-axis data value (measurement sample) is available for that pixel location as counted across the X-axis (time). *Figure 5-5* shows the sequential transfer of data for each pixel location on the X-axis. When transferring Histogram data, a 0 is used to indicate that no samples fell into the bin at that measurement location (see *figure 5-6*).

The :FORMat? query response indicates the format the Analyzer will use. ASC indicates that ASCII format is active. ASCII can be used with either vs Time or Histogram.

Commands Used to Transfer Measured Data Values

If BLOCK format is active, the :DIM:FORM? query response of INT,+16 indicates that vs Time data is formatted as 2 bytes (representing a 16-bit signed integer) per measured data value is sent.

The :DIM:FORM? query response of REAL,+64 indicates that histogram data is formatted as 8 bytes (representing a 64-bit floating point number) per measured data value is sent.

The LABEL? query retrieves the label of the specified axis. You can use :DIM:LAB? to query the Y-axis label and :DIM2:LAB? to query the X-axis label.

The :SCALE? and :OFFSet? queries are used to obtain the actual measured data value as defined in the following table:

| Dimension | Equation | Where |
|---------------|--|--|
| DIM1 (Y-axis) | $Y = (\text{scale} * D) + \text{offset}$ | D is a value returned by the :DATA? query. |
| DIM2 (X-axis) | $X = (\text{scale} * S) + \text{offset}$ | S = 1 to value returned by the :DIM:SIZE? query. |

The :SIZE? query returns the number of measurement samples in the measurement data that will be placed in the Output Queue when any query that returns measured data is used. This includes :CALC:DATA?, :SENS:DATA?, and any of the applicable queries in the :MEASure, :FETCh, or :READ subsystems.

Commands Used to Transfer Measured Data Values

Figure 5-5 shows the X- and Y-axis for a vs Time measurement. A data value is transferred for each pixel point on the X-axis. If there is a Y-axis measurement sample for a point, that value is sent. If there is no Y-axis measurement sample for a point, -1 is sent (-1 is the "no value" indication for the Analyzer). Therefore, a value (or "no value") measured data point is sent for each pixel location on the X-axis.

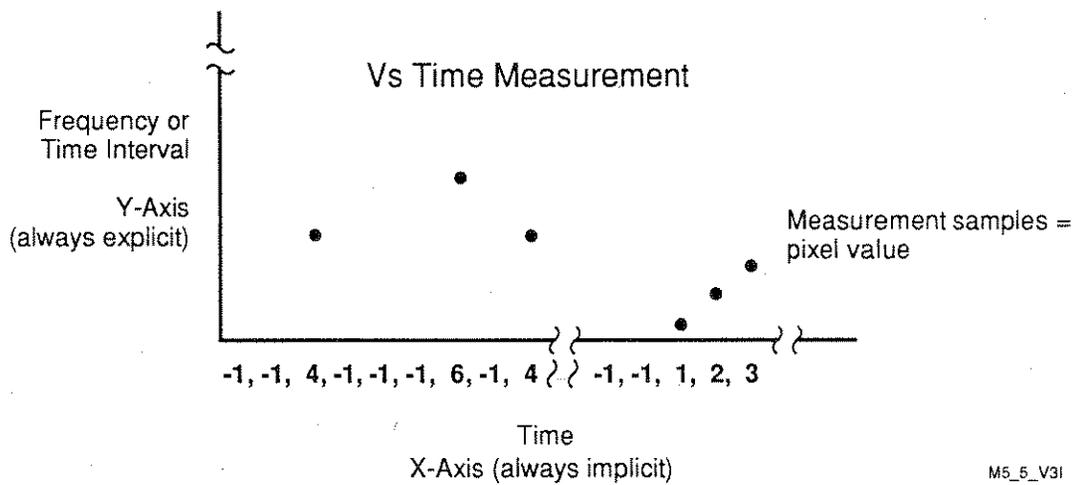


Figure 5-5. X- and Y-axis Description for vs Time

Commands Used to Transfer Measured Data Values

Figure 5-6 shows the X- and Y-axis for a histogram. A data value is transferred for each measurement bin on the X-axis. There is a Y-axis probability value for each bin, and that value is sent. In a histogram, a value of 0 is significant as it indicates no measurements hit that bin. Therefore, there are no bins for which a "no value" indication is sent. The probability values in the histogram range from 0.0 to 1.0 (to achieve the corresponding percentage value, multiply by 100).

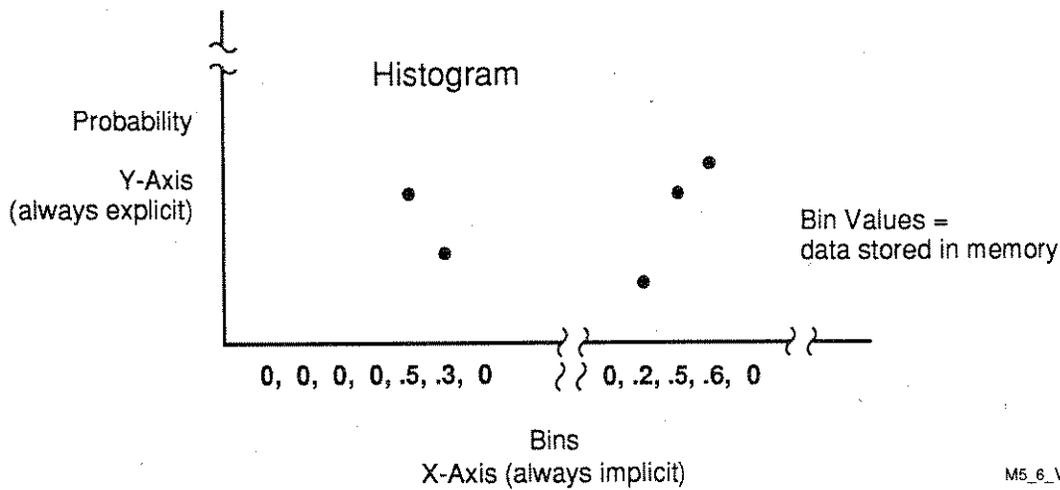


Figure 5-6. X- and Y-axis Description for Histogram

Commands Used to Transfer Measured Data Values

A

HP-IB INTERCONNECTION

In This Appendix

This appendix describes how to connect the Analyzer to a computer and a printer. It also contains procedures for setting the Analyzer's HP-IB address and operating mode.

NOTE

For the Analyzer to respond to your programs, the correct HP-IB address must be entered and the addressed (talk\listen) mode must be selected before you run the program.

Connecting to a Computer

Connect the Analyzer to a computer by simply installing an HP-IB cable (such as an HP 10833A cable) between the two units (*Figure A-1*).

Connecting to a Computer

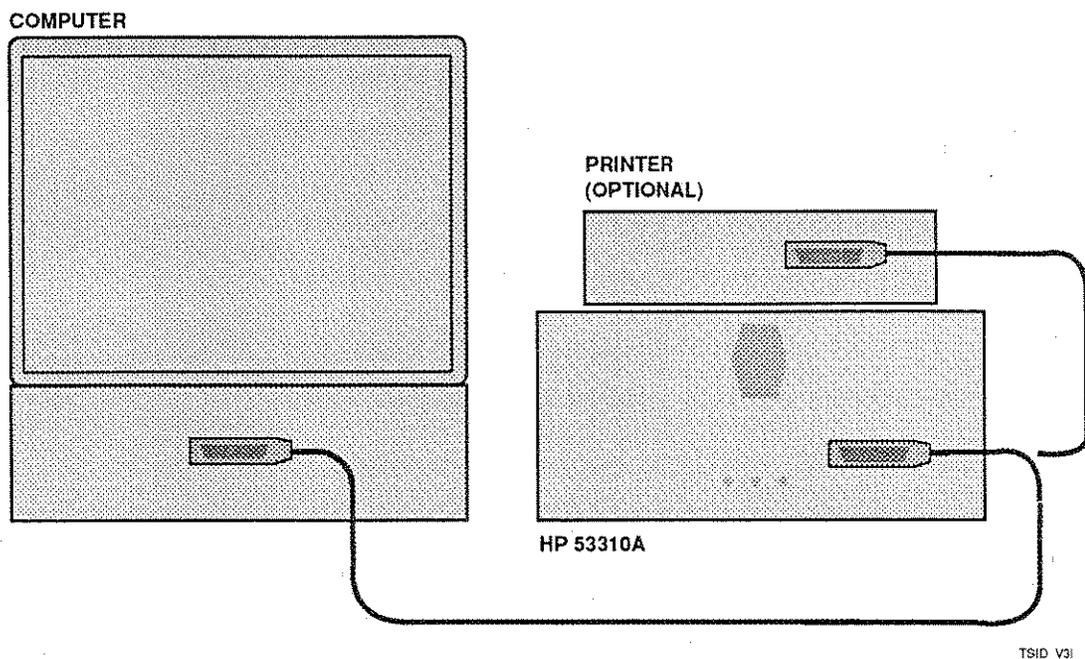


Figure A-1. HP-IB Interconnection Diagram

Figure A-2, provides interconnection data about the rear panel HP-IB connector on the Analyzer. This connector is compatible with the HP 10833A/B/C/D cables. Up to 15 HP-IB compatible instruments (including the controller) can be interconnected in a system. The HP-IB cables have identical "piggy-back" connectors on both ends so that several cables can be connected to a single source without special adapters or switch boxes. System components and devices may be connected in virtually any configuration desired. There must, of course, be a path from the controller to every device operating on the bus.

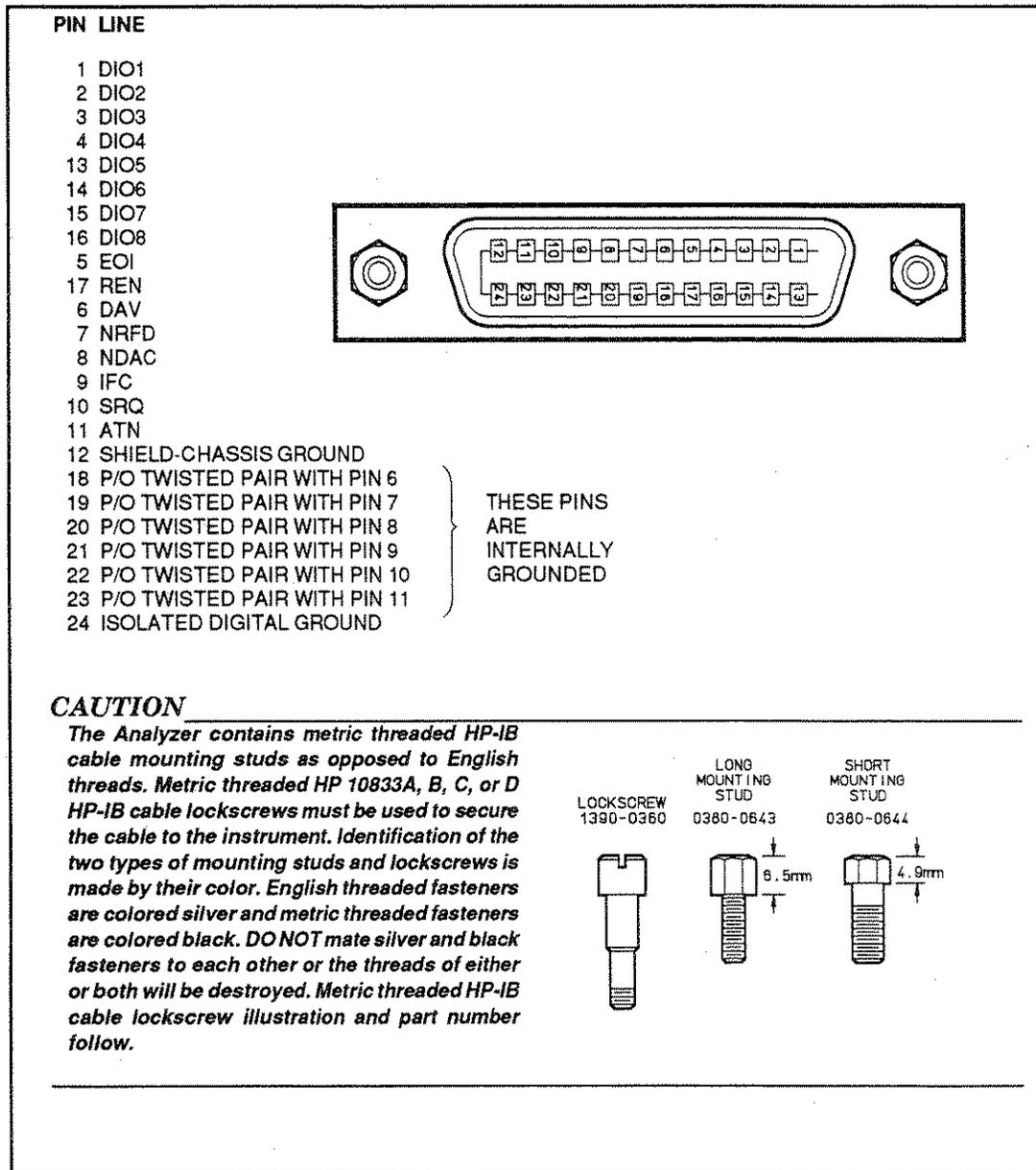


Figure A-2. Hewlett-Packard Interface Bus Connection

Connecting to a Computer

Logic Levels

The Hewlett-Packard Interface Bus logic levels are TTL compatible, i.e., the true (1) state is 0.0V dc to 0.4V dc and the false (0) state is +2.5V dc to +5.0V dc.

Mating Connector

HP 1251-0293; Amphenol 57-30240.

Mating Cables Available

HP 10631A, 1 metre (3.3 ft.), HP 10631B, 2 metres (6.6 ft.),
HP 10631C, 4 metres (13.2 ft.), HP 10631D, 1/2 metre (1.6 ft.).

Cabling Restrictions

1. A Hewlett-Packard Interface Bus System may contain no more than 2 metres (6.6 ft.) of connecting cable per instrument.
2. The maximum accumulative length of connecting cable for any Hewlett-Packard Interface Bus System is 20.0 metres (65.6 ft.)
3. The maximum number of instruments in one system is fifteen.

Figure A-2. Hewlett-Packard Interface Bus Connection (Continued)

CAUTION

AVOID stacking more than three or four cables on any one connector. Multiple connectors produce leverage that can damage a connector mounting. Using a screwdriver can damage the threads inside the head of the lock screw. Instead of using a screwdriver, finger-tighten connector lock screws so they cannot come loose during operation.

Connecting to a Printer

You can connect the Analyzer to a printer either with or without a computer. If you connect a computer, you can operate the printer under program control. In the absence of a computer, the Analyzer, when in Talk Only, becomes the controller for the printer.

When operating a printer without a computer, set the Analyzer HP-IB mode to Talk Only. When operating in Talk Only mode, the printer must be in Listen Only or Listen Always mode, and may be set to output a service request (SRQ). Refer to the respective printer manual for these settings.

The Using the :SYSTEM:PRINT? Command example in appendix D, "Programming Examples" controls the printing feature of the Analyzer (when using a computer). This program takes advantage of the Operation Status Register and Service Request Enable (SRE) register to determine when hardcopy is complete.

Operating Modes

The Analyzer has two HP-IB operating modes:

Addressed (Talk/Listen) — This mode is for bi-directional communication. The Analyzer can receive commands and setups from the computer, and can send data and measurement results.

Setting the HP-IB Operating Mode

- **Talk Only** — In this mode, the Analyzer can send data to a printer. It cannot receive commands or setups from the computer.

Setting the HP-IB Operating Mode

Perform the following steps to set the HP-IB operating mode:

1. Press the **Utility** key to display the **Utility Menu**.
2. Press the **HP-IB/Print Menu** softkey to display the **HP-IB/PRINT** menu.
3. Press the **Addressed/Talk Only** softkey to highlight **Addressed**.

Setting the HP-IB Address

Perform the following steps to set the HP-IB address:

1. Press the **Utility** key to display the **Utility Menu**.
2. Press the **HP-IB/Print Menu** softkey to display the **HP-IB/PRINT** menu.
3. Press the **HP-IB Address** softkey to highlight it.
4. Enter **12** (or any valid address) on the **Entry** keys and verify **12** (or the value entered) is displayed.

B

HP-IB GENERAL INFORMATION

In This Appendix

This appendix contains programming information for remotely operating the Analyzer through the Hewlett-Packard Interface Bus (HP-IB) using the SCPI defined command set. The information in this chapter includes descriptions of:

- The HP-IB.
- The Analyzer interface capabilities.
- Remote operation characteristics.

NOTE

All the Analyzer SCPI programming commands are described in chapters 2 and 3. Command entries include one or more program lines. Appendix D has expanded program examples that demonstrate in greater detail some of the programming features of the Analyzer.

The Analyzer operates remotely via the HP-IB. All menu functions and front panel controls (excluding the power switch) are programmable through the HP-IB. At the simplest level, the Analyzer can transmit data in the Talk Only mode to output devices such as a printer. In more sophisticated systems, a computer can direct the Analyzer to perform a specific measurement, process the resulting measurement data, and transfer the results.

Most Analyzer remote programming operations are extensions of the front panel functions. Other operations, such as data transfer and instrument identification, are only available in remote operation.

HP-IB Description

You can use any computer that is compatible with the HP-IB standard. Most program examples given in this manual are written assuming an HP Series 9000 as the computer. Refer to Appendix A for information about connecting a computer to the Analyzer.

HP-IB Description

The HP-IB is a parallel interface bus. All devices on the bus can be addressed at one time, however, only one device may respond at a time. The computer commands a specific device to respond, and maintains the data flow and interface functions. Data is transferred at the speed of the slowest device.

