Programmer’s Guide

HP ESA-L1500A Spectrum Analyzer

© Copyright Hewlett-Packard Company 1997. All Rights Reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws. 1400 Fountaingrove Parkway, Santa Rosa CA, 95403-1799, USA
The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

The following safety symbols are used throughout this manual. Familiarize yourself with the symbols and their meaning before operating this instrument.

**CAUTION**

*Caution* denotes a hazard. It calls attention to a procedure that, if not correctly performed or adhered to, would result in damage to or destruction of the instrument. Do not proceed beyond a caution until the indicated conditions are fully understood and met.

**WARNING:**

*Warning* denotes a hazard. It calls attention to a procedure which, if not correctly performed or adhered to, could result in injury or loss of life. Do not proceed beyond a warning note until the indicated conditions are fully understood and met.

**WARNING:**

This is a *Safety Class 1 Product (provided with a protective earthing ground incorporated in the power cord)*. The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. Any interruption of the protective conductor inside or outside of the product is likely to make the product dangerous. Intentional interruption is prohibited.
1 Preparing for Use and Printing

Connecting Your Spectrum Analyzer 16

Connecting the HP-IB to the Computer 18

Connecting the RS-232 Interface to an HP Vectra or IBM Compatible Personal Computer 20

If There Is a Problem 22

Printing 23

What is HP-IB? 27

What is the RS-232 Bus 28

2 Programming Basics

What’s In This Chapter 36

Writing Your First Program 37

Modifying the Program 41

Enhancing the Program with Variables 44

Getting Data from the Spectrum Analyzer 46

3 Programming Examples

What You’ll Find in This Chapter 50

Reading Trace Data Into a Variable 51

Saving Trace Data in a computer 52
Contents

Command Syntax Conventions 124
Programming Command Descriptions 131
ALIGN Self-alignment Routines 132
AMPCOR Amplitude Correction 135
ANNOT Annotation 138
APB Trace A Plus Trace B 140
AT Attenuation 141
ATC Transfer Trace A to Trace C 143
AUNITS Amplitude Units 144
AUTO Auto Couple 146
AXB Exchange Trace A and Trace B 148
BAUDRATE Baud Rate of Spectrum Analyzer 150
BLANK Blank Trace 152
BML Trace B Minus Display Line 153
BRIGHT Display Brightness 154
BTC Transfer Trace B to Trace C 155
BXC Trace B Exchange Trace C 156
CAT Catalog 158
CF Center Frequency 161
CLRAVG Clear Average 163
Contents

CLRW Clear Write 164
CLS Clear Status Byte 165
CMDERRQ Command Error Query 166
CONFIG Configuration 167
CONTRAST Display Contrast 169
CONTS Continuous Sweep 170
CORREK Correction Factors On 171
DATASTAT Data Status 172
DATEMODE Date Mode 174
DEFCONFIG Default Configuration 175
DELETE File 176
DEMOD Demodulation 178
DEMODT Demodulation Time (Dwell Time) 180
DET Detection Mode 182
DL Display Line 184
DN Down 186
DONE Done 187
ERASE Erase 189
EXITSHOWSYS Exit Config/Show System Screen 190
FA Start Frequency 191
Contents

FB Stop Frequency 193
FOFFSET Frequency Offset 195
FREF Frequency Reference 197
FS Full Span 198
GETPRNT Get Print 199
GRAT Graticule 202
HD Hold Data Entry 204
ID Identify 205
INVERTLCD Inverse Video 206
INZ Input Impedance 208
IP Instrument Preset 210
LG Logarithmic Scale 213
LIMIDEL Delete limit line Table 215
LIMIDISP Limit Line Display 216
LIMIFAIL Limits Failed 218
**LIMIFT** Select Frequency or Time Limit Line 220
LIMIHALF Select Upper or Lower Limit Line 221
LIMILINE Limit Lines 223
LIMIREL Relative Limit Lines 226
LIMISEG Enter limit line Segment for Frequency 228
Contents

LIMISEGT Enter limit line Segment for Sweep Time  231
LIMITEST Enable Limit Line Testing  234
LN Linear Scale  236
LOAD Load  237
LSPAN Last Span  239
MDS Measurement Data Size  240
MDU Measurement Data Units  243
MEAN Trace Mean  246
MEANTH Trace Mean Above Threshold  248
MF Marker Frequency Output  250
MINH Minimum Hold  252
MKA Marker Amplitude  253
MKACT Activate Marker  255
MKBW Marker Bandwidth  257
MKCF Marker to Center Frequency  259
MKD Marker Delta  260
MKF Marker Frequency  262
MKFA Marker to Start Frequency  265
MKFB Marker to Stop Frequency  266
MKFC Marker Counter  267
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MKFCR Marker Counter Resolution</td>
</tr>
<tr>
<td>MKMIN Marker Minimum</td>
</tr>
<tr>
<td>MKN Marker Normal</td>
</tr>
<tr>
<td>MKNOISE Marker Noise</td>
</tr>
<tr>
<td>MKOFF Marker Off</td>
</tr>
<tr>
<td>MKP Marker Position</td>
</tr>
<tr>
<td>MKPAUSE Marker Pause</td>
</tr>
<tr>
<td>MKPK Marker Peak</td>
</tr>
<tr>
<td>MKPP Marker Peak-to-Peak</td>
</tr>
<tr>
<td>MKPX Marker Peak Excursion</td>
</tr>
<tr>
<td>MKREAD Marker Readout</td>
</tr>
<tr>
<td>MKRL Marker to Reference Level</td>
</tr>
<tr>
<td>MKSP Marker to Span</td>
</tr>
<tr>
<td>MKSS Marker to Step Size</td>
</tr>
<tr>
<td>MKTBL Marker Table</td>
</tr>
<tr>
<td>MKTH Peak Threshold</td>
</tr>
<tr>
<td>MKTRACE Marker Trace</td>
</tr>
<tr>
<td>MKTRACK Marker Track</td>
</tr>
<tr>
<td>ML Mixer Level</td>
</tr>
<tr>
<td>MOV Move</td>
</tr>
</tbody>
</table>
Contents

MXMH Maximum Hold 304
NDB Number of dB 305
NDBPNT dB Points 307
NDBPNTR dB Points Bandwidth 310
NORMLIZE Normalize Trace Data 311
NRL Normalized Reference Level 313
NRPOS Normalized Reference Position 315
OL Output Learn String 317
PCTAM Percent AM 319
PCTAMR Percent AM Response 321
PEAKS Peaks 323
POWERON Power-On State 327
PREAMPG External Preamplifier Gain 328
PRINT Print 329
PRNPRT Printer Port 331
PRNTADRS Printer Address 332
PRNTMARB Printer Margin Bottom 333
PRNTMARGT Printer Margin Top 335
PRNTRES Printer Resolution 337
PSTATE Protect State 338
Contents

PWRBW Power Bandwidth 339
PWRUPTIME Power Up Time 342
RB Resolution Bandwidth 343
REV Revision 345
RFCALIB 50 MHz Signal 347
RL Reference Level 349
RMS Root Mean Square Value 351
ROFFSET Reference Level Offset 353
RQS Service Request Mask 355
SAVE Save File 358
SEGDEL Segment Delete 361
SER Serial Number 363
SETDATE Set Date 364
SETTIME Set Time 365
SHOWSYS Show System 366
SMOOTH Smooth Trace 367
SNGLS Single Sweep 370
SP span 371
SPEAKER Speaker 373
SPZOOM Span Zoom 374
Contents

SQR Square Root 375
SRCAT Source Attenuator 377
SRCPOFS Source Power Offset 379
SRCPSTP Source Power-Level Step Size 381
SRCPSWP Source Power Sweep 383
SRCPWR Source Power 385
SRQ Force Service Request 387
SS Center Frequency Step Size 391
ST Sweep Time 393
STB Status Byte Query 395
STDEV Standard Deviation of Trace Amplitudes 397
SUM Sum of Trace Amplitudes 399
SUMSQR Sum of Squared Trace Amplitudes 401
SWPCPL Sweep Couple 403
TDF Trace Data Format 404
TH Threshold 411
TIMEBASEC Timebase Coarse Adjustment 413
TIMEBASEF Timebase Fine Adjustment 415
TIMEDATE Time Date 417
TIMEDSP Time Display 419
Contents

TITLE Title 420
TM Trigger Mode 421
TOI Third-Order Intermodulation Measurement 423
TOIR Third-Order Intermodulation Response 426
TRA/TRB/TRC Trace Data Input and Output 428
TRSTAT Trace Status 432
TS Take Sweep 434
UP Up 436
VARIANCE Variance of Trace Amplitudes 437
VAVG Video Average 439
VB Video Bandwidth 441
VBR Video Bandwidth Ratio 443
VIEW View Trace 444
XCH Exchange 445

6 Error Messages
Descriptions 448

Index 455
Preparing for Use and Printing
Connecting Your Spectrum Analyzer

These topics are covered in the following section:

- Connecting the Hewlett-Packard Interface Bus (HP-IB) to the computer
- Connecting the RS-232 interface to your computer
- Connecting your spectrum analyzer to a printer
- If there is a problem
- Information about HP-IB and RS-232

Connecting Your Spectrum Analyzer to a Computer

The spectrum analyzer works with many popular computers. However, the steps required to connect your spectrum analyzer to a specific computer depend on the computer you are using. Before turning to the interconnection instructions for your computer, please read the following general information.