Party Line Structure Explained

The HP-IB uses a party-line structure (devices share signal lines). A maximum of 15 devices may be connected in an HP-IB system, in virtually any configuration desired. There must be an uninterrupted path to every device operating on the bus. Sixteen lines interconnect devices in parallel. Eight lines transfer data and commands, three accomplish handshaking, and five lines manage the bus.

Interface System Terms

The following terms and concepts apply to HP-IB operations.

Address: Each device on the interface is assigned an address. The address specifies which device receives or sends information.

Byte: A byte is a unit of information consisting of eight binary digits (bits).

Device: Any instrument or unit that is HP-IB compatible is defined as a device.

Device-Independent Command: A command predefined by the interface standard to have a specified bit pattern and resulting action.

Device-Dependent Command: A command that is specific to a particular instrument or family of instruments, which is not predefined by the interface

HP-IB Description

standard. Device-dependent commands are usually sent as ASCII strings of characters.

Polling: Polling is a process typically used by a computer to locate a device that requires service from the computer. There are two types of polling, Serial Poll and Parallel Poll.

Serial Poll: When the computer executes a serial poll, the addressed device sends one byte of operational information called a status byte. If more than one device on the interface is capable of requesting service, each device on the interface must be polled individually until the device that requested service is located.

Parallel Poll: Parallel poll is a computer-initiated operation to obtain information from several devices simultaneously. The Analyzer does not have parallel poll capability.

Major Interface Functions

Each device on the interface bus may have one or more of the following major device capabilities: **Controller, Talker, or Listener.**

The Controller manages interface activity, and must have the applicable interface module. Controllers transmit all device-independent commands to other devices in the interface and usually have Talker and Listener capabilities. Only one device on the interface may be the active controller at any one time. The Analyzer has no controller capabilities.

Talkers are devices that can send data or device-dependent commands through the interface. Note that a talker will not actually send data or information until the appropriate command is sent by the computer. The Analyzer has Talker capabilities. When the instrument is talking on the interface, or is addressed to talk, the "rem tlk" message is displayed on the Analyzer's front-panel display.

In special situations, a device may be classified as a Talk Only device. A Talk Only device sends information to Listen Always devices or any devices addressed to listen. Such a system has no controller. For example, the

Analyzer Interface Capabilities

Analyzer can be configured to Talk Only and send display screens to a printer (configured as Listen Always) by setting the instrument to the Talk Only mode (the "tlk only" message is displayed on the Analyzer's front-panel display) and pressing the Print hardkey.

Listeners can receive information over the interface. When the Analyzer is listening, or addressed to listen, the "rem lsn" message is displayed on the Analyzer's front-panel display. Listeners must also be enabled by the Controller to receive data or information.

Analyzer Interface Capabilities

The capabilities of a device connected to the bus are specified by the list of interface functions supported. These functions provide a way for a device to receive, process, and send messages over the bus.

Table B-1 lists the HP-IB interface functions defined by the IEEE 488.1-1987 standard, including the name, mnemonic, and a brief description. Also listed in the table are the subset identifiers (the interface function mnemonic followed by a number). These identifiers indicate the specific HP-IB interface function capabilities of the Analyzer.

Table B-1. HP-IB Interface Functions

| Name and Mnemonic | General Description | Subset Identifier | Specific Analyzer Capability |
|-------------------------|--|-------------------|--|
| Source Handshake (SH) | Capability to properly translate a multiline message. | SH1 | The Analyzer can generate messages. |
| Acceptor Handshake (AH) | Capability to guarantee proper reception of remote multiline messages. | AH1 | The Analyzer can interpret received messages. |
| Talker (T) | Capability to transmit data over the bus when addressed. | T5 | The Analyzer can function as a talker. In addition, it can operate as a Talk Only instrument and will respond to serial poll. It will unlisten if addressed as a talker. |

Analyzer Interface Capabilities

Table B-1. HP-IB Interface Functions (Continued)

| Name and Mnemonic | General Description | Subset Identifier | Specific Analyzer Capability |
|--------------------------|---|--------------------------|--|
| Extended Talker (TE) | Talker capability with address extension. | TE0 | The Analyzer cannot function as an extended talker. |
| Listener (L) | Capability to receive data over the bus when addressed. | L4 | The Analyzer can function as a listener. In addition, it will untalk if addressed as a listener. |
| Extended Listener (LE) | Listener capability with address extension. | LE0 | The Analyzer cannot function as an extended listener. |
| Service Request (SR) | Capability permitting a device to asynchronously request service from the controller. | SR1 | The Analyzer can generate a service request. |
| Remote/Local (RL) | Capability to select between two sources of input information: local (front panel controls) and remote (input information from the bus). | RL1 | The Analyzer can operate both in remote and local modes. In addition, it can respond to local lockout. |
| Parallel Poll (PP) | Provides capability for a device to uniquely identify itself if it requires service and the controller is requesting a response. This capability differs from service request in that it requires a commitment of the controller to periodically conduct a parallel poll. | PP0 | The Analyzer does not support parallel poll. |
| Device Clear (DC) | This function allows a device to be initialized to a predefined state. | DC1 | The Analyzer supports both the Device Clear (DCL) and Selected Device Clear (SDC) commands. |

Analyzer Interface Capabilities

Table B-1. HP-IB Interface Functions (Continued)

| Name and Mnemonic | General Description | Subset Identifier | Specific Analyzer Capability |
|---------------------|--|-------------------|---|
| Device Trigger (DT) | This function permits a device to have its basic operation initiated by the talker on the bus. | DT1 | The Analyzer can be remotely triggered. |
| Controller (C) | This function permits a device to send addresses, universal commands, and addressed commands to other devices on the HP-IB. It may also include the ability to conduct polling to determine devices requiring service. | C0 | The Analyzer cannot function as a controller. |
| Drivers (E) | This code describes the type of electrical drivers used in a device. | E2 | The Analyzer has three-state drivers. |

Via the HP-IB, you can remotely program most controls (except Power On/Off). You also receive measurement data via the HP-IB. The Analyzer operates as both a talker and a listener, as described in *Table B-1*.

TALK: The Analyzer can be addressed to Talk by a computer or by setting the instrument to the Talk Only mode. When addressed as a Talker, the instrument sends data to other devices on the bus. This data may result from a measurement, an error condition, a diagnostic test, or other operations.

LISTEN: When addressed as a Listener, the instrument accepts any number of commands from a computer on the bus. These commands are used to program the instrument operation.

Analyzer Interface Capabilities

| | |
|-------------------------------------|--|
| SERVICE REQUEST: | A Service Request (SRQ) is generated on the interface when an enabled status bit is set. The Analyzer can asynchronously request service from the computer that is in charge of the bus. |
| REMOTE/ LOCAL: | At power-up, the instrument is under front panel (local) control. When the Analyzer receives a programming command and it is in addressed mode, it switches to Remote. Once in Remote, programmable functions cannot be affected by the front panel controls. The LOCAL key may be used to manually return to local control only if the Local Lockout (LLO) is off. If Lockout is on, the LOCAL key is ignored, and the bus command LOCAL 7 must be sent to disable LLO and return to local control. |
| PARALLEL POLL: | The Analyzer does not respond to parallel poll. |
| DEVICE CLEAR: | When a universal or selected Device Clear is received, the Analyzer clears the input buffer and Output Queue, resets the parser, clears any command that would prevent processing a *RST or other commands, disables the effect of a prior *OPC command, and terminates the holdoff action of a *WAI or data query waiting for a measurement to complete. |
| DEVICE TRIGGER: | When a Device Trigger is received, the instrument initiates a single measurement. |
| CONTROLLER: | The Analyzer cannot be used as a controller. |
| Front Panel Interface Status | The remote status of the Analyzer is indicated by messages on the front-panel display. |

HP-IB Address Selection

To use the Analyzer in an HP-IB system, you must set the desired HP-IB address. The address is used by the computer to identify the Analyzer. The range of permitted addresses is "0" to "30", inclusive. The address is saved when the Analyzer is off or unplugged. If the saved address cannot be recalled due to memory or battery failure, a default address of "12" is used.

When a computer is installed, the Analyzer functions as a talker and a listener. In the absence of a computer, you can set the Analyzer to Talk Only mode. In this way you can send display screens to a printer. In Talk Only mode, the Analyzer functions only in an output condition. The receiving device must be set to Listen Always.

Interface Commands

The Analyzer recognizes two classes of commands: device-independent commands and device-dependent commands. Device-independent commands are defined by the interface standard document and are the same for all instruments. Device-dependent commands are unique to the instrument and are defined by the instrument designer.

The device-independent commands are described in the following paragraphs. Device-dependent commands are described in chapter 2, "Subsystems Command Reference Dictionary" and chapter 3, "Common Command Reference Dictionary."

Device-Independent Commands

Device-independent commands are identified by three-letter mnemonics such as GTL, which represents Go To Local. Device-independent commands are sent as encoded bytes on the interface bus and not as ASCII strings, thus these commands cannot be sent from the computer using the OUTPUT statement. Many computers, however, do incorporate a command of the form SEND#;CMD#, where # is a decimal number.

Table B-2 lists the mnemonic, full name, and a brief description of the instrument response for each supported device-independent command.

Interface Commands

Table B-2. Device-Independent Commands

| Mnemonic | Command Name | Description |
|----------|-----------------------|--|
| ATN | Attention | Alerts the instrument of each device independent message being sent, so the instrument is ready to accept data and interpret them as commands. |
| DCL | Device Clear | This command clears all errors, aborts all partially completed commands and pending send data commands, and clears all input and output buffers. |
| EOI | End Or Identify | If ATN is false and the instrument is a listener, EOI acts as a message delimiter, and indicates the last data byte of a multibyte sequence. |
| GET | Group Execute Trigger | If the instrument is addressed to listen, GET aborts the current measurement, and triggers the next measurement immediately. |
| GTL | Go To Local | If the instrument is addressed to listen, GTL returns the instrument to front panel (local) operation. Local Lockout is not cleared. |
| IFC | Interface Clear | The instrument untalks and unlistens, and the interface initializes to an idle state (no activity on the bus). |
| LAD# | Listen Address n | If n matches the instrument address, the instrument becomes a listener. |
| LLO | Local Lockout | The front panel LOCAL key is disabled if the instrument is in remote mode. All front panel key presses are ignored. |
| MLA | My Listen Address | MLA is the listen address (LAD#) that matches the instrument address. |

Interface Commands

Table B-2. Device-Independent Commands (Continued)

| Mnemonic | Command Name | Description |
|----------|-----------------------|---|
| MTA | My Talk Address | MTA is the talker address (TAD#) that matches the instrument address. |
| NRE | Not Remote Enable | The instrument returns to front panel (local) operation; Local Lockout is cleared. |
| NUL | Null | No effect when received by the instrument. |
| REN | Remote Enable | The instrument enters the remote state, and is enabled to respond to interface commands when addressed as a listener. |
| SDC | Selected Device Clear | If the instrument is a listener, will cause the same response as DCL. |
| SPD | Serial Poll Disable | Terminates serial polling, and returns the instrument to a normal talker state to output device dependent data rather than status information. |
| SPE | Serial Poll Enable | Establishes serial polling, and enables the instrument to send the serial poll status byte when addressed to talk. |
| TAD# | Talk Address n | If n matches the instrument address, the instrument becomes a talker. |
| UNL | Unlisten | The instrument is unaddressed and terminates listening. A single device cannot be unaddressed without unaddressing all listeners. |
| UNT | Untalk | Unaddresses the instrument, if currently a talker, and terminates talking. Addressing another talker on the interface automatically unaddresses any current talker. |

Meta Messages

To simplify using the HP-IB interface, Hewlett-Packard developed the Meta Message concept, shown in *Table B-12*. Rather than require you to remember all the device-independent messages and their interactions, useful command sequences are integrated into a corresponding single command, on many Hewlett-Packard computers. For example, to clear the instrument at address three using the device-independent commands, you must send the sequence ATN, UNL, MTA, LAD 12, SDC. The HP BASIC command "CLEAR 712" sends the same sequence with no further user interaction required, thus greatly simplifying the programming interface.

Many messages implemented in HP BASIC may be sent in either of two forms, with addressing and without addressing. The form with addressing normally addresses a particular device to listen. For example, the command "REMOTE 7" sends REN without making any device a listener, while the command "REMOTE 712" sends REN and makes the device at address 12 a listener.

NOTE

Care must be taken when using Meta messages. If a Meta message is issued shortly after sending another command, there is a possibility that the Meta message will prevent the completion of the previously issued command. If problems are encountered, WAIT commands should be used to allow the command sufficient time to complete execution.

Through meta messages, bus devices can exchange control and measurement information. The command sequences are typical in that different computers may send different sequences for a given meta message, but will produce the same result.

Interface Commands

Table B-3 Meta Messages

| Meta Message | Command Sequence | General Description | Specific Analyzer Response |
|---------------------------|----------------------|--|--|
| DATA | UNL, MTA, LAD#, data | Transfers device dependent information from one device to one or more devices on the bus. | The Analyzer sends measurement data as defined by the device-dependent command received from the controller. |
| TRIGGER | UNL, MTA, LAD#, GET | Causes a group of selected devices to simultaneously initiate a set of device-dependent actions. | Starts a new measurement. |
| CLEAR | UNL, MTA, LAD#, SDC | Causes the instrument to be set to a predefined state, such as a certain range or function. | Causes the Analyzer to clear all input and output buffers. |
| REMOTE | REN, UNL, MTA, LAD# | Permits selected devices to be set to remote operation, allowing parameters and device characteristics to be controlled by bus messages. | Causes the Analyzer to go to remote operation if REN is true, and if instrument is addressed to listen. Locks out all front panel keys except LOCAL; instrument is controlled by bus messages. |
| LOCAL | UNL, MTA, LAD#, GTL | Causes selected devices to return to local (front panel) operation. | Returns the Analyzer to front panel control. Instrument status is that set just prior to receipt of the Local message. |
| LOCAL LOCKOUT | LLO | Disables local (front panel) controls of selected devices. | Disables LOCAL key. The Analyzer remains in remote operation until a Local message is received on the bus. |
| LOCAL/CLEAR LOCAL LOCKOUT | LCLL | Returns all devices to local (front panel) control and simultaneously clears the Local Lockout message. | Returns Analyzer to local control and clears Local Lockout. |

Interface Commands

Table B-3 Meta Messages (Continued)

| Meta Message | Command Sequence | General Description | Specific Analyzer Response |
|-----------------|-------------------------------------|--|---|
| SERVICE REQUEST | SRQ | Indicates a device requires service by the controller. Service requests occur when specific hardware- related conditions occur such as a time base oscillator out of lock, an input pod removed, and so on. | The Analyzer sends a Service Request message to the controller under certain conditions, as defined by the settings of the Status Data Structure. This message is ignored by the Analyzer when received. |
| STATUS BYTE | UNL, MLA, TAD#, SPE, data, SPD, UNT | Presents status information of a particular device; one bit indicates whether or not the device currently requires service, the other seven bits (optional) are used to indicate the type of service required. | The Analyzer sends status information to the controller. |
| STATUS BIT | Not applicable | A single bit of device-dependent status information which may be logically combined with status bit information from other devices on the controller. | The Analyzer does not use this message. |
| PASS CONTROL | Not applicable | Passes bus controller responsibilities from the current controller to a device which can assume the bus supervisory role. | The Analyzer does not use this message. |

Interface Commands

Table B-3 Meta Messages (Continued)

| Meta Message | Command Sequence | General Description | Specific Analyzer Response |
|--------------|------------------|---|--|
| ABORT | IFC | Unconditionally terminates bus communications and returns control to the system controller. | <p>All HP-IB activity terminated and control returns to the system controller. Talk and Listen are cleared for the Analyzer and all other devices on the bus, which terminates all bus communications. The Analyzer status remains as it was just prior to receipt of the Abort message. Any partially entered HP-IB data message is aborted.</p> <p>The ABORT Meta message is distinct from the Analyzer ABORT Command.</p> |

Remote Operation

When the Analyzer operates in Remote mode, all front panel controls are disabled except the power switch and the LOCAL key (if Local Lockout is not active). Conditions for changing from Remote to Local, and the operational status of each mode are described in following paragraphs.

Local Mode

In Local mode, all front panel controls are operational, and the instrument will respond to input commands over the bus. If addressed to talk, the Analyzer sends data messages or the status byte. Whether specifically addressed or not, the instrument responds to:

- Remote
- Local
- Local Lockout
- LCLL
- Trigger
- Abort

The Analyzer can send a require-service message when in Local mode.

The Analyzer always switches from Remote to Local when a GTL or LCLL message is received. The LCLL message sets the remote enable control line (REN) false. If the Analyzer is in Local Lockout, the LOCAL key on the front panel is disabled. Instrument settings remain unchanged during Remote to Local transition.

Switching from Local to Remote Operation

The Analyzer switches from Local to Remote when it is addressed mode and one of the following conditions occur:

- The listen address is received, while the REN (Remote Enable) control line is true.
- The REMOTE programming command is received (use the LOCAL command to return to Local when using the REMOTE command).

Remote Operation

The instrument settings remain unchanged during Local-to-Remote transitions. The front panel "rmt" message is displayed while the Analyzer is in Remote mode.

HP-IB Default Startup Conditions

To establish a generally preferred set of conditions, default HP-IB states are established during power-up. The Analyzer address and addressing mode are saved in non-volatile memory.

HP-IB default conditions are:

- HP-IB Local mode active
- Local Lockout cleared
- Unaddressed (if in normal addressing mode)
- RQS bit in the status byte register is set to 0
- All event registers, the Standard Event Status Enable Register, Service Request Enable Register, and the status Byte Register are cleared.
- All Status negative and positive registers and enable register are preset to the values described in the :STATus:PRESet command described in chapter 2.