Configuring Your Computer System

Every computer system has a specific configuration. Your system configuration might include a printer or an external disk drive. Whenever you add another piece of equipment (for example, your spectrum analyzer), you may need to reconfigure your computer system so that the computer knows where and how to send information to the newly added device.

Some computers do not require configuring when a spectrum analyzer is connected. Others require a simple modification. The most common modification is changing the configuration information stored on the computer’s operating system disk. A few computers require the insertion of an add-on board, or “card.” Refer to your computer documentation if your system needs these modifications.

All of the test programs for HP-IB and RS-232 interfaces are written using the BASIC language. If you have never entered or run a BASIC program, refer to your computer documentation.

The Test Program

To test the system configuration, a simple test program is provided for each computer listed. After you have connected your computer and spectrum analyzer, you should enter and run the test program on your computer to make sure the com-
puter is sending instructions to the spectrum analyzer through the interface cable. If the interface is working and the program is entered correctly, a statement is displayed on the computer screen.

**NOTE:**
The listed computer and spectrum analyzer equipment includes the minimum components necessary to establish communication between your spectrum analyzer and computer. If you are using application software, check with your software supplier for specific computer hardware and memory requirements.

**NOTE:**
Using an interface cable other than the one listed with your computer’s interconnection instructions may prevent proper communication between the spectrum analyzer and computer.

Pressing the spectrum analyzer’s **System (Local)** key removes it from remote mode and enables front panel control.

**For the HP-IB Interface**
Refer to the end of this chapter for a detailed description of the HP-IB interface. It also contains instructions for connecting the spectrum analyzer’s HP-IB interface to several different computers. If your computer is not listed, but it supports an HP-IB interface, there is a good possibility that it can be connected to the spectrum analyzer. Consult your computer documentation to determine how to connect external devices on the bus.

**For the RS-232 Interface**
Refer to the end of this chapter for a detailed description of the **RS-232** interface. It also contains instructions for connecting the spectrum analyzer’s **RS-232** interface to several different computers. If your computer is not listed, but it supports a standard **RS-232** interface, there is a good possibility that the spectrum analyzer may be connected to the computer. Consult your computer documentation to determine how to connect external devices to your computer’s **RS-232** connector.

There are two types of **RS-232** devices: data terminal equipment (DTE) and data communication equipment (DCE). Types of DTE devices include display terminals. DCE equipment includes modems and, generally, other computer **RS-232** devices. The spectrum analyzer **RS-232** port is the DTE-type. Connections from the computer (DCE) to the spectrum analyzer (DTE) are shown.
Connecting the HP-IB to the Computer

**Equipment**
- HP ESA-L1500A spectrum analyzer with option A4H
- HP 10833A (or equivalent) HP-IB cable
- Computer:
  - HP Series 300 technical computer (with RMB)
  - HP workstation (with RMB)
  - HP Vectra or IBM compatible pc (with HP BASIC for Windows)

**Interconnection Instructions**
Connect the spectrum analyzer to the computer using the HP-IB cable. Figure 1 shows the spectrum analyzer interface board.

![Figure 1: Connecting the Spectrum Analyzer HP-IB to a Computer](image)

1. **The** parallel connector is for printing only.
2. Connect to the computer using the HP-IB cable.
NOTE: The spectrum analyzer rear panel slots 1 and 2 can both be used for the remote interface board. The analyzer will search for an interface board in slot 1 before looking in slot 2.

Test Program
To test the connection between the computer and the spectrum analyzer, turn on your spectrum analyzer and follow the instructions below.

1. To use the test program below, you will need to have BASIC available on your computer. Consult your BASIC manual for further information on loading BASIC on your system.

2. Check the HP-IB address of the spectrum analyzer: press System, Remote Port HP-IB. The usual address for the spectrum analyzer is 18. If necessary, reset the address of the spectrum analyzer to 18 (or select an appropriate address) and press Enter.

3. Enter the following test program and run it. The program shows that the computer is able to send instructions to, and read information from, the spectrum analyzer. If you need help entering and running the program, refer to your computer and software documentation.

   10 PRINTER IS 1
   20 Analyzer=718
   30 CLEAR Analyzer
   40 OUTPUT Analyzer;"IP;SNGLS;"
   50 OUTPUT Analyzer;"CF 300MZ;TS;"
   60 OUTPUT Analyzer;"CF?;"
   70 ENTER Analyzer;A
   80 PRINT "CENTER FREQUENCY = ";A;"Hz";
   90 END

The program tells the spectrum analyzer to perform an instrument preset and enter single-sweep mode. Next, the program sets the center frequency to 300 MHz and takes a sweep.

The program then queries the center frequency value and tells the computer to display CENTER FREQUENCY = 3.0E+8 Hz.

If the computer does not display the center frequency, see “If There Is a Problem” at the end of this chapter.
Connecting the RS-232 Interface to an HP Vectra or IBM Compatible Personal Computer

Equipment
- HP ESA-L1500A spectrum analyzer with Option 1AX
- HP 245426 9-pin to 25-pin RS-232 cable for computers with a 25-pin female RS-232 port
- HP 24542U9-pin to 9-pin RS-232 cable for computers with a 9-pin female RS-232 port
- Computer:
  HP Vectra or IBM compatible PC (with Microsoft QuickBasic for Windows)

NOTE: Refer to the section at the end of this chapter for information on cable wiring.

Interconnection Instructions
Connect the spectrum analyzer to the computer using the RS-232 cable. Figure 2 shows the spectrum analyzer interface.

![Figure 2](image)

Connecting the Personal Computer to the Spectrum Analyzer

1. The parallel connector is for printing only.
2. Connect to the computer using the RS-232 cable.
NOTE: The spectrum analyzer rear panel slots 1 and 2 can both be used for the remote interface board. If there are boards in both slots, the analyzer will only use the board in slot 1.

Test Program

To test the connection between the computer and the spectrum analyzer, turn on your spectrum analyzer and follow the instructions below.

1. To use the test program below, you will need to have Microsoft QuickBasic available on your computer. Load the QuickBasic and specify a communications buffer of 4096 bytes. (Consult your QuickBasic manual for further information on loading QuickBasic on your system.)

2. Set the spectrum analyzer baud rate to 9600, to match the baud rate set up for the computer port in the test program. In line 20, the “9600” indicates 9600 baud for the computer port. Press the following keys to set the baud rate: System, Remote Port (serial), Baud Rate, 9600, Enter.

3. Enter the following test program and run it. The program shows that the computer is able to send instructions to, and read information from, the spectrum analyzer. If you need help entering and running the program, refer to your computer and software documentation.

```
10 !File = TESTPGM
20 OPEN "COM1:9600,N,8,1" AS #1
30 PRINT #1,"IP-"
40 PRINT #1,"SNchLS-"
50 PRINT #1,"CF 30;MZ;TS;"
60 PRINT #1,"CF?;"
70 INPUT #1,CEN
80 PRINT,"CENTER FREQ = ";CEN;"Hz"
90 END
```

The program tells the spectrum analyzer to perform an instrument preset and enter single sweep mode. Next, the program sets the center frequency to 300 MHz and takes a sweep. The program then queries the center frequency value and tells the computer to display CENTER FREQUENCY = 3.0E+8 Hz.

If the computer does not display the center frequency, see “If There Is a Problem” at the end of this chapter.
If There Is a Problem

This section offers suggestions to help get your computer and spectrum analyzer working as a system. The test programs provided in this chapter let you know if the connection between the computer and the spectrum analyzer is working properly. The analyzer Esc key can be used to abort the printing process.

If the test program does not run, try the following suggestions:

1. Check your program for errors.
2. You may need to modify the program syntax to work with your computer. Refer to your software manual for correct syntax.
3. The program must be executed correctly. Refer to your computer manual for information about program execution.

If the test program runs on the computer, but the spectrum analyzer does not respond, try the following suggestions:

1. Make sure the spectrum analyzer is turned on. If the spectrum analyzer has power, the fan should be running.
2. Make sure the interface cable is connected securely. Check the interface cable for defects. Make sure the correct cable is used.
3. If you are using an HP-IB interface, the spectrum analyzer must be set to the correct address. Press System, Remote Port.
4. If you are using the RS-232 interface, check the spectrum analyzer baud rate. Refer to the section at the end of this chapter for information about setting the baud rate on the spectrum analyzer.
5. If you wish to reset the spectrum analyzer configuration to the state it was in when it was originally shipped from the factory, load the defaults. Press System, More, Default Config.

If you suspect your computer is causing the problems, check it by running a program that you know works. If your system still has problems, contact the sales and service office nearest you. Your salesperson will either be able to help solve the problem or refer you to someone who can.
Printing

You may wish to obtain a permanent record of data displayed on the spectrum analyzer screen. This can be done using the Print key of the spectrum analyzer, and a printer.

Printing Using a Parallel Interface

Equipment

- HP ESA-L1500A spectrum analyzer with option 1AX or A4H
- HP C2950A parallel printer cable
- Printer with parallel interface. (Some models support color printing.)

Interconnection and Printing Instructions

1. Turn off the printer and the spectrum analyzer.
2. If your printer has configuration switches, now is the time to set them. See Figure 3 or refer to your printer’s documentation for more specific information on your printer’s configuration.

Figure 3 Parallel Printer Switch Settings
Preparing for Use and Printing

Printing

3 Connect the printer to the spectrum analyzer using the parallel printer cable.
4 Turn on the spectrum analyzer and printer.
5 On the spectrum analyzer, press **System, Printer, Port, Parallel** (so that Parallel is underlined).
6 Choose your printer from those available in the **System, Printer, Select Printer** menu. If your printer is not listed, you may be able to use it by selecting **User Defined** and making the appropriate choices. Refer to the user’s guide for more information about defining a printer.
7 If you want the softkey labels to be printed with the spectrum analyzer display printout, press **System, Printer, Print Softkeys On Off** so that On is underlined.
8 Press Print.

**Printing Using an RS-232 Interface**

**Equipment**
- HP ESA-L1500A spectrum analyzer with option 1AX
- One of the following cables:
  - HP 245426 9-pin to 25-pin RS-232 cable
  - HP 24542U 9-pin (f) to 9-pin (f) RS-232 cable
  - HP C2932A 9-pin (f) to 9-pin (m) RS-232 cable (for use with LaserJet 4P and 4Plus)
- Printer with **RS-232** Interface. (Some models support color printing.)

**Interconnection and Printing Instructions**

1 Turn off the spectrum analyzer and the printer.

**NOTE:**

The **RS-232** interface allows only one device to be connected to the spectrum analyzer. Refer to the section at the end of the chapter for more information on **RS-232** protocol and cable wiring.

2 Figure 4 shows examples of some of the printer’s configuration switches set up for 9600 baud. See your printer’s documentation for more specific information.
Preparing for Use and Printing

Printing

Figure 4  9600 Baud Settings for Serial Printers

3 Connect the printer using an RS-232 cable.
4 Turn on the spectrum analyzer and printer.
5 Choose your printer from those available in the analyzer System, Printer, Select Printer menu. If your printer is not listed you may be able to use it by selecting User Defined and making the appropriate choices. Refer to the user’s guide for more information about defining a printer.
6 Press System, Printer, Port, Serial (so that Serial is underlined).
7 To set the baud rate to 9600 baud, press Baud Rate, 9600, Enter.
8 If you want the softkey labels to be printed with the spectrum analyzer display printout, press System, Printer, Print Softkeys On Off so that On is underlined.
9 Press Print.

Printing Using an HP-IB Interface

Equipment

- HP ESA-L1500A spectrum analyzer with option A4H
- HP 10833A (or equivalent) HP-IB cable
- Printer with HP-IB Interface. (Some models support color printing.)

---

**Figure 4** 9600 Baud Settings for Serial Printers
Preparing for Use and Printing

Printing

Interconnection and Printing Instructions

1. Turn off the printer and the spectrum analyzer.

2. Connect the printer to the spectrum analyzer using the HP-IB cable. The printer is usually set to device address 1. The printer’s mode switches must be set correctly for use with the spectrum analyzer. Refer to Figure 5 for the correct settings for using an HP 2225A ThinkJet printer.

![ThinkJet Printer Switch Settings](image)

**Figure 5**

**ThinkJet Printer Switch Settings**

*NOTE:* Because the spectrum analyzer cannot print with two controllers (the computer and the spectrum analyzer) connected, the computer must be disconnected from the HP-IB before printing. In most cases, it is sufficient to simply turn the computer OFF.

3. Turn on the spectrum analyzer and printer.

4. On the spectrum analyzer, press **System, Printer, Port**, (so that HP-IB is underlined).

5. The printer is usually set to device address 1. To enter address 1 for the printer, press **1, Enter**.

6. Choose your printer from those available in the **System, Printer, Select Printer** menu. If your printer is not listed you may still be able to use it by selecting **User Defined** and making the appropriate choices. Refer to the user’s and calibration guide for more information about a user defined printer.

7. If you want the **softkey** labels to be printed with the spectrum analyzer display printout, press **System, Printer, Print Softkeys On Off** (so that On is underlined).

8. Press **Print**.
What is HP-IB?

Your Option A4H spectrum analyzer has a Hewlett-Packard Interface Bus (HP-IB) connector on the rear panel, as shown in Figure 6.

![Figure 6: HP-IB Connector](image)

**Item 2** is the HP-IB connector. The HP-IB system utilizes a party-line bus structure. Devices such as the spectrum analyzer are connected on the party line with HP-IB cables. A computer gives instructions and is the “controller.” The spectrum analyzer takes orders and is the “listener.” The spectrum analyzer is also capable of transmitting data over the party line. Devices that transmit data back to the computer are “talkers.”

Each device on the party line has an address. Device addresses are used by the controller to specify who talks and who listens. A device’s address is usually set at the factory.

The number preceding the device’s address (for example, the number 7 when Analyzer=718), signifies that the HP-IB interface is selected.

When you turn on the spectrum analyzer, the HP-IB address appears on the screen (for example, HP-IB **ADRS**: 18). If necessary, you can reset the address of the spectrum analyzer by pressing **System, Remote Port, HP-IB**, entering in the address number using the front-panel number keys, then pressing **Enter**. You may use any address between 0 and 30 which is not already being used in your system. (Usually, 1 is reserved for printers, and 21 is reserved for controllers.)
What is the RS-232 Bus

Your option 1 AX spectrum analyzer has an RS-232 serial connector on the rear panel, as shown in Figure 7. This section contains information pertaining to RS-232 signals, cable connections, and baud rate.

Figure 7

RS-232 Connector

The RS-232 interface utilizes serial data transmission. Data is sent, one bit at a time, in groups of 10 to 12 data-bits.

Two devices, such as the spectrum analyzer and a computer, can exchange commands and data over the RS-232 connection. This interface uses two serial data lines and five handshaking lines. Handshaking signals are required for full hardware control of the information exchange. It is possible to use a three wire connection, in some situations (typically at a low “baud”, or data rate).

Another parameter for the RS-232 interface is the “baud”, or data rate. This is the speed at which the computer and spectrum analyzer exchange data. The baud rate of each of the two RS-232 devices must be the same.

The RS-232 Data Lines

RS-232 uses serial data transmission, meaning that data is transmitted one bit at a time. There are two data lines carrying signals:

- Transmit data (TxD) – the serial data output. This line is connected to the RxD input line.
- Receive data (RxD) - the serial data input. This line is connected to the TxD output line.
The RS-232 Handshaking Lines

In addition to the data signals, there are five other signal lines (called handshaking lines), used to control the flow of data. Listed below are the handshake signal descriptions:

- Request to send (RTS) — Output signal indicates that the spectrum analyzer is ready to communicate. This line is true at power-up and stays true while power is on.
- Clear to send (CTS) — Input signal indicates that the external controller is ready to receive data.
- Data terminal ready (DTR) — Output signal from the spectrum analyzer. When the input buffer is full, this line goes false.
- Data set ready (DSR) — Is not available.
- Data carrier detect (DCD) — Input to the spectrum analyzer. If DCD is true, the spectrum analyzer will receive data from the controller. If false, no data will be input. The data will be ignored.