Switching from Remote to Local Operation

Under certain predefined conditions the Analyzer switches from Remote mode to Local mode. These conditions arise via commands sent from the computer or from internal changes.

Local Lockout

If the Analyzer is under Remote (program) control and the front panel LOCAL key is inadvertently pressed, the instrument returns to Local control. Accordingly, data or settings could change. To prevent undesired front panel control, use the Local Lockout message. This message allows return-to-local only under program control.

Remote Operation

You can cycle the power switch to return to local control; but, this technique has serious disadvantages (such as loss of data). Such action may make the system computer lose control of the instrument. During such a cycle, several HP-IB conditions are reset to the default state at power up, and the status of any operation that was in progress is unknown by the computer. It is possible that this could stop activity on the system bus as well.

Required Conditions

The Analyzer switches from Remote to Local mode if one of the following conditions occurs:

- The GTL (Go To Local) interface command is received.
- The REN (Remote Enable) line goes false, in other words the LCLL meta-message is received.
- The LOCAL programming command is received. (Use this command when the REMOTE command is used to go to Remote.)
- An internal return-to-local message goes active (generated by the front panel LOCAL key, but only if Local Lockout is inactive).

The Analyzer settings remain unchanged during remote-to-local transitions. The front panel "rmt" is not displayed while the Analyzer is in the Local mode.

Addressing

When the "addressed (talk/listen)" HP-IB mode is selected from the front panel, the instrument may be addressed to talk or listen using the bus commands. When so addressed the instrument remains configured until you send an abort message (IFC), send a listen address matching the listen address of the instrument, or send a universal untalk command (UNT).

Remote Operation

The instrument is shipped from the factory in the addressable mode, with the talk and listen addresses set to 12. You can set the Analyzer to Talk Only. Talk Only mode enables limited bus operation without an HP-IB computer connected. You can display or change the addresses from the front panel.

When set to talk-only mode, the instrument does not respond to any bus message. Select this mode if you want to directly output data to a plotter or printer without using a computer.

Receiving the Data Message

The instrument responds to data messages when set to Remote mode and addressed to listen.

Sending Data Messages

The instrument sends data messages when set to Local or Remote modes, when addressed to talk, or when in the Talk Only mode. Measurement results will be available for reading when the Analyzer is queried in an addressed mode.

Receiving the Device Clear Message

In response to either the Device Clear message or the Selected Device Clear message, the Analyzer clears the input buffer and Output Queue, resets the parser, clears any command that would prevent processing a *RST or other commands, disables the effect of a prior *OPC command, and terminates the holdoff action of a *WAI or data query waiting for a measurement to complete.

Receiving the Group Execute Trigger Message

The Analyzer responds to the Group Execute Trigger (GET) command by initiating a measurement. That is, the instrument responds exactly as if it had received the *TRG command or :INIT:IMM to initiate a measurement.

Data Messages

The Analyzer communicates on the HP-IB primarily with data messages. The instrument interprets a byte on the eight data lines as a data message when the bus is in the data mode. The bus is in the data mode when the Attention (ATN) control line is false. If the ATN control line is true, the bus is in the command mode.

In the data mode, the Analyzer can receive and send data messages. Input data messages include the program front panel functions and remote functions. Output data messages include instrument status information the setting of specific functions, and measurement results.

Receiving the Abort Message

The ABORT message (IFC control line true) halts all bus activity. When the instrument receives the abort message, it becomes unaddressed and stops talking or listening. The Request Service message (RQS) and the Status Byte are unaffected by the abort message.

Receiving the Abort Message



SCPI CONCEPTS

In This Appendix

This appendix provides information about the Standard Commands for Programmable Instruments (SCPI) command set as it applies to the Analyzer. It is intended to supplement (not replace) the SCPI documentation listed in appendix E "Related Documentation." In some cases, small portions of the standard SCPI documentation are repeated for your convenience. In other cases, information is provided to show which technique (of several possibilities allowed by SCPI) is used in the Analyzer. When this information is contained in other chapters in this manual, you are directed to the applicable chapter.

NOTE

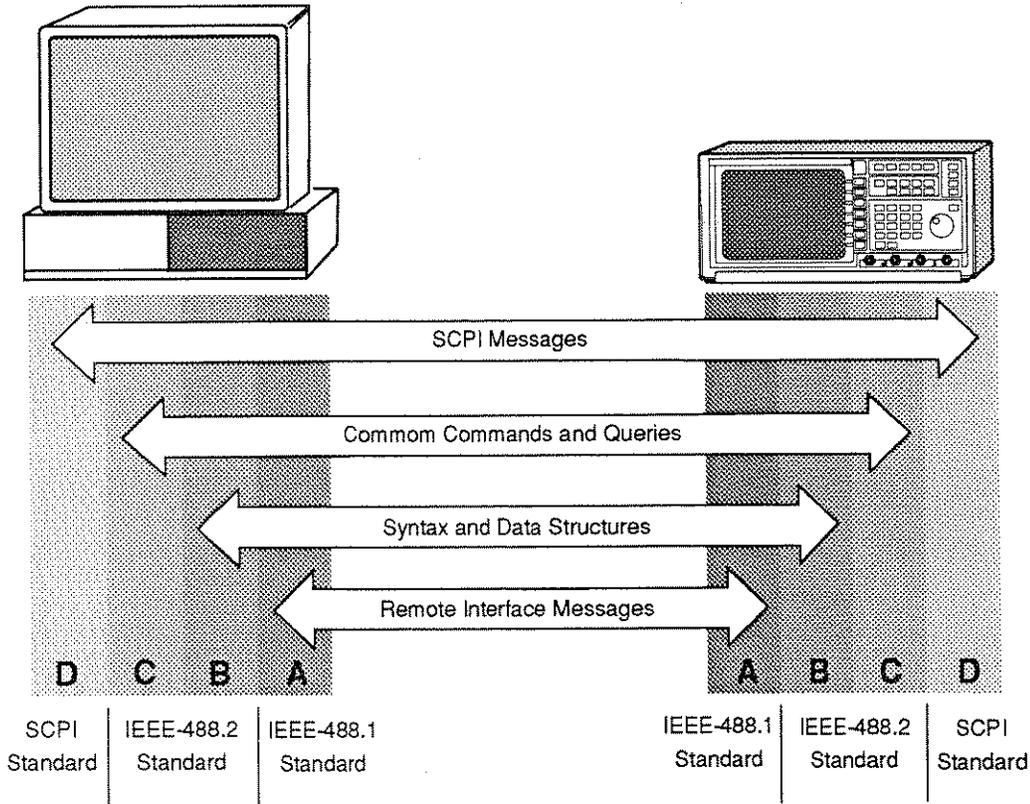
When reading any SCPI documentation, keep in mind that the Analyzer is a SENSE instrument and not a SOURCE instrument.

SCPI Overview

The SCPI standard defines how you communicate with an instrument from a computer. It is a command set designed specifically for electronic test and measurement instruments. Each instrument implements only the portion of the SCPI command set that is needed to meet its operational objectives.

In order to build on existing standards and practices, SCPI is structured in layered standards as shown in *figure C-1*. You can think of each layer of standards as answering a set of questions.

SCPI Overview



MSBS_V31

Figure C-1. Standards Behind SCPI

NOTE

If you are familiar with the Open System Integration (OSI) seven-layer model, the SCPI structure uses a similar concept.

SCPI Overview

Layer A (IEEE 488.1-1987 Standard)

The IEEE 488.1 standard answers questions such as the following:

- What is the physical connection between devices?
- What is the electrical connection between devices?
- How is a byte of data transmitted?
- How are devices instructed to talk and listen?

Layer B (IEEE 488.2-1987 Standard)

The IEEE 488.2 standard answers questions such as the following:

- What syntax is used to send data from one device to another?
- How does a device format or interpret data?
- What registers exist inside a device that record its internal state?

Layer C (IEEE 488.2-1987 Standard)

The IEEE 488.2 standard answers questions such as the following:

- What are the commands to do common tasks?
- What commands reset an instrument, read its status registers, or trigger a measurement?

Layer D (SCPI Standard 1.0-1990)

The SCPI standard answers questions such as the following:

- What commands control each programmable function in an instrument?
- What type of parameters do they accept?
- What type of values do they return?
- How should new commands be structured?

Program and Response Messages

Program and Response Messages

To understand how the Analyzer and computer communicate using the SCPI command set, you must understand the concepts of program and response messages (see *figure C-2*).

Program Messages

Program messages are a combination of one or more properly formatted SCPI commands. Program messages always go from a computer to the Analyzer. They are sent to the Analyzer over the Analyzer HP-IB interface as a sequence of ASCII data messages. These messages are transferred using an OUTPUT command that includes the device address, program message, and program message terminator.

Response Messages

Responses are data sent from the Analyzer to a computer in response to a query. (A query is a command followed by a question mark. Queries are used to find out how the Analyzer is currently configured, transfer data from the Analyzer to the computer, and print the current display.)

After receiving a query, the Analyzer interrogates the requested function and places the response in its HP-IB output queue. The output message remains in the queue until it is read or another command is issued. When read, the message is transmitted across the HP-IB to the computer. You read the message by using an ENTER command that includes the device address and an appropriate variable. Use the PRINT statement to display the message. The following example illustrates how to query the Analyzer and display the message:

```
10 OUTPUT 712;"FUNC?"  
20 ENTER 712;A$  
30 PRINT A$  
40 END
```

Program and Response Messages

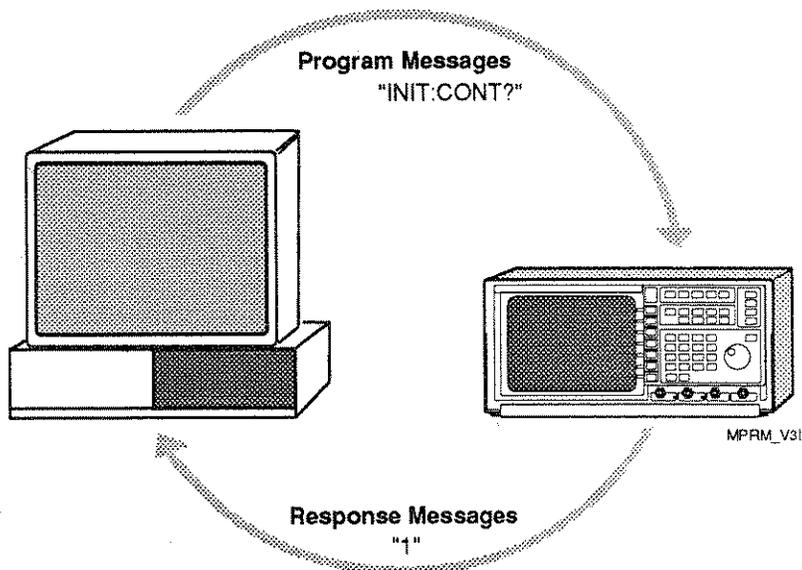


Figure C-2. Program and Reponse Messages

NOTE

The actual OUTPUT and ENTER commands you use when programming are dependent on the programming language you are using.

Command Types

A command is part of a valid message sent from a computer to the Analyzer. It includes the command mnemonic, and any required parameters and punctuation. SCPI commands are divided into two types:

- Common Commands
- Subsystem Commands

Common Commands

The common commands are commands defined by IEEE Standard 488.2 and always begin with an asterisk. They are common to all instruments that conform to IEEE Standard 488.2. These commands generally deal with status reporting. See chapter 3 “Common Commands Reference Dictionary” for a complete description of each command.

Subsystem Commands

Subsystems are groups of commands that are organized into major categories. Subsystem commands are hierarchical. A subsystem command consists of one or more program mnemonics, with a colon separating each mnemonic. The colon specifies that the subsequent command is one level down in the hierarchy. (This is also referred to as a command tree.)

The subsystem commands are described in chapter 2 “Subsystem Command Reference Dictionary.”

Command Parameter Types

Program data is used to convey a variety of parameter information related to the command mnemonic. Parameter types are described in chapter 1 “Programming Overview.”

Typical Measurements

Typical Measurements

Examples of typical measurements are provided in appendix D “Programming Examples” and in the *Operating Reference Manual*.

Status Reporting

A complete description of the status system used in the Analyzer is contained in chapter 4 “Status Reporting.”

Data Transfer

A complete description of how to transfer data between the Analyzer and the computer is contained in chapter 5, Data Transfer.

Arming and Triggering

Arming and Triggering commands are described in the :SENS:FREQ:ARM and :SENS:TINT:ARM subtree commands and in the :TRIGger subsystem commands (for vs Time and Histogram from vs Time) and :SENS:HIST:ARM subsystem commands (for Fast Histogram). Examples of different combinations of arming and triggering are described in the *Operating Reference Manual*.

Arming and Triggering

D

PROGRAM EXAMPLES

In This Appendix

This appendix provides the following sample programs:

- ASCII Transfer of vs Time and Histogram Data (HP BASIC).
- BLOCK Transfer of vs Time Data (HP BASIC).
- BLOCK Transfer of Fast Histogram Data (HP BASIC).
- Using the :STATus:OPERation Register (HP BASIC).
- Using the :SYSTem:PRINt? Command (HP BASIC).
- ASCII Transfer of vs Time and Histogram Data (QuickBASIC).
- BLOCK Transfer of vs Time Data (QuickBASIC).
- BLOCK Transfer of Histogram Data (QuickBASIC).
- Using the :STATus:OPERation Register (QuickBASIC).

Each example contains a brief description of the program and its input requirements.

Overview

Refer to these sample programs for guidance in programming the Analyzer. These program examples are written for two different computers. The first 5 programs are written in HP BASIC for use on the HP 9000 series 200/300 computers. The last 4 programs are written in Microsoft QuickBASIC 4.5. The QuickBASIC programs require an HP 82335A HP-IB interface card in an IBM PC or compatible.

A signal source is required for all examples. The Quick Start Signal Source included with the Analyzer is recommended.

For more information on transferring data, see chapter 5, "Data Transfer."

Running the Programs

To run a program, perform the following steps:

- Load the program into your computer.
- Connect the signal source and HP-IB cables.
- Run the program and follow any directions on the computer's display.

Data Formats

The Analyzer can use two formats (ASCII and BLOCK) when transferring data. Block format sends a definite length block of data and is much faster than ASCII. It is the recommended method if your computer is capable of handling this format.

Program Descriptions

The following is a brief description of the individual programs.

ASCII Transfer of vs Time and Histogram Data

This program consists of two parts: ASCII transfer of frequency vs Time data and then ASCII transfer of the histogram based on vs Time data. It requires a signal with some FM modulation. Use the Frequency Modulation signal on the Quick Start Signal Source.

The program does the following:

- Initialize the Analyzer and set the data format to ASCII.
- Transfer the frequency vs Time data to the computer and call the appropriate graphing subprogram.
- Transfer the histogram data based on the vs Time graph and call the appropriate graphing subprogram.

This example is written in both HP BASIC and QuickBASIC.

BLOCK Transfer of vs Time Data

This program transfers and plots an FM signal using block format. Use the Frequency Modulation signal on the Quick Start Signal Source. In the HP BASIC example, a TRANSFER statement is used to move the data into the computer. Note that the COUNT number is equal to the number of data bytes plus one. This is because the instrument sends a line feed at the end of the block. Also, note that the dimension of the BUFFER used to store the data is equal to the number of measurements plus 1.

This example is written in both HP BASIC and QuickBASIC.

BLOCK Transfer of Fast Histogram Data

This program transfers data taken using the Fast Histogram function of the Analyzer. The Fast Histogram function is useful for taking large (millions or more) measurements very quickly. The Data-to-Clock signal on the Quick Start signal source is recommended.

This example is written in both HP BASIC and QuickBASIC.

Using the :STATus:OPERation Register

This program demonstrates the use of the :STATus:OPERation Register for the purpose of alerting the computer when a measurement is complete. When the Analyzer is in the process of making a measurement, bit 4 of the :STATus:OPERation Condition Register is set to one. When the measurement is complete, the bit is set to zero.

Overview

In order for an SRQ to be generated, the :STATus:OPERation:NTRansition Register's bit 4 (corresponding to the same bit in the :STATus:OPERation condition Register) must be set to one, the :STATus:OPER:PTR Register's bit 4 set to zero, and the :STATus:OPER:ENABLE Register's bit 4 set to one. When the operation is complete, bit 4 of the :STATus:OPERation condition Register goes to zero, the transition is passed through the transition filter and sets bit 4 of the Operation Event register, which in turn sets bit 7 of the Status Byte Register. By setting bit 7 of the service request enable mask, an SRQ will then be generated. The program will indicate each phase of a measurement. No data is plotted.

This example is written in both HP BASIC and QuickBASIC.

*Using the
:SYSTem:PRINT?
Command*

This program demonstrates how to use a printer over HP-IB.