The spectrum analyzer checks its CTS input before transmitting data to the computer. If the CTS line is false, the spectrum analyzer will not transmit data. The spectrum analyzer transmits data when the CTS line is true.

The spectrum analyzer sets the DTR line (PC CTS) false when its input buffer is full.

Baud Rate

The speed at which data is exchanged is called the baud rate or data rate. This is usually expressed in baud or bits per second. Common baud rates are 1200 and 9600. It is critical that the baud rate of the spectrum analyzer be the same as that of the printer.

If you need to change the baud rate, refer to the “Setting the Spectrum Analyzer Baud Rate” in this section.

Protocol

The RS-232 protocol is as follows:

- Baud rate 300 to 57,000 baud
- 8 bits per character
- 1 stop bit
- No parity
- Software handshake — none
- Xon/Xoff and ENQ/ACK not supported by the spectrum analyzer
Preparing for Use and Printing

What is the RS-232 Bus

When BREAK is issued to the spectrum analyzer, the following occurs:

1. The present command is aborted
2. The input buffer is cleared
3. The output buffer is cleared
4. All trace output is stopped
5. The command parser is reinitialized

BREAK does not perform any of the following:

- Invoke instrument preset
- Clear illegal command off screen

The RTS signal goes true on power-up and does not go false during any communication. It stays true while power is on.

Figure 8 and Figure 9 lists the signal connections between a personal computer and the spectrum analyzer.

**Figure 8** HP 245426 25-pin to 9-pin Full Handshaking Connection
**Figures 9 and 10**

**ThinkJet Printer Connections**

To connect an HP ThinkJet printer to the spectrum analyzer, use the information in Figure 10 which describes the wiring configuration. Information about the printer mode switch settings is indicated in the following tables. See the ThinkJet printer manual for more information.
### Table 1
**Setting of Thinkjet Printer Mode Switches**

<table>
<thead>
<tr>
<th>Switch Number</th>
<th>Setting</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>down</td>
<td>Printer performs a carriage return only.</td>
</tr>
<tr>
<td>2</td>
<td>down</td>
<td>Printer perform a line feed only.</td>
</tr>
<tr>
<td>3</td>
<td>up</td>
<td>Sets the printer to skip paper perforations.</td>
</tr>
<tr>
<td>4</td>
<td>down</td>
<td>Sets the printer for a paper length of 11 inches.</td>
</tr>
<tr>
<td>5</td>
<td>down</td>
<td>Sets the printer to HP MODE.</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>down</td>
<td>Sets the printer to USASCII.</td>
</tr>
<tr>
<td>8</td>
<td>down</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2
**Setting of RS-232 Switches**

<table>
<thead>
<tr>
<th>Switch Number</th>
<th>Setting</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>up</td>
<td>DTR.</td>
</tr>
<tr>
<td>2</td>
<td>down</td>
<td>no parity, 8 bits.</td>
</tr>
<tr>
<td>3</td>
<td>down</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>down</td>
<td>9600 baud.</td>
</tr>
<tr>
<td>5</td>
<td>down</td>
<td></td>
</tr>
</tbody>
</table>

### Table 3
**Setting the Baud Rate**

<table>
<thead>
<tr>
<th>Switch Number</th>
<th>Baud Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1200</td>
</tr>
<tr>
<td>4</td>
<td>up</td>
</tr>
<tr>
<td>5</td>
<td>up</td>
</tr>
</tbody>
</table>
Connecting a Laser Jet Printer

To connect an HP LaserJet printer to the spectrum analyzer, use the information in Figure 11.

<table>
<thead>
<tr>
<th>LaserJet 4P/4MP</th>
<th>Analyzer</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTS 1</td>
<td>1 DCD</td>
</tr>
<tr>
<td>TxO 2</td>
<td>2 RxO</td>
</tr>
<tr>
<td>RxO 3</td>
<td>3 TxO</td>
</tr>
<tr>
<td>DSR 4</td>
<td>4 DTR</td>
</tr>
<tr>
<td>GND 5</td>
<td>5 GND</td>
</tr>
<tr>
<td>DTR 6</td>
<td>6 DSR</td>
</tr>
<tr>
<td>nc 7</td>
<td>---optional--- 7 RTS</td>
</tr>
<tr>
<td>DTR 8</td>
<td>8 CTS</td>
</tr>
<tr>
<td>nc 9</td>
<td>---optional--- 7 RI</td>
</tr>
</tbody>
</table>

Figure 11  HP C2932A 9-pin to 9-pin LaserJet Printer Connection

Connecting a Modem

To connect a modem to the spectrum analyzer for remote queries, use the information in Figure 12. The connection is for a Hayes 1200 Modem and the spectrum analyzer.

<table>
<thead>
<tr>
<th>Modem</th>
<th>Analyzer</th>
</tr>
</thead>
<tbody>
<tr>
<td>TxO 2</td>
<td>3 TxO</td>
</tr>
<tr>
<td>RxO 3</td>
<td>2 RxO</td>
</tr>
<tr>
<td>RTS 4</td>
<td>7 RTS</td>
</tr>
<tr>
<td>CTS 5</td>
<td>8 CTS</td>
</tr>
<tr>
<td>DSR 6</td>
<td>6 DSR</td>
</tr>
<tr>
<td>GND 7</td>
<td>5 GND</td>
</tr>
<tr>
<td>DCD 8</td>
<td>1 DCD</td>
</tr>
<tr>
<td>DTR 20</td>
<td>4 DTR</td>
</tr>
<tr>
<td>RI 22</td>
<td>no connection 9 RI</td>
</tr>
</tbody>
</table>

Figure 12  25-pin to 9-pin Modem Connection

Select 1200 baud for both the modem and the spectrum analyzer.
Preparing for Use and Printing

What is the RS-232 Bus

**Setting the Spectrum Analyzer Baud Rate**

The baud rates of the spectrum analyzer and the personal computer must be the same. For example, to set the spectrum analyzer to 9600 baud, use the following procedure:

1. Press the **System, Remote Port** (with Serial selected.)

2. **Press the keys:** 9600, **Enter**. To set the **baud rate** to 1200 baud, press these keys: 1200, **Enter**.
Programming Basics
What’s In This Chapter

- Writing your first program
- Modifying your program
- Enhancing your program
- Getting data from the spectrum analyzer

If the computer is not connected to the spectrum analyzer, follow the instructions in Chapter 1.

A general knowledge of the BASIC programming language and the spectrum analyzer is recommended before reading this chapter. Refer to your software documentation manuals for more information about BASIC. Chapter 5 of this manual provides information about the spectrum analyzer commands in alphabetical order.

**NOTE:** All programming examples in this chapter for the HP-IB interface are written in HP BASIC for Windows, using an HP Vectra PC. For the RS-232 interface, examples are written in Microsoft Quick BASIC, using an HP Vectra personal computer or compatible controller.
When the spectrum analyzer has been connected to a computer via HP-IB or RS-232 interface, the computer can be used to send instructions to the spectrum analyzer. These instructions tell the spectrum analyzer such things as frequency, span, resolution bandwidth, and sweep mode. If a properly selected sequence of instructions is sent to the spectrum analyzer, a measurement is made. Sequences of coded instructions are called programs.

Composing the Program

Most spectrum analyzer programs contain several common statements, or “commands,” that address the spectrum analyzer, preset it, and select its sweep mode. As an example, we will write a short program that executes only these common commands.

The following programs are for the HP-IB and the RS-232 interfaces. Note the quotation marks that contain spectrum analyzer commands in each line. Also note the semicolons at the end of each line, inserted at the end of each set of spectrum analyzer commands within the quotation marks. Using semicolons makes programs easier to read, prevents command misinterpretation, and is recommended by IEEE Standard 728.

**NOTE:**

In commands where quotation marks occur, the computer recognizes data as character data and not BASIC programming language commands.

Program Example for the HP-IB Interface

```
05 !File:  "IBPROGl"
10 Analyzer=718
20 CLEAR Analyzer
30 OUTPUT Analyzer;"IP;"
40 OUTPUT Analyzer;"SNGLS;TS;"
50 LOCAL 7
60 END
```

Line 10 of our program assigns a variable called “Analyzer” to our spectrum analyzer at address 718. This instruction is followed by the HP BASIC CLEAR command, which resets the spectrum analyzer on the HP-IB. With these two program lines, we have set up a clear communication path between the computer and the spectrum analyzer.
Writing Your First Program

Line 30 introduces the instrument preset (IP) command, which corresponds to the Preset key on the spectrum analyzer. The IP command sets all of the analog parameters of the spectrum analyzer to known values and provides a good starting point for every measurement.

**NOTE:**

All softkey functions on the spectrum analyzer have corresponding programming commands. As you continue programming, you will learn the command names that correspond to the front-panel keys and softkeys.

Line 40 activates the single-sweep mode. Most remotely controlled measurements require control of the sweep. Once SNGLS has activated the single-sweep mode, take sweep (TS) starts and completes one full sweep. TS maintains absolute control over the sweep, which is necessary for accurate computer data transfer and reduced program execution time.

Before we end the program, we return the spectrum analyzer to front-panel control with line 50, `LOCAL 7`. The LOCAL command corresponds to the System (Local) key on the front panel of the spectrum analyzer. (`LOCAL 7` commands everything on the bus to go to local mode.)

Finally, in line 60, we end the program with the END command. (If you forget to include the END command, the computer will give an error message.)

Enter the program lines, press RUN on the computer, and watch the spectrum analyzer display as it completes each instruction.

**Program Example for the RS-232 Interface**

```
10 'File = 232PROGl
20 OPEN "COM1: 9600,N,8,1" AS #1
30 PRINT #1,"IP-
40 PRINT #1,"SNGLS;TS;
50 END
```

Line 20 of the program opens the RS-232 COM1: line, identifies it as #1, and sets the RS-232 parameters as follows:

- 9600 baud
- no parity
- 8 bits/character
- 1 stop bit

Line 30 of the program introduces the instrument preset (IP) command, which corresponds to the Preset key on the spectrum analyzer. The IP command sets all of the analog parameters of the spectrum analyzer to known values and provides a good starting point for every measurement.
NOTE: Most softkey functions on the spectrum analyzer have corresponding programming commands. As you continue programming, you will learn the command names that correspond to the front-panel keys and softkeys.

Line 40 activates the single-sweep mode. Most remotely controlled measurements require control of the sweep. Once SNGLS has activated the single-sweep mode, take sweep (TS) starts and completes one full sweep. The TS command maintains absolute control over the sweep, which is necessary for accurate computer data transfer and reduced program execution time. Finally, in line 50, end the program with the END command.

Make sure that the spectrum analyzer baud rate is 9600 using the Baud Rate softkey. Enter the program lines, then press RUN on the computer. Watch the spectrum analyzer display as it completes each instruction.

Programming Guidelines

The following steps should be used to generate a spectrum analyzer program:

1. Perform the measurement manually, keeping track of the sequence of functions used.
2. In the written program, execute an instrument preset (IP) and set single-sweep mode (SNGLS) before setting other spectrum analyzer functions.
3. Use variables for function values. List variables at the beginning of the program.
5. After setting spectrum analyzer functions, execute a take sweep (TS) command before reading data or activating markers.
6. The spectrum analyzer can return only one value per programming line. Do not have more than one query per programming line.
7. Use the exclamation point (!) to include comment lines when using HP BASIC for Windows. Use the apostrophe (’) or REM to create comment lines when using Microsoft Quick BASIC. The use of the exclamation point and the apostrophe to create comment lines are dependent on the controller and the programming language (not interface-dependent) and may be different for your system.

File Naming Rules

File names for storing states, traces, limit lines or amplitude correction data files in the analyzer should follow the conventions as indicated below:

- They can be up to eight characters long. In addition, they can have a file extension up to three characters long. The analyzer assigns the extension.
- They are not case sensitive. It does not matter whether you use upper case or lower case letters when you type them.
- They can contain only the letters A through Z, the number 0 through 9, and the following special characters:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_</td>
<td>underscore</td>
</tr>
<tr>
<td>^</td>
<td>carat</td>
</tr>
<tr>
<td>$</td>
<td>dollar sign</td>
</tr>
<tr>
<td>~</td>
<td>tilde</td>
</tr>
<tr>
<td>!</td>
<td>exclamation point</td>
</tr>
<tr>
<td>#</td>
<td>number sign</td>
</tr>
<tr>
<td>%</td>
<td>percent sign</td>
</tr>
<tr>
<td>&amp;</td>
<td>ampersand</td>
</tr>
<tr>
<td>-</td>
<td>hyphen</td>
</tr>
<tr>
<td>{}</td>
<td>braces</td>
</tr>
<tr>
<td>@</td>
<td>at sign</td>
</tr>
<tr>
<td>'</td>
<td>single quotation mark</td>
</tr>
<tr>
<td>`</td>
<td>apostrophe</td>
</tr>
<tr>
<td>0</td>
<td>parenthesis</td>
</tr>
</tbody>
</table>

No other characters are valid.
- They cannot contain spaces, commas, backslashes, or periods. (except the period that separates the name from the extension.)
- They cannot be identical to the name of another file in the same directory.
Modifying the Program

Remote operation of the spectrum analyzer is similar to manual operation. Remote measurements are executed by commands that correspond to front-panel keys and softkeys.

The spectrum analyzer user’s and calibration guide shows you how to make a simple measurement. We can add instructions to our program so that it will make the same measurement. Because the manual process closely resembles that of the program, you may want to review the some of the measurements in the user’s and calibration guide.

By inserting a few lines into the initial program, we can set functions such as the center frequency and span, and we can activate a marker to find a signal’s frequency and amplitude.

Program Example for the HP-IB Interface

This example requires a 300 MHz signal at the input of the spectrum analyzer.

First, we set the center frequency to 300 MHz. The CF command corresponds to the center frequency function, Center Freq. (All spectrum analyzer commands, such as CF, are described in Chapter 5.)

Insert the following program line between lines 40 and 50:

```
41 OUTPUT Analyzer;“CF 300MZ;”
```

Next, we set the span to 200 MHz with the SP command. Add the following program line:

```
42 OUTPUT Analyzer;“SP 200MZ;”
```

Because we are controlling the sweep, we must update the spectrum analyzer display screen with the following program line:

```
43 OUTPUT Analyzer;“TS;”
```

When the program is executed, the spectrum analyzer takes one full sweep before executing line 41. Line 41 changes the center frequency to 300 MHz, and line 42 changes the span to 200 MHz.

Enter the following program line to place a marker at the highest peak on the trace with a MKPK HI command:
Programming Basics
Modifying the Program

The completed program is shown below:

```
05   !File: "IBPROG2"
10  Analyzer=718
20  CLEAR Analyzer
30  OUTPUT Analyzer;"IP;"
40  OUTPUT Analyzer;"SNGLS;TS;"
41  OUTPUT Analyzer;"CF 300MZ;"
42  OUTPUT Analyzer;"SP 200MZ;"
43  OUTPUT Analyzer;"TS;"
44  OUTPUT Analyzer;"MKPK HI;"
50   LOCAL 7
60   END
```

Run the program to make the measurement. Watch the spectrum analyzer display as it completes each instruction. Notice that the program executes the instructions faster than is possible from the front panel.

When a certain measurement is repeated often, a computer program can save time. In addition, the computer is less likely to make an error than an operator manually entering the same instructions from the front panel.

**Program Example for the RS-232 Interface**

*NOTE:* This example requires a 300 MHz signal at the input of the spectrum analyzer.

First, we set the center frequency to 300 MHz. The CF command corresponds to the center frequency function, **Center Freq.** (All spectrum analyzer commands, such as CF, are described in Chapter 5.)

Insert the following program lines between lines 40 and 50 of the previous program.

```
41 PRINT #1,"CF 300MZ;"
```

Next, set the span to 200 MHz with the SP command. Add the following program line:

```
42 PRINT #1,"SP 200MZ;"
```

Because we are controlling the sweep, we must update the spectrum analyzer display with the following program line:

```
43 PRINT #1,"TS;"
```
When the program is executed, the spectrum analyzer takes one full sweep before executing line 41. Line 41 changes the center frequency to 300 MHz. Line 42 changes the span to 200 MHz.

Enter the following program line to place a marker at the highest peak on the trace:

```
44 PRINT #1,"MKPK HI;"
```

The completed program is shown below:

```
10 'File = 232PROG2
20 OPEN "COM1:9600,N,8,1" AS #1
30 PRINT #1,"IP;"
40 PRINT #1,"SNGLS;TS;"
41 PRINT #1,"CF 300MZ;"
42 PRINT #1,"SP 200MZ;"
43 PRINT #1,"TS;"
44 PRINT #1,"MKPK HI;"
50 END
```

Run the program to make the measurement. Watch the spectrum analyzer display as it completes each instruction. When a certain measurement is repeated often, a computer program can save time. Also, the computer is much less likely to make an error than an operator manually entering the same instructions from the front panel.
Enhancing the Program with Variables

In the last program, specific center frequency and span values were set. By modifying the program, we can cause different values to be set each time the program is run.