Example 1. ASCII Transfer of vs Time and Histogram Data (HP BASIC)

```

10 !This program demonstrates using the ASCII format for
20 !data output on the HP 53310A.
30 !The data taken in this program uses Frequency vs Time, and
40 !needs an FM signal source for Channel A.
50 !A good source is the Frequency Modulation signal generated
60 !by the Quick Start Signal Source that is included with
70 !the HP 53310A.
80 !
90 Hp53310a=712 !Select code 7, address 12
100 INTEGER I
110 CLEAR SCREEN
120 CLEAR Hp53310a
130 INPUT "Connect an FM signal to Channel A. Press RETURN.",A$
140 OUTPUT Hp53310a;"*RST" !Reset the HP 53310A
150 OUTPUT Hp53310a;"CONF:XTIM:FREQ" !Autoscale for FREQ VS TIME
160 OUTPUT Hp53310a;"FORM ASC" !Output format = ASCII
170 !
180 GOSUB Example_1 !Get frequency vs. time results
190 !
200 INPUT "Press ENTER for Histogram from Vs Time Data.",A$
210 !
220 CLEAR SCREEN
230 GOSUB Example_2 !Get Histogram from vs. time results
240 !
250 GOTO End_of_program
260 !
270 !
280 Example_1: !
290 !Dimension arrays which will be reading vs time data from the HP53310A
300 !For Vs Time *Panorama* data, the dimension necessary would be 450*N where
310 !N is the number of screens (450 samples/screen) acquired
320 INTEGER Meas_a(450) !Dimensioned for vs. time data
330 !
340 !Dimension arrays which will hold the processed results
350 !For Vs Time *Panorama* data, the dimension necessary would be 450*N where
360 !N is the number of screens (450 samples/screen) acquired
370 REAL Freq_a(450) !Frequency dimension
380 REAL Time_a(450) !Time dimension
390 !
400 !Read just measurement results on Channel A
410 PRINT
420 PRINT "EXAMPLE 1: MEASURING FREQUENCY A VS TIME"
430 OUTPUT Hp53310a;"INIT" !Initiate measurement
440 OUTPUT Hp53310a;"*WAI" !Wait for command to complete
450 OUTPUT Hp53310a;"DIM:SIZE?"

```

Overview

Example 1. ASCII Transfer of vs Time and Histogram Data (HP BASIC) (Continued)

```
460 ENTER Hp53310a;Num_samples
470 OUTPUT Hp53310a;"DIM:SCAL?"           !Frequency dimension scale
480 ENTER Hp53310a;Freq_scale
490 OUTPUT Hp53310a;"DIM:OFFS?"         !Frequency dimension offset
500 ENTER Hp53310a;Freq_offset
510 OUTPUT Hp53310a;"DIM2:SCAL?"       !Time dimension scale
520 ENTER Hp53310a;Time_scale
530 OUTPUT Hp53310a;"DIM2:OFFS?"       !Time dimension offset
540 ENTER Hp53310a;Time_offset
550 OUTPUT Hp53310a;"FETC:XTIM:FREQ?"   !Request measurement results
560 !
570 FOR I=1 TO Num_samples               !Read measurement results
580   ENTER Hp53310a USING "%,K";Meas_a(I)
590 NEXT I
600 !
610 Graph_wfm: !
620 CALL Label_it(Freq_scale,Freq_offset,Time_offset,Time_scale,
Num_samples,Meas_a(*))
630 !
640 FOR I=1 TO Num_samples               !Convert date to frequency and time
650   Time_a(I)=I*Time_scale+Time_offset
660   IF Meas_a(I)<>-1 THEN
670     Freq_a(I)=Meas_a(I)*Freq_scale+Freq_offset
680   ELSE
690     Freq_a(I)=Meas_a(I)
700   END IF
710 NEXT I
720 RETURN
730 !
740 !
750 Example_2: !
760 !Dimension arrays which will be reading histogram data from the HP53310A
770 !Dimension of 450 was chosen because it works for all histograms:
780 !Fast Histogram Data would require an array dimension to 450
790 !because it has a maximum of 450 bins.
800 !Histogram from Vs Time Data would require an array dimension of 257
810 !if panorama disabled and 225 if panorama enabled.
820 REAL Prob_a(450)                     !Dimensioned for histogram
830                                     ! (Probability dimension)
840 !
850 !Dimension arrays which will hold the processed results
860 !Note that Fast Histogram Data would require an array dimension to 450
870 !because it has a maximum of 450 bins
880 REAL Freq_bin_a(450)                 !Frequency bin dimension
890 !
```

Example 1. ASCII Transfer of vs Time and Histogram Data (HP BASIC) (Continued)

```

900 PRINT
910 PRINT "EXAMPLE 2: READING HISTOGRAM FROM VS TIME DATA"
920 OUTPUT Hp53310a;"CALC:DIM:SIZE?"           !Number of histogram bins
930 ENTER Hp53310a;Num_bins
940 OUTPUT Hp53310a;"CALC:DIM2:SCAL?"         !Frequency dimension scale
950 ENTER Hp53310a;Freq_scale
960 OUTPUT Hp53310a;"CALC:DIM2:OFFS?"        !Frequency dimension offset
970 ENTER Hp53310a;Freq_offset
980 !
990 OUTPUT Hp53310a;"CALC:HIST ON"
1000 OUTPUT Hp53310a;"FETC:HIST:FREQ?"       !Request measurement results
1010 !
1020 FOR I=1 TO Num_bins                      !Read measurement results
1030   ENTER Hp53310a USING "%,K";Prob_a(I)
1040 NEXT I
1050 Max_prob=MAX(Prob_a(*))
1060 OUTPUT Hp53310a;"CALC:HIST:SAMP:IN?"
1070 ENTER Hp53310a;Num_samples              !Number of samples in histogram
1080 !
1090 Graph_hist: !
1100 CALL Hist_label(Freq_offset,Freq_scale,Max_prob,Num_bins,Num_samples,
Prob_a(*))
1110 !
1120 FOR I=1 TO Num_bins                      !Compute bin frequencies
1130   Freq_bin_a(I)=I*Freq_scale+Freq_offset
1140 NEXT I
1150 RETURN
1160 !=====
1170 End_of_program: !
1180 END
1190 !
1200 !
1210 SUB Label_it(F_scale,F_offset,T_offset,T_scale,Num_samples,
INTEGER Meas_a(*))
1220 CLEAR SCREEN
1230 GINIT
1240 PLOTTER IS CRT,"INTERNAL"
1250 GRAPHICS ON
1260 VIEWPORT .05*(100*MAX(1,RATIO)),.95*(100*MAX(1,RATIO)),
.30*(100*MAX(1,1/RATIO)),.95*(100*MAX(1,1/RATIO))
1270 FRAME
1280 WINDOW 1,Num_samples+1,0,256
1290 AXES 9,8.03125,226,128,5,4             ! Set up axes to emulate HP 53310A
1300 CLIP OFF
1310 MOVE 226,-15

```

Overview

Example 1. ASCII Transfer of vs Time and Histogram Data (HP BASIC) (Continued)

```
1320  CSIZE 5
1330  LORG 6
1340  LABEL USING "11A";"Frequency A"
1350  CSIZE 3
1360  MOVE 1,0
1370  LORG 1
1380  LABEL USING "D.5DE,2A";F_offset,"Hz"
1390  MOVE 1,256
1400  LORG 3
1410  LABEL USING "D.5DE,2A";(F_offset+256*F_scale),"Hz"
1420  MOVE 1,128
1430  LORG 1
1440  LABEL USING "D.5DE,2A";((2*F_offset+256*F_scale)/2),"Hz"
1450  MOVE 1,0
1460  LORG 3
1470  LABEL USING "S3D.DE,A";T_offset+T_scale,"s"
1480  MOVE 451,0
1490  LORG 9
1500  LABEL USING "S3D.DE,A";(T_offset+451*T_scale),"s"
1510  PEN 1
1520  MOVE 226,0
1530  LORG 6
1540  LABEL USING "SZ.2D,A";(T_offset+226*T_scale),"s"
1550  MAT SEARCH Meas_a,LOC(>-1);Location,1
1560  MOVE Location,Meas_a(Location)
1570  FOR I=1 TO (Num_samples)
1580      IF Meas_a(I)<>-1 THEN
1590          DRAW I,Meas_a(I)
1600      END IF
1610  NEXT I
1620  SUBEND
1630  !
1640  !
1650  SUB Hist_label(Freq_offset,Freq_scale,Max_prob,Num_bins,Num_samples,
REAL Prob_a(*))
1660  CLEAR SCREEN
1670  GINIT
1680  PLOTTER IS CRT,"INTERNAL"
1690  GRAPHICS ON
1700  VIEWPORT .15*(100*MAX(1,RATIO)),.75*(100*MAX(1,RATIO)),
.30*(100*MAX(1,1/RATIO)),.95*(100*MAX(1,1/RATIO))
1710  FRAME
1720  Max_x=2*Num_bins
1730  WINDOW 1,Max_x,0,Max_prob
1740  PEN 1
```

Overview

Example 1. ASCII Transfer of vs Time and Histogram Data (HP BASIC) (Continued)

```
1750 CLIP OFF
1760 CSIZE 3
1770 MOVE 1,0
1780 LORG 3
1790 LABEL USING "D.5DE,2A";Freq_offset+Freq_scale,"Hz"
1800 MOVE Max_x,0
1810 LORG 9
1820 LABEL USING "D.5DE,2A";Freq_offset+257*Freq_scale,"Hz"
1830 MOVE 1,Max_prob
1840 LORG 8
1850 LABEL USING "D.3D,A";Max_prob*100,"% "
1860 MOVE Max_x/2,-Max_prob/15
1870 LORG 6
1880 LABEL USING "6D,X,7A";Num_samples,"Samples"
1890 J=1
1900 MOVE J,0
1910 FOR I=1 TO Num_bins
1920   DRAW J,Prob_a(I)
1930   J=J+2
1940   DRAW J,Prob_a(I)
1950 NEXT I
1960 SUBEND
```

Overview

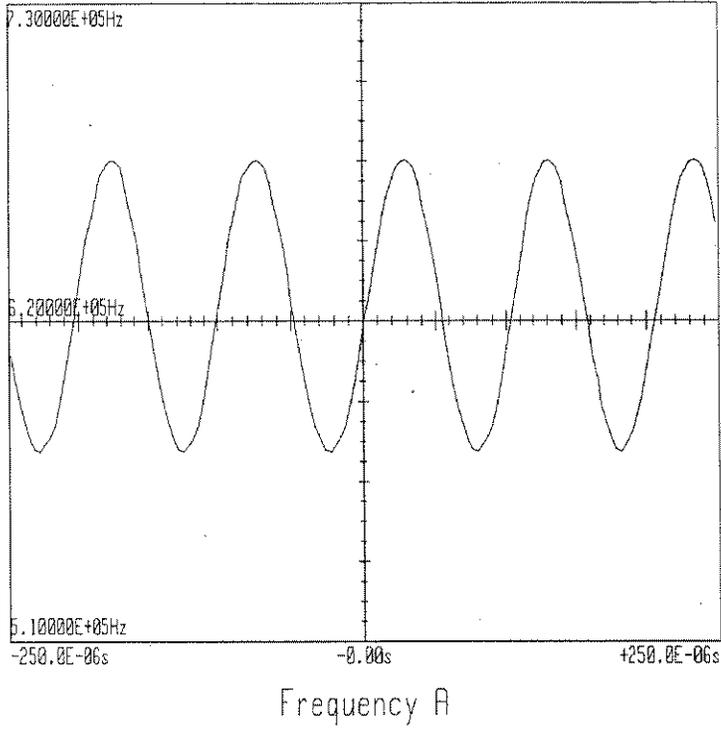


Figure D-1. ASCII Transfer of vs Time Data (HP BASIC).

Overview

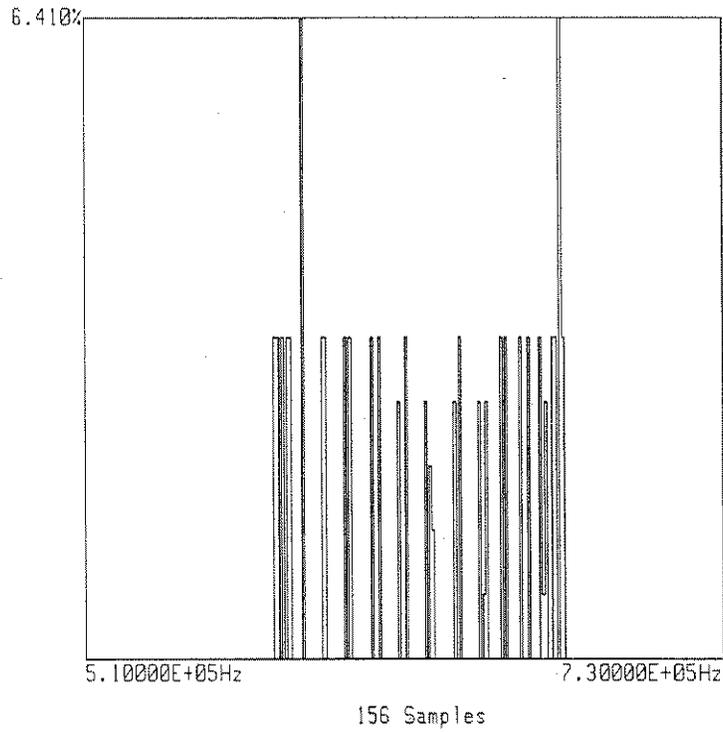


Figure D-2. ASCII Transfer of Histogram Data (HP BASIC).

Overview

Example 2. BLOCK Transfer of vs Time Data (HP BASIC)

```
10 !This program demonstrates using the BLOCK format for
20 !integer (vs time) data output on the HP 53310A.
30 !The data taken in this program uses a Frequency vs Time measurement.
40 !The program assumes an FM signal is connected to channel A.
50 !A good signal source is the Frequency Modulation signal
60 !generated by the Quick Start Signal Source that is included
70 !with the HP 53310A.
80 !
90 !Dimension arrays which will be reading data from the HP 53310A
100 !For Vs Time *Panorama* data, the dimension necessary would be 450*N+1
110 !where N is the number of screens (450 samples/screen) acquired.
120 !
130 OPTION BASE 1
140 INTEGER Buff(451) BUFFER      !450 + 1 to handle linefeed
150 !
160 !Dimension arrays which will hold the processed results
170 !For Vs Time *Panorama* data, the dimension necessary would be 450*N where
180 !N is the number of screens (450 samples/screen) acquired
190 !
200 REAL Freq_a(450)              !Frequency dimension
210 REAL Time_a(450)              !Time dimension
220 !
230 Hp53310a=712
240 ASSIGN @Hp53310a TO 712;FORMAT OFF      !This assign used for TRANSFER
250 ASSIGN @Controller_buf TO BUFFER Buff(*);FORMAT OFF
260 !
270 !
280 INPUT "Connect an FM signal to Channel A. Press RETURN",A$
290 CLEAR Hp53310a
300 OUTPUT Hp53310a;"*RST"                !Reset the HP 53310A
310 OUTPUT Hp53310a;"CONF:XTIM:FREQ"      !Autoscale for FREQ VS TIME
320 OUTPUT Hp53310a;"FORM BLOC"          !Output format = BLOCK
330 !
340 GOSUB Measure                          !Get measurement results
350 !
360 GOTO End_of_program
370 !
380 Measure: !
390 !
400 !Read measurement results
410 CLEAR SCREEN
420 GCLEAR
430 PRINT
440 PRINT "EXAMPLE 1: MEASURING/READING FREQ A VS TIME MEASUREMENT"
450 !
```

Example 2. BLOCK Transfer of vs Time Data (HP BASIC) (Continued)

```

460 OUTPUT Hp53310a;"INIT"           !Initiate measurement.
470 OUTPUT Hp53310a;"*OPC?"         !Synchronize to measurement
480 ENTER Hp53310a;Opc_response     !complete, Opc_response=1.
490 OUTPUT Hp53310a;"DATA?"        !Request measurement results.
500 GOSUB Read_header              !Read block data header.
510 RESET @Controller_buf
520 TRANSFER @Hp53310a TO @Controller_buf;COUNT Num_wfm_bytes+1,WAIT
530 Num_samples=Num_wfm_bytes/2     !16 bit integer = 2 bytes
540 OUTPUT Hp53310a;"DIM:SCAL?"     !Frequency dimension scale
550 ENTER Hp53310a;Freq_scale
560 OUTPUT Hp53310a;"DIM:OFFS?"     !Frequency dimension offset
570 ENTER Hp53310a;Freq_offset
580 OUTPUT Hp53310a;"DIM2:SCAL?"    !Time dimension scale
590 ENTER Hp53310a;Time_scale
600 OUTPUT Hp53310a;"DIM2:OFFS?"    !Time dimension offset
610 ENTER Hp53310a;Time_offset
620 Graph_wfm: !Graph the results
630 !
640 CALL Label_it(Freq_scale,Freq_offset,Time_offset,Time_scale,Num_samples,
Buff(*)
650 !
660 FOR I=1 TO Num_samples           !Convert data to frequency and time.
670   Time_a(I)=I*Time_scale+Time_offset
680   IF Buff(I)<>-1 THEN             !-1 means no data at that pixel
690     Freq_a(I)=Buff(I)*Freq_scale+Freq_offset
700   ELSE
710     Freq_a(I)=Buff(I)
720   END IF
730 NEXT I
740 RETURN
750 !
760 Read_header: !
770 ENTER Hp53310a USING "#,A";Pound_sign$
780 ENTER Hp53310a USING "#,A";Num_len_bytes$
790 ENTER Hp53310a USING "#,"&Num_len_bytes$&"A";Num_wfm_bytes$
800 Num_wfm_bytes=VAL(Num_wfm_bytes$)
810 RETURN
820 !
830 End_of_program: !
840 END
850 !
860 !
870 SUB Label_it(F_scale,F_offset,T_offset,T_scale,Num_samples,
INTEGER Buff(*)
880 CLEAR SCREEN