Program Example for the HP-IB Interface

In the following program, the exclamation point (!) allows the words that follow to be ignored by the computer. Thus, they serve as comments in the program.

```
10 !FILE: "VAR10"
20 REAL C_freq, S_pan  !define the variables
30 Analyzer = 718
40 CLEAR Analyzer
50 OUTPUT Analyzer; "IP;SNGLS;TS;"
60 !ask for the desired center frequency:
70 INPUT "CENTER FREQUENCY (MHz)?", C_freq
80 !ask for the desired span:
90 INPUT "SPAN (MHz)?", S_pan
100 !send the center frequency and span to the analyzer and take a sweep to update the analyzer screen:
110 !analyze
120 OUTPUT Analyzer; "CF "; C_freq; "MZ;"
130 OUTPUT Analyzer; "SP "; S_pan; "MZ;"
140 OUTPUT Analyzer; "TS;"
150 OUTPUT Analyzer; "MKPK HI;"
160 !find the signal peak with peak search:
170 OUTPUT Analyzer; "MKPK HI;"
180 LOCAL 7
190 END
```

Three modifications are made to a previous programming example so it includes center frequency and span variables. First, using the HP BASIC REAL command, we define two variables, \texttt{C\_freq} and \texttt{S\_pan}. The frequency and span parameters are stored in these variables. (Refer to line 20.)

Second, using the HP BASIC INPUT command, we prompt the user to enter the desired center frequency and span. The center frequency and span values are entered on the computer; because the measurement units will be entered by the program, the user does not enter them. (See lines 70 to 140.)

Third, we modify the output parameter statements so that the values stored in \texttt{C\_freq} and \texttt{S\_pan} are sent to the spectrum analyzer. (See lines 130 to 140.)

A sweep is taken after the parameters are sent to the spectrum analyzer, to ensure that the spectrum analyzer screen is updated before the marker is placed on the highest signal peak.
Program Example for the RS-232 Interface

In the following program, the apostrophe (‘) allows the words that follow to be ignored by the computer. Thus, they serve as comments in the program.

```
10 'File = 232PROG3
20 OPEN "COM1:9600,N,8,1" AS #1
30 PRINT #1,"IP-"
40 PRINT #1,"SNGLS;TS;"
50 'Ask for the center frequency and span
60 PRINT "INPUT THE CENTER FREQUENCY (MZ) ";
70 INPUT CENTER
80 PRINT "INPUT THE SPAN (MZ) ";
90 INPUT SPAN
100 'Send center freq and span to spectrum analyzer
110 'take a sweep to update screen
120 PRINT #1,"CF ";CENTER;"MZ;"
130 PRINT #1,"SP ";SPAN;"MZ;"
140 PRINT #1,"TS;"
150 'find the signal peak with peak search
160 PRINT #1,"MKPK HI;"
170 END
```

Three modifications are made to an earlier programming example in order to include center frequency and span variables. First, we use two variables, CENTER and SPAN, to store the frequency and span parameters.

Second, using the BASIC INPUT command, we ask the computer operator to enter the desired center frequency and span. (See lines 70 and 90 in previous program example). Next, we modify the output parameter statements so that the values stored in the CENTER and SPAN variables are sent to the spectrum analyzer. (See lines 120 and 130.) A sweep is taken after the parameters are sent to the spectrum analyzer, to ensure that the spectrum analyzer screen is updated before the marker is placed on the highest signal peak.
Getting Data from the Spectrum Analyzer

This section demonstrates a technique for getting information out of the spectrum analyzer. In an earlier program in this chapter, we placed a marker at the highest peak of a trace and the value of the marker could be read in the upper right-hand corner of the spectrum analyzer display. In the following program, we will add some commands that will read the marker’s frequency and amplitude value and return those values to the computer.

Program Example for the HP-IB Interface

This example requires a 300 MHz signal at the input of the spectrum analyzer.

```plaintext
10 !FILE: "MKR"
20 REAL A_mpmarker, F_reqmarker !define variables
30 Analyzer = 718
40 OUTPUT Analyzer ; "IP; "
50 !set the output format of the spectrum analyzer for
60 !real numbers:
70 OUTPUT Analyzer ; "TDF P; "
80 !set the spectrum analyzer parameters:
90 OUTPUT Analyzer ; "SNGLS;"
100 OUTPUT Analyzer ; "CF 300MZ;"
110 OUTPUT Analyzer ; "SP 200MZ;"
120 OUTPUT Analyzer ; "TS;"
130 OUTPUT Analyzer ; "MKPK HI;"
140 !ask the spectrum analyzer for the marker's
150 !amplitude value:
160 OUTPUT Analyzer ; "MKA?;"
170 !send the amplitude value to the computer:
180 ENTER Analyzer; A_mpmarker
190 !ask the spectrum analyzer for the marker's
200 !frequency value:
210 OUTPUT Analyzer ; "MKF?;"
220 !send the frequency value to the computer:
230 ENTER Analyzer; F_reqmarker
240 !print the amplitude and frequency:
250 PRINT "THE SIGNAL PEAK IS "; A_mpmarker,
260 PRINT " dBm AT ";F_reqmarker / l.E+6; " MHz
270 !set the spectrum analyzer to continuous sweep mode:
280 OUTPUT Analyzer; "CONTS;"
290 LOCAL 7
300 END
```

First, using the HP BASIC REAL command, we define two variables, A_mpmarker and F_reqmarker. The amplitude and frequency values of the marker are stored in these variables. (See line 20.)
Second, we set the output format of the spectrum analyzer for real numbers with the spectrum analyzer’s trace data format (TDF) command. (See line 70.) As in our original program, we set the center frequency and span values. A sweep is taken and the marker is placed on the trace.

Next, we ask the spectrum analyzer for the amplitude value of the marker. We have the spectrum analyzer send the marker amplitude value to the computer. Note that there can be only one spectrum analyzer query per programming line. We also ask the spectrum analyzer for the frequency value of the marker, and we have the spectrum analyzer send the marker frequency value to the computer. (See lines 140 through 230.)

Finally, we print the values on the computer screen:

"THE SIGNAL PEAK IS . . . dBm AT . . . MHz"

Before we end the program, we return the spectrum analyzer to continuous-sweep mode and local control.

Program Example for the RS-232 Interface

This example requires a 300 MHz signal at the input of the spectrum analyzer.

```
10 'File = 232PROG4
20 OPEN "COM1:9600,N,8,1" AS #1
30 PRINT #1 "IP;"
40 'Set the output format of the spectrum analyzer for real numbers
50 'Set the spectrum analyzer's parameters
60 PRINT #1,"TDF P-"
70 'set the spectrum analyzer's parameters
80 PRINT #1,"SNGLS;"
90 PRINT #1,"CF 300MZ;"
100 PRINT #1,"SP 200MZ;"
110 PRINT #1,"TS;"
120 PRINT #1,"MKPK HI;"
130 'ask the spectrum analyzer for the marker's amplitude value
140 'get the frequency value from the spectrum analyzer
150 PRINT #1, "MKF?;"
160 'ask the spectrum analyzer for the marker's frequency value
170 INPUT #1,"AMPMARKER"
180 PRINT #1,"MKF?;"
190 INPUT #1,"FREQMARKER"
200 'get the frequency value from the spectrum analyzer
210 PRINT #1, "FREQMARKER;"
220 'print the amplitude and frequency
230 PRINT "THE SIGNAL PEAK IS ";AMPMARKER;
240 PRINT " dBm AT ";FREQMARKER/1000000!; "MZ"
250 'set the spectrum analyzer to continuous sweep mode
260 PRINT #1,"CONTS;"
270 END
```
Programming Basics

Getting Data from the Spectrum Analyzer

First, set the output format of the spectrum analyzer to real numbers with the spectrum analyzer’s trace data format (TDF) command (line 60).

As in the original program, the center frequency and span values are set and a sweep is taken. Next, the marker is placed on the trace.

Two additional variables are used for AMPMARKER and FREQMARKER. The amplitude and frequency values of the marker are stored in these variables (lines 170 and 210). The program requests the amplitude and frequency values of the marker (lines 150 and 190). Note that there can be only one spectrum analyzer query per programming line.

Finally, the amplitude and frequency values are displayed on the computer screen:

"THE SIGNAL PEAK IS . . . dBm AT . . . MZ"
What You’ll Find in This Chapter

- Reading Trace Data Into a Variable on page 51
- Saving Trace Data in a computer on page 52
- Loading Trace Data from a Computer Disk on page 54
- Different Formats for Trace Data Transfers on page 55
- Saving and Loading Instrument States on page 66
- Returning the Spectrum Analyzer to its Former State on page 68
- Measuring Harmonic Distortion on page 69
- Monitoring System Operation on page 72
- Using Markers on page 79
- Using Limit Lines on page 81
- Measure Signal Bandwidth on page 82
- Measuring Noise on page 83
- Using Amplitude Correction on page 84
- Examples Using the RS-232 Interface on page 85

NOTE:
The following programming examples are written in HP BASIC using the HP-IB interface. (Use HP BASIC for windows on the PC or RMB on a UNIX workstation.) See Examples Using the RS-232 Interface on page 85, for examples using Quick BASIC on the RS-232 interface.

General knowledge of the BASIC programming language is recommended before reading this chapter. (Refer to your software documentation manuals.) Chapter 5 of this manual defines spectrum analyzer commands alphabetically.
Reading Trace Data Into a Variable

The following program reads a trace from the spectrum analyzer and stores the trace data in a variable.

**Program Example:**

```
10 ! Program reads in trace data and stores it in a variable
20 !
30 ! Create a 401 point trace array
40 REAL Trace_a(1:401)
50 ASSIGN @Sa TO 718
60 ! Set the output format of the spectrum analyzer for real numbers
70 !
80 OUTPUT @Sa;"TDF P:"
90 ! Set the spectrum analyzer parameters
100 OUTPUT @Sa;"IP;RFCALIB ON;SNGLS:" ! Turns 50 MHz sig on
110 OUTPUT @Sa;"CF 50 MZ;SP 10 MZ;TS;" ! Set up and take sweep
120 OUTPUT @Sa;"MKPK HI:" ! Marker to peak of response
130 ! Move peak to center of spectrum analyzer screen
140 OUTPUT @Sa;"MKCF;TS;"
150 ! Ask the spectrum analyzer for trace data
160 OUTPUT @Sa;"TRA?;"
170 ! Send the trace data to the computer
180 ENTER @Sa;Trace_a(*)
190 PRINT Trace_a(*)
200 END
```

Trace data can be read with the computer. We will start with a program created in Chapter 2. We modify the program to create a **401-point** trace array, called Trace-a, in which the trace data will be stored. The program uses the TRA command to request trace A data. (The MKA and MKF commands from the previous program have been deleted.) The spectrum analyzer then sends trace A data to the variable, Trace-a.
Saving Trace Data in a computer

The trace data in the previous program can be stored on a computer disk by making a few program modifications.

Program Example:

10 ! Reads in trace data and stores it in a variable
20 !
30 REAL Trace_a(1:401)! Create 401 point array
40 ASSIGN @Sa TO 718
50 ! Set the output format of the spectrum analyzer for real numbers
60 OUTPUT @Sa;"TDF P;"
70 ! Set the spectrum analyzer parameters
80 OUTPUT @Sa;"IP;RFCALIB ON;SNGLS;"
90 OUTPUT @Sa;"CF 50 MZ;SP 10 MZ;TS;"
100 OUTPUT @Sa;"MKPK HI;"
110 ! Move peak to center of spectrum analyzer screen
120 OUTPUT @Sa;"MKCF;"
130 OUTPUT @Sa;"TS;"
140 ! Ask the spectrum analyzer for trace data
150 OUTPUT @Sa;"TRA?;"
160 ! Send the trace data to the computer
170 ENTER @Sa;Trace_a(*)
180 ! Create file to store trace
190 ! File is 13 records long
200 CREATE BDAT "DATA-A",13
210 ! Assign path for the file
220 ASSIGN @File TO "DATA-A"
230 ! Send trace data to the file
240 OUTPUT @File;Trace_a(*)
250 OUTPUT @File;Trace_a(*)
260 OUTPUT @Sa;"CONTS;"
270 LOCAL 7
280 ! Close file
290 ASSIGN @File TO *
300 ASSIGN @Sa TO *
310 END

First, using the CREATE command, we create an empty file on the disk for storing the trace. The file is 13 records long. (To determine the number of records, the 401-point trace is multiplied by 8 bytes per point, the storage required for real numbers, then divided by 256 bytes per record. The result is rounded to the next largest integer.)

Next, we assign an input and an output path to the file DATA-A. Then, we send the trace data to the file. (See lines 220 through 260.) Finally, in line 290, we close the file.
NOTE: If a program containing the CREATE command is run twice, the computer will report an error the second time because the file already exists. To prevent this error, place an exclamation mark before the CREATE command to “comment out” the line after you run the program the first time. (See line 210.)

When using this program as part of a larger program, the “file exists” error should be trapped out and then the new file will overwrite the existing file.
Loading Trace Data from a Computer Disk

If we want to return trace data to the spectrum analyzer for later viewing, we must work the “saving” process in reverse. The following program loads a trace that was previously saved on a computer disk into an array variable in the computer.

**Program Example:**

```
10 !Program reads trace data from computer file into a variable
20 !
30 REAL Trace_a(1:401) ! Create a 401-point trace array
40 ASSIGN @File TO "DATA_A" ! Assign path to file
50 ENTER @File;Trace_a(*) ! Enter trace into variable Trace-a
60 ASSIGN @File TO * ! Close file
70 END
```
Different Formats for Trace Data Transfers

One way to format trace data using the TDF command was introduced earlier in this chapter (TDF P). This section describes all the available trace data formats.

The spectrum analyzer provides five formats for trace data transfers: real number (P) format, binary (B) format, A-block format, I-block format, and measurement units (M) format.

The following table shows the data transmission sequence for trace data transfers for each trace data format, for a 5-element trace. The values of these elements are -5.23 dBm, -12.18 dBm, -24.83 dBm, -16.47 dBm, and -9.34 dBm. The reference level is 0.0 dBm and the logarithmic scale is 10 dB/div.

Parenthesis indicate the ASCII character represented by the data transmitted. Table entries without parenthesis indicate the decimal value of the data. A value followed by “-EOI” indicates that the HP-IB line EOI is asserted along with that data byte. For example, j0-EOI means that a linefeed (decimal value 10) is sent with EOI asserted.

Table 4  HP-IB Transmission Sequence for Trace Data Formats (TDF)

<table>
<thead>
<tr>
<th>Byte</th>
<th>TDF P</th>
<th>TDF M</th>
<th>TDF A (byte)</th>
<th>TDF A (word)</th>
<th>TDF I (byte)</th>
<th>TDF I (word)</th>
<th>TDF B (byte)</th>
<th>TDF B (word)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(-)</td>
<td>(7)</td>
<td>(#)</td>
<td>(##)</td>
<td>(##)</td>
<td>(##)</td>
<td>233</td>
<td>29</td>
</tr>
<tr>
<td>2</td>
<td>(5)</td>
<td>(4)</td>
<td>(A)</td>
<td>(A)</td>
<td>(I)</td>
<td>(I)</td>
<td>211</td>
<td>53</td>
</tr>
<tr>
<td>3</td>
<td>(6)</td>
<td>(7)</td>
<td>0</td>
<td>0</td>
<td>233</td>
<td>29</td>
<td>172</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>(2)</td>
<td>(7)</td>
<td>5</td>
<td>10</td>
<td>211</td>
<td>53</td>
<td>198</td>
<td>126</td>
</tr>
<tr>
<td>5</td>
<td>(3)</td>
<td>(6)</td>
<td>233</td>
<td>29</td>
<td>172</td>
<td>26</td>
<td>220-EOI</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>(7)</td>
<td>(6)</td>
<td>211</td>
<td>53</td>
<td>198</td>
<td>126</td>
<td>141</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(-)</td>
<td>(7)</td>
<td>172</td>
<td>26</td>
<td>220-EOI</td>
<td>21</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(1)</td>
<td>(8)</td>
<td>198</td>
<td>126</td>
<td>141</td>
<td>209</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>(2)</td>
<td>(2)</td>
<td>220-EOI</td>
<td>21</td>
<td>24</td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(6)</td>
<td>(6)</td>
<td>141</td>
<td>209</td>
<td>154-EOI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>(1)</td>
<td>(5)</td>
<td>24</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>(8)</td>
<td>(5)</td>
<td>209</td>
<td>154-EOI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>(6)</td>
<td>(1)</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4 | HP-IB Transmission Sequence for Trace Data Formats (TDF)