```

Overview

Example 2. BLOCK Transfer of vs Time Data (HP BASIC) (Continued)

```
890 GINIT
900 PLOTTER IS CRT,"INTERNAL"
910 GRAPHICS ON
920 VIEWPORT .05*(100*MAX(1,RATIO)),.95*(100*MAX(1,RATIO)),
.30*(100*MAX(1,1/RATIO)),.95*(100*MAX(1,1/RATIO))
930 FRAME
940 WINDOW 1,451,0,256
950 AXES 9,8.03125,226,128,5,4 ! Set up axes to emulate HP 53310A.
960 CLIP OFF
970 MOVE 226,-15
980 CSIZE 5
990 LORG 6
1000 LABEL USING "11A";"Frequency A"
1010 CSIZE 3
1020 MOVE 1,0
1030 LORG 1
1040 LABEL USING "D.5DE,2A";F_offset,"Hz"
1050 MOVE 1,256
1060 LORG 3
1070 LABEL USING "D.5DE,2A";(F_offset+256*F_scale),"Hz"
1080 MOVE 1,128
1090 LORG 1
1100 LABEL USING "D.5DE,2A";((2*F_offset+256*F_scale)/2),"Hz"
1110 MOVE 1,0
1120 LORG 3
1130 LABEL USING "S3D.DE,A";T_offset+T_scale,"s"
1140 MOVE 451,0
1150 LORG 9
1160 LABEL USING "S3D.DE,A";(T_offset+451*T_scale),"s"
1170 PEN 1
1180 MOVE 226,0
1190 LORG 6
1200 LABEL USING "SZ.2D,A";(T_offset+226*T_scale),"s"
1210 MAT SEARCH Buff,LOC(>-1);Location !Move to first real data point
1220 MOVE Location, Buff(Location)
1230 FOR I=1 TO (Num_samples)
1240 IF Buff(I)<>-1 THEN
1250 DRAW I, Buff(I)
1260 END IF
1270 NEXT I
1280 SUBEND
```

Overview

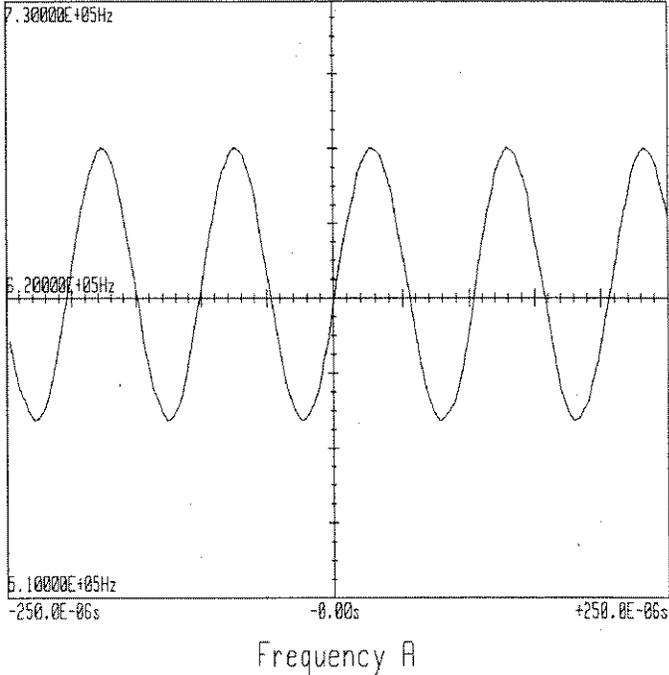


Figure D-3. BLOCK Transfer of vs Time Data (HP BASIC).

Overview

Example 3. BLOCK Transfer of Fast Histogram Data (HP BASIC)

```
10 !This program demonstrates using the BLOCK format for
20 !floating point (histogram) data output on the HP 53310A.
30 !The data taken in this program uses Time Interval measurements, and
40 !needs input sources for both Channels A and B. A good source
50 !is the Quick Start Signal Source included with the HP 53310A.
60 !Use the Data-to-Clock jitter signal.
70 !
80 !Dimension buffer which will be reading data from the HP 53310A
90 !450 was chosen because it works for all histograms:
100 !Fast Histogram Data requires space for 450 bins.
110 !Histogram from Vs Time Data would require space for 257 bins
120 !if panorama disabled and 225 bins if panorama enabled.
130 !Each individual histogram probability value is 8 bytes.
140 OPTION BASE 1
150 REAL Buff(451) BUFFER           !450 + 1 to take care
160                               !of linefeed terminator.
170                               !Buff(*) dimensioned for
180                               !Fast Histogram.
190 !Now dimension array which will hold the processed results.
200 !450 was chosen because it works for all histograms:
210 !Fast Histogram Data requires space for 450 bins.
220 !Histogram from Vs Time Data would require space for 257 bins
230 !if panorama disabled and 225 bins if panorama enabled.
240 REAL Tint_bin_a(450)           !For Fast Histogram
250 !
260 Hp53310a=712
270 ASSIGN @Hp53310a TO 712;FORMAT OFF !This assignment only used for
280                               !TRANSFER statement.
290 ASSIGN @Controller_buf TO BUFFER Buff(*);FORMAT OFF
300 CLEAR Hp53310a
310 INPUT "Connect a signal to Channel A and B. Press RETURN",AS
320 OUTPUT Hp53310a;"*RST"         !Reset the HP 53310A
330 OUTPUT Hp53310a;"CONF:HIST:TINT" !Autoscale for TI FAST HIST
340 OUTPUT Hp53310a;"FORM BLOC"    !Output format = BLOCK
350 !
360 GOSUB Measure                 !Get measurement results
370 !
380 GOTO End_of_program
390 !
400 Measure: !
410 !
420 !Read just measurement results
430 CLEAR SCREEN
440 GCLEAR
450 PRINT "Making histogram measurements"
```

Example 3. BLOCK Transfer of Fast Histogram Data (HP BASIC) (Continued)

```

460 OUTPUT Hp53310a;"INIT"           !Initiate measurement
470 OUTPUT Hp53310a;"*OPC?"         !Synchronize to meas complete
480 ENTER Hp53310a;Opc_response
490 OUTPUT Hp53310a;"DIM2:SCAL?"    !TI dimension scale
500 ENTER Hp53310a;Tint_scale
510 OUTPUT Hp53310a;"DIM2:OFFS?"    !TI dimension offset
520 ENTER Hp53310a;Tint_offset
530 OUTPUT Hp53310a;"DATA?"        !Request measurement results
540 GOSUB Read_header              !Read block data header
550 RESET @Controller_buf
560 !Must have count = Num_wfm_bytes + 1 to take care of linefeed
570 !at end of block.
580 TRANSFER @Hp53310a TO @Controller_buf;COUNT Num_wfm_bytes+1,WAIT
590 Num_bins=Num_wfm_bytes/8
600 Max_prob=0
610 Max_prob=MAX(Buff(*))
620 OUTPUT Hp53310a;"HIST:SAMP:IN?"
630 ENTER Hp53310a;Num_samples
640 Graph_hist:      ! Graph the data
650 CALL Hist_label (Tint_offset,Tint_scale,Max_prob,Num_bins,Num_samples,
Buff(*))
660 !
670 FOR I=1 TO Num_bins           !Compute bins
680   Tint_bin_a(I)=I*Tint_scale+Tint_offset
690 NEXT I
700 RETURN
710 !
720 Read_header:      !
730 ENTER Hp53310a USING "#,A";Pound_sign$
740 ENTER Hp53310a USING "#,A";Num_len_bytes$
750 ENTER Hp53310a USING "#,"&Num_len_bytes$&"A";Num_wfm_bytes$
760 Num_wfm_bytes=VAL(Num_wfm_bytes$) !Number of DATA bytes to be transferred
770 RETURN
780 !
790 End_of_program:      !
800 END
810 !
820 !
830 SUB Hist_label (Tint_offset,Tint_scale,Max_prob,Num_bins,Num_samples,
Buff(*))
840 CLEAR SCREEN
850 GINIT
860 PLOTTER IS CRT,"INTERNAL"
870 GRAPHICS ON
880 VIEWPORT .15*(100*MAX(1,RATIO)),.75*(100*MAX(1,RATIO)),

```

Overview

Example 3. BLOCK Transfer of Fast Histogram Data (HP BASIC) (Continued)

```
.30*(100*MAX(1,1/RATIO)),.95*(100*MAX(1,1/RATIO))
890  FRAME
900  Max_x=2*Num_bins
910  WINDOW 1,Max_x,0,Max_prob
920  PEN 1
930  CLIP OFF
940  CSIZE 3
950  MOVE 1,0
960  LORG 3
970  LABEL USING "D.4DE,A";Tint_offset+Tint_scale,"s"
980  MOVE Max_x,0
990  LORG 9
1000 LABEL USING "D.4DE,A";Tint_offset+450*Tint_scale,"s"
1010 MOVE 1,Max_prob
1020 LORG 8
1030 LABEL USING "D.3D,A";Max_prob*100,"% "
1040 MOVE Max_x/2,-Max_prob/15
1050 LORG 6
1060 LABEL USING "6D,X,7A";Num_samples,"Samples"
1070 J=1
1080 MOVE J,0
1090 FOR I=1 TO Num_bins
1100   DRAW J, Buff(I)
1110   J=J+2
1120   DRAW J, Buff(I)
1130 NEXT I
1140 SUBEND
```

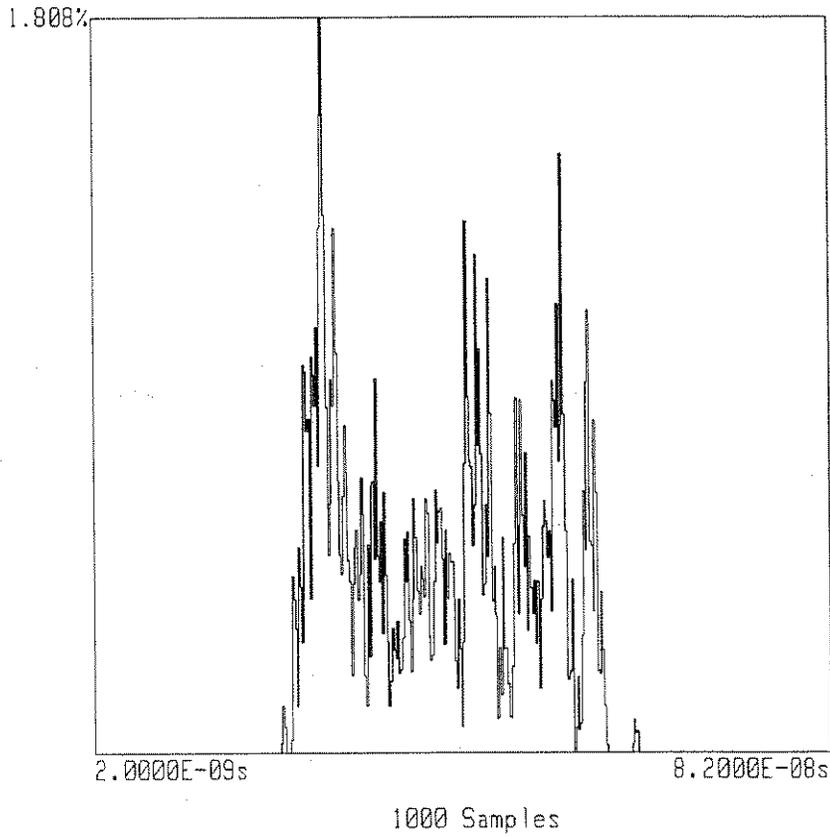


Figure D-4. BLOCK Transfer of Fast Histogram Data (HP BASIC).

Overview

Example 4. Using the :STATus:OPERation Register (HP BASIC)

```
10 !This program demonstrates the use of the Operation Status Register.
20 !The program is set up to cause the HP 53310A to enable SRQ when
30 !a measurement completes.
40 !
50 !To run this demo, you will need to connect an FM signal to Channel A.
60 !A good source is the Quick Start Signal Source that is included
70 !with the HP 53310A.
80 !
90 CLEAR SCREEN !Clear the CRT
100 ASSIGN @Hp53310a TO 712 !Set up path for HP 53310A
110 CLEAR @Hp53310a !Clear any messages
120 OUTPUT @Hp53310a;"*RST" !Reset the HP 53310A
130 OUTPUT @Hp53310a;"DISP:MENU FUNC" !Bring up the function menu
140 OUTPUT @Hp53310a;"*CLS;:STAT:PRES" !Clear/preset status data structures
150 INPUT "Connect an FM signal to Input A; press RETURN",AS
160 OUTPUT @Hp53310a;"CONF:XTIM:FREQ" !Autoscale freq A vs time measurement
170 !The HP 53310A CRT will show the
180 !measurement.
190 !
200 !To show how the Operation Status Register
210 !can be used to enable SRQ, do the following:
220 OUTPUT @Hp53310a;"*CLS;STAT:PRES" !Clear/preset status data structures
230 OUTPUT @Hp53310a;"STAT:OPER:ENAB 16" !Set the Operation Status Enable
240 !Register
250 OUTPUT @Hp53310a;"STAT:OPER:NTR 16;PTR 0" !Set the Operation Status
260 !Transition Filter
270 OUTPUT @Hp53310a;"*SRE 128" !Set the Service Request Enable
280 !register to pull SRQ when the
290 !Operation Status Bit is set
300 !
310 OUTPUT @Hp53310a;"INIT" !Start a new measurement
320 ON INTR 7 GOTO Srq !If SRQ then go to Srq and process it
330 ENABLE INTR 7;2 !Enable the interrupt
340 DISP "Measurement in progress."
350 !
360 !
370 LOOP !Stay in this loop until meas completes
380 END LOOP !An SRQ will exit this loop.
390 !
400 !
410 Srq: ! An SRQ caused the program to branch here.
420 OFF INTR 7
430 PRINT "SRQ generated by HP 53310A."
440 PRINT
450 Test_bit=SPOLL(@Hp53310a) !Serial Poll to clear SRQ
```

Example 4. Using the :STATus:OPERation Register (HP BASIC) (Continued)

```

460 IF BIT(Test_bit,7) THEN      !Bit 7 goes high because of Operation Status
470                               !Register activity.
480 DISP                          !Clear display line
490 PRINT "Operation Status Register enabled SRQ."
500 PRINT
510 OUTPUT @Hp53310a;"STAT:OPER?" !Query the Operation Status Register
520                               !contents.
530 ENTER @Hp53310a;Osr_value     !Put the contents in Osr_value
540 PRINT "Operation Status Register returned a value of ";Osr_value
550 IF BIT(Osr_value,4) THEN      !If bit 4 then measurement is complete.
560                               !Note: Other bits may also be set.
570                               !Consult the manual for details.
580 PRINT
590 PRINT "Operation Status Register Bit 4 set (See manual for other Bits)."

```

Overview

Example 5. Using the :SYSTEM:PRINT? Command (HP BASIC)

```
10 ! This program demonstrates using a computer to access
20 ! the printer dump features of the HP 53310A
30 ! Modulation Domain Analyzer.
40 ! The program also demonstrates the use of the
50 ! Operation Status Register and Service Request Enable Register
60 ! to determine when a print completes; the HP 53310A communicates this
70 ! fact via SRQ.
80 ! To use this program:
90 !
100 ! 1) Make a measurement using the front panel of the instrument.
110 ! 2) Set up the display to your liking.
120 ! 3) RUN this program.
130 !
140 !
150 Hpib=7 ! Selctt code of HP-IB Interface
160 Printer=1 ! HP-IB address of printer
170 Hp53310a=12 ! HP-IB address of HP53310A
180 ASSIGN @Hp53310a TO Hpib*100+Hp53310a
190 !
200 !
210 CLEAR @Hp53310a
220 OUTPUT @Hp53310a,"STAT:OPER?" ! Clear the Operation Status Register
230 ENTER @Hp53310a;Osr
240 Temp=SPOLL(@Hp53310a) ! Make sure the status byte is cleared
250 OUTPUT @Hp53310a,"STAT:OPER:ENAB 256;PTR 0;NTR 256" ! Set up for SRQ on
260 OUTPUT @Hp53310a;"*SRE 128" ! PRINT complete
270 !
280 !
290 !*****
300 ! THE FOLLOWING TELLS THE HP 53310A TO DO A PRINTER DUMP, AND DOES THE
310 ! NECESSARY HP-IB ADDRESSING
320 OUTPUT @Hp53310a;"SYST:PRIN?"
330 SEND Hpib;UNL TALK Hp53310a LISTEN Printer DATA
340 !*****
350 !
360 !
370 !
380 ! WAIT FOR AN SRQ (WHICH HAPPENS WHEN THE HP 53310A IS FINISHED PRINTING)
390 ON INTR Hpib GOTO Print_done
400 ENABLE INTR Hpib;2
410 Wait:GOTO Wait
420 !
430 Print_done:!
440 OUTPUT @Hp53310a;"STAT:OPER?" ! Clear out OSR
```

Example 5. Using the :SYSTEM:PRINT? Command (HP BASIC) (Continued)

```
450 ENTER @Hp53310a;Osr
460 Temp=SPOLL(@Hp53310a)           ! Clear status byte
470 !
480 LOCAL @Hp53310a                 ! Return HP 53310A to local
490 !
500 END
```

Overview

Example 6. ASCII Transfer of vs Time and Histogram Data (QuickBASIC)

```
DECLARE SUB histgraph (freqohist!, freqshist!, numbins!, numsamples!, maxbin!,  
probability!())  
DECLARE SUB LABELIT (freqscale!, freqoffset!, timeoffset!, timescale!, samples!,  
result() AS INTEGER)
```

'This program demonstrates the use of ASCII transfer.

'To run this program you must have an HP 82335A HP-IB interface card.

'You will also need an FM source connected to Channel A.

'A good source is the Quick Start Signal Source that is included with
'the HP 53310A.

```
REM $INCLUDE: 'QBSETUP.BAS'           'Required by HP 82335A  
SOURCE& = 712                        'HP 53310A at address 12  
isc& = 7                             'Select code 7  
state% = 1                           'Variable used in some subprograms  
match$ = CHR$(44)                   'comma ","  
SCREEN 11                            'Screen is VGA  
DIM i AS INTEGER                    'i is used for loops
```

```
CALL iomatch(isc&, match$, state%)   'Terminate enters on comma  
CLS 0  
CALL IOEOI(isc&, state%)            'Make sure EOI enabled  
CODE$ = "*RST"                      'Reset instrument  
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))  
CODE$ = "CONF:XTIM:FREQ"            'Configure for measurement  
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))  
CODE$ = "FORM ASC"                 'Use ASCII format  
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
```

```
INPUT "Connect an FM signal to Channel A. Press ENTER to continue", a$  
GOSUB EXAMPLE1
```

```
LOCATE 25, 1  
INPUT "Press RETURN for Histogram from vs. time data", a$  
CLS 0  
GOSUB EXAMPLE2  
PRINT "All done!"  
GOTO ENDPROG
```

EXAMPLE1:

```
DIM meas AS STRING * 24              'meas used for data entry
```

Example 6. ASCII Transfer of vs Time and Histogram Data (QuickBASIC) (Continued)

```

DIM result(1 TO 450) AS INTEGER      'result() contains value from meas
DIM freqdata(1 TO 450)              'frequency array
DIM timedata(1 TO 450)              'time array
DIM samp AS STRING * 24             'These strings used throughout
DIM freqs AS STRING * 24            'program
DIM freqo AS STRING * 24
DIM times AS STRING * 24
DIM timeo AS STRING * 24

```

```
LOCATE 24, 1
```

```
PRINT "Example 1: Measuring frequency vs. time"
```

```

CODE$ = "DISP:TRAC:CONNECT OFF"    'Do not connect data points
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CODE$ = "INIT"                      'Make measurement
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CODE$ = "*WAI"                      'Wait for data
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CODE$ = "DIM:SIZE?"                'Get number of samples
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, samp, 24, ACTX%)
samples = VAL(samp)

CODE$ = "DIM:SCAL?"                'Get frequency scale
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, freqs, 24, ACTX%)
freqscale = VAL(freqs)
CODE$ = "DIM:OFFS?"                'Get frequency offset
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, freqo, 24, ACTX%)
freqoffset = VAL(freqo)
CODE$ = "DIM2:SCAL?"                'Get time scale
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, times, 24, ACTX%)
timescale = VAL(times)
CODE$ = "DIM2:OFFS?"                'Get time offset
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, timeo, 24, ACTX%)
timeoffset = VAL(timeo)

CODE$ = "FETC:XTIM:FREQ?"          'Ask for data
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
FOR i = 1 TO samples                'Get pixel data from HP 53310A
CALL IOENTERS(SOURCE&, meas, 22, act%)
result(i) = VAL(meas)

```

Overview

Example 6. ASCII Transfer of vs Time and Histogram Data (QuickBASIC) (Continued)

```
NEXT i

FOR i = 1 TO samples
    timedata(i) = i * timescale + timeoffset 'Convert result() to frequency
                                         'and time data
    IF result(i) > -1 THEN
        freqdata(i) = result(i) * freqscale + freqoffset
    ELSE
        freqdata(i) = result(i)
    END IF
NEXT i

'This subprogram labels graph and plots data

CALL LABELIT(freqscale, freqoffset, timeoffset, timescale, samples, result())
RETURN

EXAMPLE2:
DIM probability(1 TO 450) 'Array for data
DIM frequencybin(1 TO 450) 'Frequency bin data
DIM numb AS STRING * 24
DIM nums AS STRING * 24
DIM freqsh AS STRING * 24
DIM freqoh AS STRING * 24
DIM samph AS STRING * 24

LOCATE 24, 1
PRINT "EXAMPLE 2: Reading histogram from vs. time data"
CODE$ = "CALC:DIM:SIZE?" 'Get number of histogram bins
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, numb, 24, ACTX%)
numbins = VAL(numb)
CODE$ = "CALC:DIM2:SCAL?" 'Get frequency scale
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, freqsh, 24, ACTX%)
freqshist = VAL(freqsh)
CODE$ = "CALC:DIM2:OFFS?" 'Get frequency offset
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, freqoh, 24, ACTX%)
freqohist = VAL(freqoh)
CODE$ = "CALC:HIST ON" 'Enable histogram display
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
```

Example 6. ASCII Transfer of vs Time and Histogram Data (QuickBASIC) (Continued)

```

CODE$ = "FETC:HIST:FREQ?"           'Ask for data
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
FOR i = 1 TO numbins               'Get data from HP 53310A
CALL IOENTERS(SOURCE&, meas, 24, act%)
probability(i) = VAL(meas)
NEXT i

CODE$ = "CALC:HIST:SAMP:IN?"       'Get total number of samples
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, nums, 24, ACTX%)
numsamples = VAL(nums)
maxbin = 0                         'Determine maximum probability
FOR i = 1 TO numbins
IF probability(i) > maxbin THEN maxbin = probability(i)
NEXT i

'This subprogram labels graph and plots histogram

CALL histgraph(freqshist, freqohist, numbins, numsamples, maxbin, probability())

FOR i = 1 TO numbins               'Calculate bin frequency
frequencybin(i) = i * freqshist + freqohist
NEXT i

ENDPROG:
'end of program

SUB histgraph (freqshist, freqohist, numbins, numsamples, maxbin, probability())
CLS 0
SCREEN 11
VIEW (185, 70)-(442, 370)
WINDOW (1, 0)-(numbins, maxbin)
DIM i AS INTEGER
LINE (1, maxbin)-(numbins, 0), , B
FOR i = 1 TO numbins
LINE (i, 0)-(i, probability(i))
NEXT i
LOCATE 26, 34
PRINT numsamples; "Samples"
LOCATE 25, 23
PRINT USING "####^0000"; freqohist + freqshist

LOCATE 25, 49
PRINT USING "####^0000"; freqohist + 257 * freqshist

```

Overview

Example 6. ASCII Transfer of vs Time and Histogram Data (QuickBASIC) (Continued)

```
LOCATE 5, 17
B$ = "%
PRINT USING "#.###!"; maxbin * 100; B$
END SUB

SUB LABELIT (freqscale, freqoffset, timeoffset, timescale, samples, result() AS
INTEGER)
CLS 0
SCREEN 11
VIEW (85, 70)-(555, 326)
WINDOW (1, 0)-(451, 256)
DIM i AS INTEGER
LINE (1, 256)-(451, 0), , B
LINE (1, 128)-(451, 128), , &HFF00
LINE (226, 0)-(226, 256), , &HFF00
FOR i = 1 TO samples
IF result(i) > -1 THEN
PSET (i, result(i))
END IF
NEXT i
LOCATE 24, 36
PRINT "Frequency A"
LOCATE 22, 11
PRINT USING "#.###^"; timeoffset + timescale
LOCATE 22, 62
PRINT USING "#.###^"; timeoffset + 451 * timescale
LOCATE 22, 39
PRINT USING "#.##"; timeoffset + 226 * timescale
LOCATE 21, 1
PRINT USING "##.###^"; freqoffset
LOCATE 13, 1
PRINT USING "##.###^"; (2 * freqoffset + 256 * freqscale) / 2
LOCATE 5, 1
PRINT USING "##.###^"; freqoffset + 256 * freqscale
END SUB
```

Overview

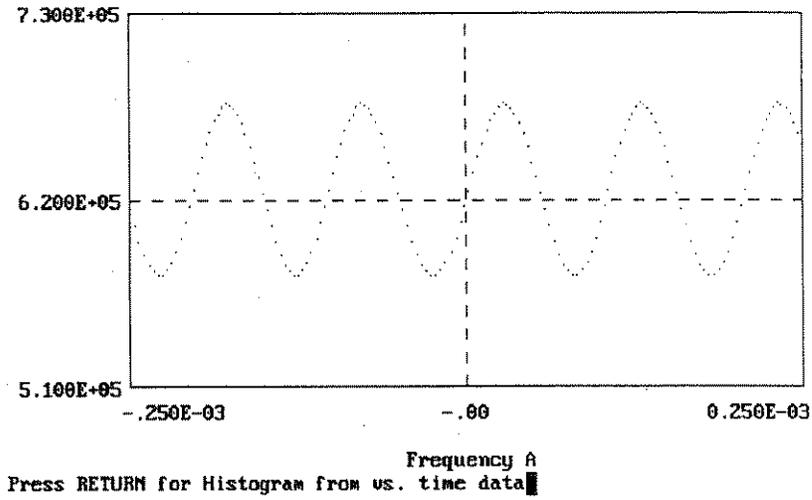
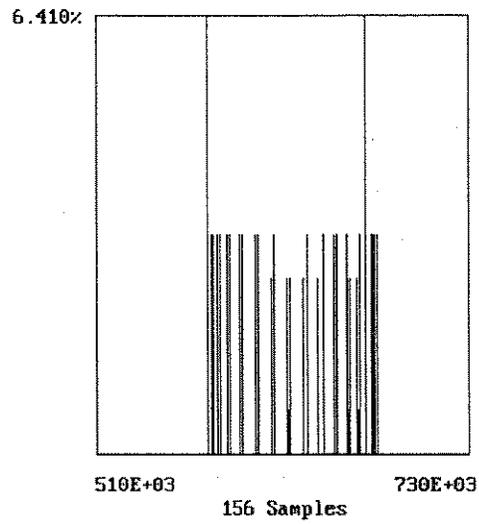


Figure D-5. ASCII Transfer of vs Time Data (QuickBASIC).

Overview



Press any key to continue

Figure D-6. ASCII Transfer of Histogram Data (QuickBASIC)

Overview

Example 7. BLOCK Transfer of vs Time Data (QuickBASIC)

```
DECLARE SUB LABELIT (freqscale!, freqoffset!, timeoffset!, timescale!, samples!,
RESULT() AS INTEGER)
' This program demonstrates the use of BLOCK data transfers
' for the HP 53310A. The HP 53310A sends over 2 byte integers for
' each measurement. This is fastest method for transferring data.
' For more details on the IO calls, see the HP 82335A manual.
'
' To run this program you must have an HP 82335A HP-IB interface card.
'
' You will also need an FM source connected to Channel A.
' A good source is the Quick Start Signal Source that is included with
' the HP 53310A.
'
REM $INCLUDE: 'QBSETUP.BAS'
SOURCE& = 712           'HP 53310A at address 12
isc& = 7                'Select code 7
state% = 1              'variable for some CALLs
SCREEN 11               'Screen is VGA
DIM i AS INTEGER        'i used in loops
CALL iomatch(isc&, CHR$(10), state%) 'Terminate IOENTERS on linefeed
CLS 0
CALL IOEOI(isc&, state%) 'Make sure EOI enabled
CODE$ = "*RST"          'Reset instrument
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CODE$ = "CONF:XTIM:FREQ" 'Configure for measurement
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CODE$ = "DISP:TRAC:CONN OFF" 'Do not connect data points
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CODE$ = "FORM BLOC"    'Use BLOCK format
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))

INPUT "Connect an FM signal to Channel A. Press ENTER to continue", a$

DIM RESULT(1 TO 450) AS INTEGER '450 measurements
DIM freqdata(1 TO 450)          '450 frequency values
DIM timedata(1 TO 450)          '450 time values
DIM OPC AS STRING * 23         'These strings are used throughout
DIM freqs AS STRING * 23       'the program
DIM freqo AS STRING * 23
DIM times AS STRING * 23
DIM timeo AS STRING * 23
DIM linefeed AS STRING * 1

PRINT "Measuring frequency vs. time"
```

Overview

Example 7. BLOCK Transfer of vs Time Data (QuickBASIC) (Continued)

```
CODE$ = "INIT" 'Make a measurement
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))

CODE$ = "*OPC?" 'Ask for operation complete
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, OPC, 23, ACTX%) 'When OPC is 1, measurement ready

act% = 0 'Initialize actual bytes to 0
CODE$ = "DATA?" 'Ask for data
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERAB(SOURCE&, SEG RESULT(1), 900, act%, 2)
samples = act% / 2 'Number of measurements = bytes/2

CALL IOENTERS(SOURCE&, linefeed, 1, acty%) 'Reads linefeed at end of block

CODE$ = "DIM:SCAL?" 'Ask for frequency scale value
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, freqo, 24, ACTX%)
freqscale = VAL(freqo)
CODE$ = "DIM:OFFS?" 'Ask for frequency offset value
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, freqo, 24, ACTX%)
freqoffset = VAL(freqo)
CODE$ = "DIM2:OFFS?" 'Ask for time offset
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, timeo, 24, ACTX%)
timeoffset = VAL(timeo)
CODE$ = "DIM2:SCAL?" 'Ask for time scale
CALL IOOUTPUTS(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, times, 24, ACTX%)
timescale = VAL(times)

FOR i = 1 TO samples 'Result() is only pixel values
timedata(i) = i * timescale + timeoffset 'and must be converted using
IF RESULT(i) > -1 THEN 'scale and offset values.
freqdata(i) = RESULT(i) * freqscale + freqoffset
ELSE
freqdata(i) = RESULT(i)
END IF
NEXT i
```

Example 7. BLOCK Transfer of vs Time Data (QuickBASIC) (Continued)

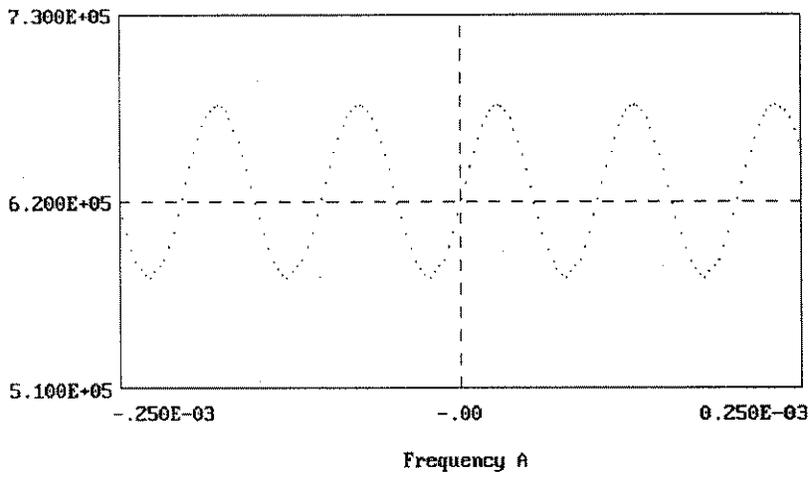
'This subprogram call labels a graph and plots the data. Note that it 'passes Result(). It is a little easier to set up and plot with integers.

```
CALL LABELIT(freqscale, freqoffset, timeoffset, timescale, samples, RESULT())

ENDPROG:
'end of program

SUB LABELIT (freqscale, freqoffset, timeoffset, timescale, samples, RESULT() AS
INTEGER)
CLS 0
SCREEN 11
VIEW (85, 70)-(536, 326)
WINDOW (1, 0)-(451, 256)
DIM i AS INTEGER
LINE (1, 256)-(451, 0), , B
LINE (1, 128)-(451, 128), , , &HFF00
LINE (226, 1)-(226, 256), , , &HFF00
FOR i = 1 TO samples
IF RESULT(i) > -1 THEN
PSET (i, RESULT(i))
END IF
NEXT i
LOCATE 24, 36
PRINT "Frequency A"
LOCATE 22, 11
PRINT USING "#.###^"; timeoffset + timescale
LOCATE 22, 62
PRINT USING "#.###^"; timeoffset + 451 * timescale
LOCATE 22, 39
PRINT USING "#.##"; timeoffset + 226 * timescale
LOCATE 21, 1
PRINT USING "##.###^"; freqoffset
LOCATE 13, 1
PRINT USING "##.###^"; (2 * freqoffset + 256 * freqscale) / 2
LOCATE 5, 1
PRINT USING "##.###^"; freqoffset + 256 * freqscale
END SUB
```

Overview



Press any key to continue

Figure D-7. BLOCK Transfer of vs Time Data (QuickBASIC).

Example 8. BLOCK Transfer of Histogram Data (QuickBASIC)

```

DECLARE SUB histogram (tintscale!, tintoffset!, numbins!, numsamples!, maxbin!,
PROBABILITY() AS DOUBLE)
'This program demonstrates the use of BLOCK transfer for floating
'point (histogram) data output on the HP 53310A. The HP 53310A will make
'a time interval measurement between Channels A and B. The program requires
'a signal connected to Channel A and B.
'A good source is the Quick Start Signal Source that is included with
'the HP 53310A. Use the Data-to-Clock jitter signal.

'To run this program you must have an HP 82335A HP-IB interface card.

REM $INCLUDE: 'QBSETUP.BAS'           'Required by HP 82335A
SOURCE& = 712                       'HP 53310A at address 12
ISC& = 7                             'Select code 7
STATE% = 1                           'Variables for some CALLs
MATCH$ = CHR$(10)
SCREEN 11                             'Screen is VGA
DIM i AS INTEGER                     'i is used in loops
CLS 0

CALL IOMATCH(ISC&, MATCH$, STATE%)   'Terminate IOENTERS on linefeed
CALL IOEOI(ISC&, STATE%)             'Make sure EOI enabled
CODE$ = "*RST"                       'Reset instrument
CALL iooutputs(SOURCE&, CODE$, LEN(CODE$))
CODE$ = "CONF:HIST:TINT"             'Configure for measurement
CALL iooutputs(SOURCE&, CODE$, LEN(CODE$))
CODE$ = "FORM BLOC"                 'Use BLOCK format
CALL iooutputs(SOURCE&, CODE$, LEN(CODE$))

INPUT "Connect a signal to Channel A and B. Press ENTER to continue", a$

DIM PROBABILITY(1 TO 450) AS DOUBLE  'Dimension for REAL data (8 bytes)
DIM tintbin(1 TO 450)                'Dimension for histogram bins
DIM OPC AS STRING * 23               'These variables are used
DIM tintsc AS STRING * 23            'throughout the program
DIM tintoff AS STRING * 23
DIM nums AS STRING * 23
DIM linefeed AS STRING * 1

PRINT "Making probability (histogram) measurement"

CODE$ = "INIT"                       'Make the measurement
CALL iooutputs(SOURCE&, CODE$, LEN(CODE$))

```

Overview

Example 8. BLOCK Transfer of Histogram Data (QuickBASIC) (Continued)

```
CODE$ = "+OPC?" 'Determine operation status
CALL iooutputs(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, OPC, 24, actx%) 'When OPC is 1, measurement ready

act% = 0
CODE$ = "DATA?" 'Ask for data. Data will each be
CALL iooutputs(SOURCE&, CODE$, LEN(CODE$)) '8 bytes.
CALL IOENTERAB(SOURCE&, SEG PROBABILITY(1), 4500, act%, 8)
numbins = act% / 8 'Total number of bins
CALL IOENTERS(SOURCE&, linefeed, 24, actz%) 'Get linefeed from end of block
' to complete transfer

CODE$ = "DIM2:SCAL?" 'Get frequency scale
CALL iooutputs(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, tintsc, 24, actx%)
tintscale = VAL(tintsc)
CODE$ = "DIM2:OFFS?" 'Get frequency offset
CALL iooutputs(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, tintoff, 24, actx%)
tintoffset = VAL(tintoff)
CODE$ = "HIST:SAMP:IN?" 'Get total number of samples
CALL iooutputs(SOURCE&, CODE$, LEN(CODE$))
CALL IOENTERS(SOURCE&, nums, 24, actx%)
numsamples = VAL(nums)

maxbin = 0 'Determine maximum value of data
FOR i = 1 TO numbins
IF PROBABILITY(i) > maxbin THEN maxbin = PROBABILITY(i)
NEXT i

'This subprogram labels a graph and plots the probability distribution

CALL histogram(tintscale, tintoffset, numbins, numsamples, maxbin, PROBABILITY())

FOR i = 1 TO numbins 'Determine the value
tintbin(i) = i * tintscale + tintoffset 'of each bin
NEXT i

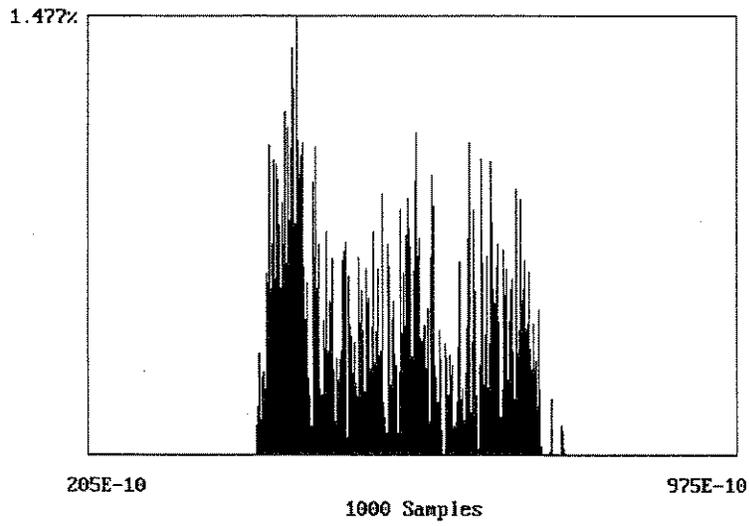
ENDPROG:
'end of program

SUB histogram (tintscale, tintoffset, numbins, numsamples, maxbin, PROBABILITY())
AS DOUBLE)
```

Example 8. BLOCK Transfer of Histogram Data (QuickBASIC) (Continued)

```
CLS 0
SCREEN 11
VIEW (70, 70)-(520, 370)
WINDOW (1, 0)-(numbins, maxbin)
DIM i AS INTEGER
LINE (1, maxbin)-(numbins, 0), , B
FOR i = 1 TO numbins
LINE (i, 0)-(i, PROBABILITY(i))
NEXT i
LOCATE 26, 31
PRINT numsamples; "Samples"
LOCATE 25, 7
PRINT USING "####^000"; tintoffset + tintscale
LOCATE 25, 59
PRINT USING "####^000"; tintoffset + 450 * tintscale
LOCATE 5, 3
B$ = "% "
PRINT USING "#.###!"; maxbin * 100; B$
END SUB
```

Overview



Press any key to continue

Figure D-8. BLOCK Transfer of Fast Histogram Data (QuickBASIC).

Overview

Example 9. Using the :STATus:OPERation Register (QuickBASIC)

'This program demonstrates the use of the Operation Status Register
'The program is set up to cause the HP 53310A to enable SRQ when a
'measurement completes.

'To run this program you must have an HP 82335A HP-IB interface card.
'You will also need an FM source connected to Channel A.
'A good source is the Quick Start Signal Source that is included with
'the HP 53310A.

```
REM $INCLUDE: 'QBSETUP.BAS'           'Required by HP 82335A
source& = 712                         'HP 53310A at address 12
ISC& = 7                               'Select code 7
STATE% = 1                            'Variable used for some CALLs
DIM OSR AS STRING * 16                'String for reading from HP 53310A

CLS 0
CALL IOEOI(ISC&, STATE%)              'Make sure EOI enabled
code$ = "**RST"                        'Reset instrument
CALL iooutputs(source&, code$, LEN(code$))
code$ = "**CLS;:STAT:PRES"
CALL iooutputs(source&, code$, LEN(code$))
code$ = "DISP:MENU FUNC"              'Display function menu
CALL iooutputs(source&, code$, LEN(code$))
INPUT "Connect an FM signal to Channel A. Press RETURN", a$
code$ = "CONF:XTIM:FREQ"              'Configure for measurement
CALL iooutputs(source&, code$, LEN(code$))
code$ = "**CLS;:STAT:PRES"            'Clear Status
CALL iooutputs(source&, code$, LEN(code$))
code$ = "STAT:OPER:ENAB 16"           'Set bit 4 of Operation Status Reg
CALL iooutputs(source&, code$, LEN(code$))
code$ = "STAT:OPER:NTR 16;PTR 0"      'Set transition filters
CALL iooutputs(source&, code$, LEN(code$))
code$ = "**SRE 128"                   'Enable SRQ on Operation Status Bit
CALL iooutputs(source&, code$, LEN(code$))
code$ = "INIT"                        'Make a measurement
CALL iooutputs(source&, code$, LEN(code$))

ON PEN GOSUB srq.received             'If SRQ, then interrupt
PEN ON                                'and process

PRIORITY% = 0
CALL IOPEN(ISC&, PRIORITY%)          'Set up branching conditions

waithere:                             'Wait in this loop for SRQ
```

Overview

Example 9. Using the :STATus:OPERation Register (QuickBASIC) (Continued)

```
IF (response% > 0) THEN                'If SRQ then
CALL iospoll(source&, response%)       'Check for another SRQ
IF response% = 0 THEN                  'If none then terminate program
PRINT "Nothing else caused an SRQ, program complete"
GOTO endprog
END IF
END IF
GOTO waithere

srq.received:
CLS
PRINT "SRQ generated by HP 53310A"
CALL iospoll(source&, response%)       'Serial Poll HP 53310A
PRINT "Status byte = "; response%      'Print the value
IF (response% AND 128) THEN            'Check for Bit 7
PRINT "SRQ caused by bit 7 of Operation Status Register"
END IF
code$ = "STAT:OPER?"                  'Check status of Operation Status
CALL icouputs(source&, code$, LEN(code$))
CALL IOENTERS(source&, OSR, 22, ACTX%)
PRINT "Operation status register value = ", VAL(OSR)
RETURN

endprog:
code$ = "**SRE 0"                      'Clear SRQ for other programs
CALL icouputs(source&, code$, LEN(code$))

'All done
```



RELATED DOCUMENTATION

In This Appendix

This appendix contains a list of documentation related to the use of the Analyzer.

Related Documentation

Additional information that you may find useful can be found in the following publications:

1. **HP 53310A Modulation Domain Analyzer Programming Quick Reference Guide**

Provides a quick reference for the Common Commands, the Subsystem Commands (including the Autoscale Group), and the Error Messages.

This guide is provided with the Analyzer.

2. **HP 53310A Modulation Domain Analyzer Quick Start Guide**

Provides a quick introduction to using the Analyzer.

This guide is provided with the Analyzer.

3. **HP 53310A Modulation Domain Analyzer Operating Reference Manual**

Provides in-depth definitions and examples of the Analyzer's capabilities.

This manual is provided with the Analyzer.

Related Documentation

4. SCPI Consortium, Standard Commands for Programmable Instruments (SCPI).

This standard is a guide for the selection of messages to be included in programmable instrumentation. It is primarily intended for instrument firmware engineers. However, you may find it useful if you are programming more than one instrument that claims conformance to the SCPI standard. You can verify the use of standard SCPI commands in different instruments.

To obtain a copy of this standard, write to :

SCPI Consortium
8380 Hercules, Suite P3
La Mesa, CA 91942

5. The International Institute of Electrical Engineers and Electronic Engineers, IEEE Standard 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation.

This standard defines the technical details required to design and build an HP-IB (IEEE 488.1) interface. This standard contains electrical specification and information on protocol that is beyond the need of most programmers. However, it can be useful to clarify formal definitions of certain terms used in related documents.

To obtain a copy of this document, write to:

The Institute of Electrical and
Electronic Engineers Inc.
345 East 47th Street
New York, NY 10017 USA

Related Documentation

6. **The International Institute of Electrical Engineers and Electronic Engineers, IEEE Standard 488.2-1987, IEEE Standard Codes, Formats, Protocols, and Common Commands for Use with ANSI/IEEE Std 488.1-1987 Programmable Instrumentation.**

This standard defines the underlying message formats and data types used in SCPI. It is intended more for firmware engineers than for instrument users/programmers. However, it can be useful if you need to know the precise definition of specific message formats, data type, or common commands.

To obtain a copy of this document, write to:

The Institute of Electrical and
Electronic Engineers Inc.
345 East 47th Street
New York, NY 10017 USA

7. **Hewlett-Packard Company, BASIC 5.0/5.1 Interfacing Techniques Vol 2., Specific Interfaces, 1987.**

This HP BASIC manual contains a good non-technical description of the HP-IB (IEEE 488.1) interface in chapter 12, The HP-IB Interface. Subsequent revisions of HP BASIC may use a slightly different title for this manual or chapter. This manual is the best reference on I/O for HP BASIC programmers.

To obtain a copy of this manual, contact your nearest Hewlett-Packard Sales office (listed at the back of the manual).

Related Documentation

8. **Hewlett-Packard Company,
Tutorial Description of the Hewlett-Packard
Interface Bus, 1987.**

To obtain a copy of this manual, contact your nearest Hewlett-Packard Sales office (listed at the back of the manual).



ERROR MESSAGES

In This Appendix

This appendix explains how to read any pending error messages from the Analyzer, discusses the types of command-language related error messages, and provides a table of all of the Analyzer's error messages and their probable causes.

Reading the Error Queue

Executing the `:SYSTEM:ERRor?` command reads the oldest message from the error queue and erases that error from the error queue. The `:SYST:ERR?` command responses to data in the form:

<error number>,"<error description string>"

An example of an error message is:

-113,"Undefined header"

Positive error numbers are specific to the Analyzer. Negative error numbers are command language-related and are discussed later in this appendix. Command language-related errors also set a corresponding bit in the Standard Event Status Register (see chapter 4, Status Reporting, for additional information).

The following short program reads all error messages (one message at a time, oldest to newest) from the error queue. After each error message is read, it is automatically erased from the error queue. When the error queue is empty, the program returns the **+0,"No Error"** message.

Error Types

```
10 ASSIGN @Anlz TO 712
20 !Assign path name
30 DIM Message$(255)
40 !Creates array for error message
50 REPEAT
60 !Repeats until error = 0
70 OUTPUT @Anlz;"SYST:ERR?"
80 !Read error number and message
90 ENTER @Anlz;Code,Message$
100 !Enter error number and message
110 PRINT Code,Message$
120 !Print error number and message
130 UNTIL Code = 0
140 END
```

NOTE

An example of how this program is incorporated into an interrupt-driven application program is shown in Chapter 4, Status Reporting.

Error Types

Error-number types are categorized as shown in *Table F-1*. Individual error messages are summarized in *Tables F-2* through *F-7*.

Execution Error

An <error number> in the range [-200 to -299] indicates that an error has been detected by the instrument's execution control block. The occurrence of any error in this class causes the execution error bit (bit 4) in the event status register to be set. One of the following events has occurred:

- A <PROGRAM DATA> element following a header was evaluated by the Analyzer as outside of its legal input range or is otherwise inconsistent with the Analyzer's capabilities.
- A valid program message could not be properly executed due to some Analyzer condition.

Execution errors are reported by the Analyzer after rounding and expression evaluation operations have taken place. Rounding a numeric data element, for example, is not reported as an execution error. Events that generate execution errors do not generate Command Errors, Analyzer-specific errors, or Query Errors (see the other error definitions in this section).

**Device-or
Analyzer-Specific
Error**

An <error number> in the range [-300 to -399] or [+1 to +32767] indicates that the instrument has detected an error that is not a command error, a query error, or an execution error; some Analyzer operations did not properly complete, possibly due to an abnormal hardware or firmware condition. These codes are also used for self-test response errors. The occurrence of any error in this class causes the Analyzer-specific error bit (bit 3) in the event status register to be set. The meaning of positive error codes is Analyzer-dependent and may be enumerated or bit mapped; the <error message> string for positive error codes is not defined by SCPI.

Error Numbers

Query Error

An <error number> in the range [-400 to -499] indicates that the output queue control of the instrument has detected a problem with the message exchange protocol. The occurrence of any error in this class should cause the query error bit (bit 2) in the event status register to be set. One of the following is true:

- An attempt is being made to read data from the output queue when no output is either present or pending.
- Data in the output queue has been lost.

Error Queue

Table F-1. Error Types

| Error Number | Error Type |
|----------------|--------------------------|
| +0 | No Errors |
| -100 to -199 | Command Errors |
| -200 to -299 | Execution Errors |
| -300 to -399 | Device-specific Errors |
| -400 to -499 | Query Errors |
| +2000 to +2011 | Analyzer-Specific Errors |

Error Queue

As errors are detected, they are placed in an error queue. This queue is first in, first out. That is, if there has been more than one error, the first one in the queue is read out with :SYST:ERR?. Subsequent responses continue until the queue is empty.

If the error queue overflows, the last error in the queue is replaced with error -350, "Too many errors". Any time the queue overflows, the least recent errors remain in the queue, and the most recent error is discarded. The length of the Analyzer's error queue is 30 (29 positions for the error messages, and 1 position for the "Too many errors" message). Reading an error from the head of the queue removes that error from the queue, and opens a position at the tail of the queue for a new error, if one is subsequently detected.

When all errors have been read from the queue, further error queries return +0, "No error".

The error queue is cleared when any of the following occur:

- Upon power up.
- Upon receipt of a *CLS command.
- Upon reading the last item from the queue.

Error Numbers

The system-defined error numbers are chosen on an enumerated ("1 of N") basis. The SCPI-defined error numbers and the <error description> portions of the ERRor query response are listed here. The first error described in each class (for example, -100, -200, -300, -400) is a "generic" error. In selecting the proper Error Number to report, more specific error codes are preferred, and the generic error is used only if the others are inappropriate.

No Error

This message indicates that the Analyzer has no errors.

Command Error

An <error number> in the range [-100 to -199] indicates that an IEEE 488.2 syntax error has been detected by the instrument's parser. The occurrence of any error in this class causes the command error bit (bit 5) in the event status register to be set. One of the following events has occurred:

- An IEEE 488.2 syntax error has been detected by the parser. That is, a controller-to-Analyzer message was received that is in violation of the IEEE 488.2 standard. Possible violations include a data element that violates the Analyzer listening formats or whose type is unacceptable to the Analyzer.
- An unrecognized header was received. Unrecognized headers include incorrect Analyzer-specific headers and incorrect or unimplemented IEEE 488.2 common commands.
- A Group Execute Trigger (GET) was entered into the input buffer inside of an IEEE 488.2 PROGRAM MESSAGE.

Events that generate command errors do not generate execution errors, Analyzer-specific errors, or query errors (see the other error definitions in this chapter).

Error Numbers

Table F-2. Error Messages

| CODE | MESSAGE | CAUSE |
|------|-----------------------------|--|
| +0 | No error | The error queue is empty. Every error in the queue has been read (SYSTEM:ERRor? query) or the queue was cleared by power-on or *CLS. |
| -100 | Command error | This is the generic syntax error used if the Analyzer cannot detect more specific errors. |
| -101 | Invalid character | A syntactic element contains a character that is invalid for that type. For example, a header containing an ampersand, :INP:COUP& AC. |
| -102 | Syntax error | An unrecognized command or data type was encountered. |
| -103 | Invalid separator | The parser was expecting a separator and encountered an illegal character. |
| -104 | Data type error | The parser recognized a data element different than one allowed. For example, numeric or string data was expected but block data was received. |
| -105 | GET not allowed | A Group Execute Trigger was received within a program message. |
| -108 | Parameter not allowed | More parameters were received than expected for the header. |
| -109 | Missing parameter | Fewer parameters were received than required for the header. |
| -112 | Program mnemonic too long | The header or character data element contains more than twelve characters. |
| -113 | Undefined header | The header is syntactically correct, but it is undefined for the Analyzer. For example, *XYZ is not defined for the Analyzer. |
| -121 | Invalid character in number | An invalid character for the data type being parsed was encountered. For example, a "9" in octal data. |
| -123 | Exponent too large | Number is too large or too small to be represented internally. |
| -124 | Too many digits | The mantissa of a decimal numeric data element contained more than 255 digits excluding leading zeros. |
| -128 | Numeric data not allowed | A legal numeric data element was received, but the Analyzer does not accept one in this position for the header. |
| -131 | Invalid suffix | The suffix does not follow the syntax described in IEEE 488.2 or the suffix is inappropriate for the Analyzer. |
| -138 | Suffix not allowed | A suffix was encountered after a numeric element that does not allow suffixes. |
| -141 | Invalid character data | Either the character data element contains an invalid character or the particular element received is not valid for the header. |
| -148 | Character data not allowed | A legal character data element was encountered where prohibited by the Analyzer. |
| -150 | String data error | This error can be generated when parsing a string data element. This particular error message is used if the Analyzer cannot detect a more specific error. |
| -151 | Invalid string data | A string data element was expected, but was invalid for some reason. For example, an END message was received before the terminal quote character. |
| -158 | String data not allowed | A string data element was encountered but was not allowed by the Analyzer at this point in parsing. |
| -160 | Block data error | This error can be generated when parsing a block data element. This particular error message is used if the Analyzer cannot detect a more specific error. |
| -168 | Block data not allowed | A legal block data element was encountered but was not allowed by the Analyzer at this point in parsing. |

Error Numbers

Table F-2. Error Messages (Continued)

| | | |
|------|---|--|
| -170 | Expression error | This error can be generated when parsing an expression data element. It is used if the Analyzer cannot detect a more specific error. |
| -178 | Expression data not allowed | Expression data was encountered but was not allowed by the Analyzer at this point in parsing. |
| -200 | Execution error | This is the generic syntax error if the Analyzer cannot detect more specific errors. This code indicates only that an Execution Error has occurred. |
| -210 | Trigger error | Used if the Analyzer cannot detect a more specific error from the :INIT, :TRIG, or :ABOR subsystems. |
| -211 | Trigger ignored | Indicates that a GET or *TRG was received and recognized by the Analyzer but was ignored because of Analyzer timing considerations. For example, the Analyzer was not ready to respond. |
| -213 | Init ignored | Indicates that a request for a measurement initiation was ignored as another measurement was in progress. |
| -220 | Parameter error | Indicates that a program data element related error occurred. This error message is used when the Analyzer cannot detect more specific errors. |
| -221 | Settings conflict | Indicates that a legal program data element was parsed but could not be executed due to the current Analyzer state. |
| -222 | Data out of range | Indicates that a legal program data element was parsed but could not be executed because the interpreted value is outside the legal range defined by the Analyzer. Typically, the value is clipped to legal limit. |
| -223 | Too much data | Indicates that a legal program data element of block, expression, or string type was received that contained more data than the Analyzer could handle due to memory or related Analyzer-specific requirements. |
| -224 | Illegal parameter value | Used where exact value, from a list of possible values, was expected. |
| -230 | Data corrupt or stale | No valid data available. New measurements started but not completed. |
| -240 | Hardware error | Indicates that a legal program command or query could not be executed because of a hardware problem in the Analyzer. |
| -241 | Hardware missing | Indicates that a legal program command or query could not be executed because of missing Analyzer hardware. For example, the C-channel option was not installed. |
| -300 | Device-specific error | This is the generic device-dependent error. |
| -310 | System error | Indicates that a system error occurred. |
| -330 | Self-test failed | Indicates at least one failure occurred when *TST? was executed. |
| -331 | Self-test failed; CPU kernel failure | Power on self-test detected this hardware failure. |
| -332 | Self-test failed; ROM checksum failure | Power on self-test detected this hardware failure. |
| -333 | Self-test failed; RAM address lines failure | Power on self-test detected this hardware failure. |
| -334 | Self-test failed; Display RAM address lines failure | Power on self-test detected this hardware failure. |
| -335 | Self-test failed; NVRAM address lines failure | Power on self-test detected this hardware failure. |
| -336 | Self-test failed; CPU DTACK failure | Power on self-test detected this hardware failure. |
| -337 | Self-test failed; Main Board DTACK failure | Power on self-test detected this hardware failure. |

Error Numbers

Table F-2. Error Messages (Continued)

| | | |
|-------|---|---|
| -338 | Self-test failed; Interrupt failure | Power on self-test detected this hardware failure. |
| -339 | Power on interpolator calibration failure | Power on self-test detected this hardware failure. |
| -350 | Queue Overflow | Indicates that there is no room in the error queue and an error occurred but was not recorded. |
| -400 | Query error | This is the generic query error. |
| -410 | Query INTERRUPTED | Indicates that a condition causing an INTERRUPTED Query error occurred. For example, a query followed by DAB or GET before a response was completely sent. |
| -420 | Query UNTERMINATED | Indicates that a condition causing an UNTERMINATED Query error occurred. For example, the Analyzer was addressed to talk and an incomplete program message was received. |
| -430 | Query DEADLOCKED | Indicates that a condition causing a DEADLOCKED Query error occurred. For example, both input buffer and output buffer are full and the Analyzer cannot continue. |
| -440 | Query UNTERMINATED after indefinite response | Indicates that a query was received in the same program message after a query requesting an indefinite response (indefinite length arbitrary block response or arbitrary ASCII response) was executed. |
| +2000 | Offset calibration on A failed | :DIAGnostic:CALibration:INPut:OFFSet:AUTO ONCE failed. |
| +2001 | Offset calibration on B failed | :DIAGnostic:CALibration:INPut2:OFFSet:AUTO ONCE failed. |
| +2002 | Gain calibration on A failed | :DIAGnostic:CALibration:INPut:GAIN:AUTO ONCE failed. |
| +2003 | Gain calibration on B failed | :DIAGnostic:CALibration:INPut2:GAIN:AUTO ONCE failed. |
| +2004 | Interpolator calibration failed | :DIAGnostic:CALibration:INTerp:AUTO ONCE, *CAL? failed. |
| +2005 | Oscillator error | Oscillator error detected. |
| +2006 | Hardware failure | Unusual hardware condition detected. |
| +2007 | Recall setup failed | *RCL <NRf> failed. |
| +2008 | Unable to auto configure the frequency or time interval range setup | One of the following commands failed: [:SENSe]:FREQuency:RANGE[1 2 3]:CENTer:AUTO ONCE, [:SENSe]:TINTerval:RANGE:CENTer:AUTO ONCE, [:SENSe]:FREQuency:RANGe[1 2 3]:AUTO ONCE, [:SENSe]:TINTerval:RANGE:AUTO ONCE, :CONFigure (partially), or :MEASure (partially). |
| +2009 | No signal found on A | One of the following commands failed: [:SENSe]:EVENT:LEVel:AUTO ONCE, :CONFigure (partially), or :MEASure (partially). |
| +2010 | No signal found on B | One of the following commands failed: [:SENSe]:EVENT2:LEVel:AUTO ONCE, :CONFigure (partially), or :MEASure (partially). |
| +2011 | Analysis function failed | Queried analysis result could not be computed. |

Error Numbers

GLOSSARY

Auto Trigger

The trigger mode which first attempts to display the measurement data using the specified frequency, time interval, or external edge trigger conditions, however, if these trigger conditions are not met, the Analyzer will disregard the trigger conditions and force the display of measurement data.

Dynamic Range

The AC voltage amplitude range of the input signal that the Analyzer is specified to perform measurements on.

Histogram

A bar graph whose vertical axis represents the probability in each bin and whose horizontal axis represents the range of measurement values segmented into uniform bin widths. The Analyzer performs two types of histogram processing: one from the data of the "vs. Time" display, and one directly from the raw data of the input signal (Fast Histogram).

Hysteresis

The voltage "window" centered around the voltage threshold, that an input signal must pass through to be detected by the Analyzer.

Least Significant Digit

The smallest incremental value displayed in a measurement.

Panorama Display

The smaller of the two display areas at the top of the CRT display (when Panorama is on) which allows viewing of consecutive sweeps of measurement data.

Resolution

The smallest variation of the input quantity (frequency or time interval) that the Analyzer can determine. Resolution is limited by random uncertainties and is therefore specified on an rms basis rather than a peak value.

Sample

A sample consists of two quantities: a cumulative count of voltage threshold crossings, referenced from the first threshold crossing of a sweep, and the precise time at which the threshold crossing occurred.

Sample Interval

The time between two adjacent samples of an input signal.

For frequency measurements up to the maximum measurement rate, the Analyzer will time every voltage threshold crossing. For frequency measurements greater than the maximum measurement rate, the Analyzer will time every *n*th voltage threshold crossing at up to the maximum measurement rate. The Analyzer specifies the sample interval at the center frequency of the CRT display, referred to as "Interval At Center."

Sampling

Sampling is the process of counting and timing the voltage threshold crossings of an input signal, and controls the rate at which this process is performed.

Sensitivity

The minimum voltage amplitude of an input signal which the Analyzer can perform a frequency measurement on.

Signal Operating Range

The voltage amplitude range of an input signal, with DC offset, that the Analyzer is specified to perform measurements on.

Sweep

A continuous series of measurements which is displayed as a single update or pass on the CRT.

Trigger

The frequency, time interval, or external edge conditions which initiate the display of measurement data.

Trigger HF Reject

A feature which controls whether high frequency modulation components from the measurement data are allowed to trigger a measurement sweep.

Voltage Threshold

The voltage level that an input signal must pass through to be detected by the Analyzer. It is also the center of the hysteresis "window."

Window Display

Activated only when Panorama is on. The Analyzer stores successive sweeps in memory and allows viewing of all sweeps through the panorama display and viewing of one sweep in the window or regular time variation display.

4 Glossary

Index

A

Address, HP-IB B-2
Addressing B-17
analysis functions 2-32
analysis range 2-33, 2-37
Analyzer-Specific Error Messages F-6
Applications 1-4
ASSIGN Statement 1-55
Assumptions 1-6
Autoscale Group 2-7
 :CONFigure 2-7
 :FETCh 2-7
 :MEASure 2-7
 :READ 2-7
Autoscale Group Commands 1-1, 1-3
Autoscale Vs :SENSe Measurements 2-9

B

Boolean parameters 1-60, 5-4

C

Command Error F-4
Command Types C-6
Commands Used to Transfer Data 5-8
Commands Used to Transfer
 Measured Data Values 5-9
Common Command Syntax 5-4
Common Commands 1-2, C-6
 *CAL? (Calibration) 3-4
 *CLS (Clear Status) 3-5
 *ESE (Event Status Enable) 3-6
 *ESR? (Event Status Register Query) 3-8
 *IDN? (Identification Query) 3-10
 *OPC (Operation Complete) 3-11
 *OPC? (Operation Complete) 3-12
 *OPT? (Identification Query) 3-13
 *RCL (Recall) 3-14
 *RST (Reset) 3-15
 *SAV (Save) 3-16

*SRE (Service Request Enable) 3-17
*STB? (Status Byte Query) 3-19
*TRG (Trigger) 3-21
*TST? (Self-Test Query) 3-22
*WAI (Wait to Continue) 3-23
Common Commands Summary Table 1-22
Condition Register 4-22
Conditions for Remote to Local B-17
Connecting the Analyzer to a computer A-1
Connecting to a computer A-1
Connecting to a printer A-5
Controller B-7
Controller, HP-IB B-3

D

Data Messages B-19
Data Transfer Overview 5-1
Definite-Length Block Response Data 5-8
Description of Transferred Data 5-11
Device Clear B-7
Device Clear message B-18
Device-Dependent command B-2
Device-Dependent commands B-8
Device-Independent command B-2
Device-Independent commands B-8
Device-Specific Error F-5

E

ENTER Statement 1-56
Error Numbers F-4
Error Queue 4-16, F-3
Error Types F-2
Event Enable Register 4-24
Event Register 4-23
Execution Error F-5
external computer 1-54

F

front panel interface status B-7
Front Panel To SCPI Command Mapping 1-23

G

GET (Group Execute Trigger message) B-18
Getting Started 1-1
Group Execute Trigger message B-18

H

How to Interpret Transferred Data 5-13
How to Report Status Events 4-5
How to Use This Manual 1-2
HP BASIC (5.0) language 1-55
HP-IB address B-2
HP-IB Address Selection B-8
HP-IB cable connection
 Cable connection, HP-IB A-1
HP-IB Controller B-3
HP-IB default conditions B-16
HP-IB description B-2
HP-IB operating modes
 Operating modes, HP-IB A-5

I

Initialization 1-57
Interface capabilities B-4
Interface Commands B-8
Interface functions B-4
Interface system terms B-2
Interfacing to an External Computer 1-54
Introduction To SCPI Programming 1-54

L

Learning to Program the Analyzer 1-2
Listener B-4, B-6
Literal parameters 1-60, 5-4
Local Lockout 1-58, B-16
Local mode B-7, B-15
Local Mode versus Remote Mode 1-58
Local-Only Capabilities 1-58

M

Major Interface Functions B-3
Manual Contents 1-4
Meta Messages B-11
Multiple Queries 5-7

N

Nrf parameters 1-60, 5-4
Numeric value parameters 1-60, 5-4

O

Optional Keywords 1-59
Optional Numeric Keyword Suffixes 1-60
Output Queue 4-18
OUTPUT Statement 1-56

P

Parallel Poll B-7
Parallel polling B-3
Parameter types 1-60
Polling, parallel B-3, B-7
Polling, serial B-3
Program and Response Messages C-4
 program data
 suffixes 1-62
 Program message parameter types 1-60, 5-4
 Program message syntax 1-3, 5-2
 Program Messages C-4
 Programming Languages 1-55

Q

Query Error F-6
Query parameters 1-60

R

Reading the Error Queue 4-17, F-1
Reading the Output Queue 4-18
Receiving the Abort message B-19
Receiving the data message B-18
Related Documentation E-1
Remote Mode 1-58, B-7

| | |
|-----------------------------|------|
| Remote Operation | B-15 |
| Remote-Only Capabilities | 1-58 |
| Remote-to-Local Changes | 1-58 |
| Response Message Data Types | 5-6 |
| Response Message Syntax | 5-5 |
| Response Messages | C-4 |
| Root keyword | 1-59 |

S

| | |
|----------------------------------|-------------|
| SCPI Commands | 1-2 |
| SCPI Conformance Information | 1-5 |
| SCPI Overview | C-1 |
| SCPI Status Register Model | 4-21 |
| SCPI Summary | 1-59 |
| Selected Device Clear message | B-18 |
| Sending data messages | B-18 |
| Serial polling | B-3 |
| Service request | B-7 |
| Service Request Enable Register | 4-13 |
| Setting the HP-IB Address | A-6 |
| Setting the HP-IB Operating Mode | A-6 |
| SRQ | B-7 |
| Standard Event Status Register | 4-15 |
| Standard Status Data Structure | 4-9 |
| Status Byte Register | 4-11 |
| Status Data Structures | 4-9 |
| Status Reporting Overview | 4-1 |
| string parameters | 1-60, 5-4 |
| Subsystem Command Syntax | 5-3 |
| Subsystem Commands | 1-3, C-6 |
| :INPut | 2-87 |
| :ABORt | 2-5 |
| :CALCulate | 2-31 |
| :CLEAr | 2-57 |
| :CONFigure | 2-15 |
| :DIAGnostic | 2-59 |
| :DISPlay | 2-65 |
| :FETCh | 2-20 |
| :FORMat | 2-79 |
| :INITiate | 2-83 |
| :MARKer | 2-91 |
| :MEASure | 2-23 |
| :READ | 2-28 |
| :STATus | 2-173, 4-24 |
| :SYSTem | 2-187 |
| :TRIGger | 2-197 |

| | |
|----------------------------------|-------|
| [[:SENSe]] | 2-103 |
| Subsystem Commands Summary Table | 1-5 |
| Suffix Elements | 1-62 |
| Suffix Multipliers | 1-62 |
| Suffixes | 1-62 |
| Switching from Local to Remote | B-15 |
| Switching from Remote to Local | B-16 |
| syntax diagram | |
| Common Commands | 3-1 |

T

| | |
|-----------------------------------|---------------|
| TALK | B-6 |
| Talk Only | B-3, B-6, B-8 |
| Talker | B-3, B-6 |
| Transferring Data to the Computer | 5-6 |
| Transition Filter | 4-23 |

U

| | |
|-------------------------|------|
| Using | |
| :CONFigure | 2-13 |
| :MEASure | 2-12 |
| :SENSe | 2-14 |
| Using Multiple Commands | 1-61 |



For more information, call your local HP sales office listed in your telephone directory or an HP regional office listed below for the location of your nearest sales office.

United States of America:

Rockville, MD
(301) 670 4300

Rolling Meadows, IL
(708) 255 9800

Fullerton, CA
(714) 999 6700

Atlanta, GA
(404) 980 7351

Canada:
(416) 678 9430

Japan:
(8113) 3335 8192

Latin America:
Mexico
(525) 202 0155

Brazil
(11) 709 1444

Australia/New Zealand:
(03) 895 2895

Far East:
Hong Kong
(852) 848 7070

Korea
(2) 769 0800

Taiwan
(2) 717 9524

Singapore
(65) 291 8554

India
(11) 690 355

PRC
(1) 505-3888

In Europe, Africa and Middle East,
please call your local HP sales office
or representative:

Austria/South East Area:
(0222) 2500-0

Belgium and Luxembourg:
(02) 778 31 11

Denmark:
45 99 10 00

Finland:
(90) 88 721

France:
(1) 69.82.65.00

Germany:
(06172) 16 0

Greece:
(01) 68 28 811

Ireland:
(01) 2844633

Israel:
(03) 5380 333

Italy:
(02) 92 122 241

Netherlands:
(020) 547 6669

Norway:
(02) 87 97 00

Portugal:
(11) 301 73 30

South Africa:
(011) 806 1000

Spain:
900 123 123

Sweden:
(08) 750 20 00

Switzerland:
(057) 31 21 11

Turkey:
(90-1) 4 125 83 13

U.K.:
(0344) 362 867

For countries not listed, contact:
Hewlett-Packard
International Sales Branch
Geneva, Switzerland
Tel: +41-22-780-7111
Fax: +41-22-780-7535

Printed in U.S.A.
Data subject to change.
Rev. 11/92