<table>
<thead>
<tr>
<th>Byte</th>
<th>TDF P</th>
<th>TDF M</th>
<th>TDF A (byte)</th>
<th>TDF A (word)</th>
<th>TDF I (byte)</th>
<th>TDF I (word)</th>
<th>TDF B (byte)</th>
<th>TDF B (word)</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>(-)</td>
<td>(7)</td>
<td></td>
<td></td>
<td>154-EOI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>(2)</td>
<td>()</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>(4)</td>
<td>(6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>()</td>
<td>(3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>(8)</td>
<td>(5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>(3)</td>
<td>(3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>()</td>
<td>()</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>(-)</td>
<td>(7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>(1)</td>
<td>(0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>(6)</td>
<td>(6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>()</td>
<td>(6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>(4)</td>
<td></td>
<td>10-EOI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>(7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>()</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>(-)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>(9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>()</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>(3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>(4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td>10-EOI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**M Format**

The M format is for sending trace data only. It formats the trace data in the internal format used by the spectrum analyzer, also known as measurement units. The range of the measurement units differs depending on the current instrument settings. The MDU command can be used to query various aspects of measurement units.
Example Range of Measurement Units

The analyzer is often set to a logarithmic display using logarithmic units of 10 dB/division. Figure 13 shows the range of measurements that the HP ESA-L1500A will use with these settings.

The displayed amplitude of each trace element falls on one of 10,000 vertical points with 8000 equal to the reference level, and -2000 equal to the bottom of the display. For log scale data, each point is equal to 0.01 dB. The peak of the signal in the figure is equal to -25 dBm, or two and a half divisions below the reference level. In measurement units, it is equal to 5500 (8000 - 2500 = 5500). The range of internal data is -32,768 to +32,767. In practice however, the range limits can be reached during trace math operations only.
Programming Examples

Different Formats for Trace Data Transfers

<table>
<thead>
<tr>
<th>Callout</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A number within the range of 8161 to 32,767 measurement units is obtainable with trace math operations only.</td>
</tr>
<tr>
<td>2</td>
<td>The area from 8000 (reference level) to 8160 (1.6 dB above reference level) represents the amount a trace element’s displayed amplitude can exceed the top graticule and still be valid.</td>
</tr>
<tr>
<td>3</td>
<td>The area from -2000 to 8000 represents the displayed range for trace amplitude data. The range of -2000 to 8000 varies according to the amplitude scale of the spectrum analyzer as follows:</td>
</tr>
<tr>
<td>4</td>
<td>A number within the range of -2000 to -32,768 measurement units is obtainable with trace math operations only.</td>
</tr>
</tbody>
</table>

Example of Using the M Format

This example sends trace data from the spectrum analyzer to the computer in M format.

Program Example:

```
10 !Example using trace data in M format
20!
30 INTEGER A(1:401) ! Declare array for number of trace points
40 ASSIGN @Sa TO 718
50 OUTPUT @Sa;"IP;RFCALIB ON;SNGLS;TS;" !50 MHz sig on
60 OUTPUT @Sa;"CF 50 MZ;SP 20 MZ;TS;" ! Take meas sweep
70 OUTPUT @Sa;"TDF M;TRA?;" ! Send trace A data in M format
80 ENTER @Sa;A(*) ! Computer receives data
90 PRINT A(*) ! Print trace data
100 END
```

**NOTE:**

All trace math functions are done using measurement units. See Chapter 4 for a list of all trace math functions. See the description for the AML command in Chapter 5 for an example of trace math subtraction in measurement units.

The result is in measurement units. For more detailed information about the M format, see the description for TDF in Chapter 5.


### Summary of the Trace Data Formats

<table>
<thead>
<tr>
<th>Trace Data Format</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDF P</td>
<td>Real number format</td>
<td>To send the trace data back to the spectrum analyzer, the data must be converted to measurement units.</td>
</tr>
<tr>
<td>TDFB</td>
<td>binary format</td>
<td>Fastest format for trace data transfers. Use the A-block format to send data back to the spectrum analyzer.</td>
</tr>
<tr>
<td>TDF A</td>
<td>A-block data format</td>
<td>Trace data preceded by “#,” “A,” and a two byte number. To use the A-block format for sending data, you must provide the number of data bytes.</td>
</tr>
<tr>
<td>TDF I</td>
<td>I-block data format</td>
<td>Trace data preceded by “#,” and “I.” This format is not recommended for use with an RS-232 interface. Unlike using the A-block format, you do not provide the number of data bytes when sending data.</td>
</tr>
<tr>
<td>TDF M</td>
<td>Measurement data format</td>
<td>TDF M can be used to send trace data back to the spectrum analyzer.</td>
</tr>
</tbody>
</table>

**P Format**

The P format allows you to receive or send trace data in a real-number format. This is the default format when the instrument is powered up. A parameter unit is a standard scientific unit. It depends on the current selected amplitude units (dBm, dBmV, dBμV, V, W). Numbers are in dBm, dBmV, dBμV, volts, or watts. The AUNITS command can be used to specify the amplitude units. Real-number data may be an advantage if you wish to use the data later in a program. However, data transfers using P format tend to be slow and take up a lot of memory (compared to binary format, the P format can take up to four times the amount of memory). Data is transferred as ASCII.

Although the spectrum analyzer can send the trace data to the computer as real numbers, the trace data cannot be sent back to the spectrum analyzer without changing it to measurement units (integers). See the following example.

---

59
Programming Examples

Different Formats for Trace Data Transfers

Example of Using the P Format

This example sends trace data to the computer using P format, changes it to measurement units, and then returns it to the spectrum analyzer. The following equations describe the relationship of logarithmic (dBm) and linear (volts) P format data to trace data in measurement units.

For a logarithmic display using logarithmic units:

\[ \text{dBm} = (\text{Trace data} - \text{MU}_{\text{ref level}}) \times \text{MU}_{\text{res}} + \text{Ref level (in dBm)} \]

For a linear display using linear units:

\[ \text{Volts} = \frac{\text{Reference level}}{\text{MU}_{\text{ref level}}} \times \text{Trace data} \]

Where:

- \( \text{MU}_{\text{ref level}} \) = Measurement Units for the Reference Level
- \( \text{MU}_{\text{res}} \) = Measurement Unit Resolution

The measurement unit resolution and the value for the reference level can be queried from the analyzer using the MDU command as in the following example.

Program Example:

```plaintext
10 ! Example using trace data in P format
20 !
30 ASSIGN @Sa TO 718
40 REAL Trace_data(1:401) ! Declare array for trace data
50 OUTPUT @Sa;"IP;TS;RFcalib ON;"! Initialize, 50 MHz Sig on
60 OUTPUT @Sa;"CF 50 MZ;SP 10 MZ;SGLS;TS;"! Take meas sweep
70 OUTPUT @Sa;"MDU RL_LOG?;"! Get meas unit at ref level in Log
80 ENTER @Sa;Mdu_rl
90 OUTPUT @Sa;"MDU LOG-RES?;"! Get meas unit resolution in Log
100 ENTER @Sa;Mdu_res
110 OUTPUT @Sa;"MDU P;TRA?;"! Use P format, output trace A
120 ENTER @Sa;Trace_data(*)! Computer receives data
130 OUTPUT @Sa;"VIEW TRA;MOV TRA,0;"! Verify transfer, set trace=0
140 OUTPUT @Sa;"RL?;"! Use RL ampl to change integers to reals
150 ENTER @Sa;Ref_level! Get ref level
160 DISP 'Press CONTINUE when ready'
170 PAUSE
180 DISP
190 MAT Trace-data= Trace-data-(Ref-level) ! Changes trace data
200 MAT Trace-data= Trace_data/(Mdu_res) ! from reals to integers
210 MAT Trace-data= Trace_data+(Mdu_rl) ! (in meas units)
220 PRINT Trace-data(*)
230 OUTPUT @Sa;"TRA;"
```
Programming Examples
Different Formats for Trace Data Transfers

For more detailed information about the P format, see the description for TDF in Chapter 5.

**B Format**

The B format allows you to receive trace data in a binary format. The B format provides the fastest data transfer and requires the least amount of memory to store data. Each data point is transferred in binary as two 8-bit bytes. The data points are in the internal representation of measurement units. The B format does not send a header. (The A-block format is described below and it does send a header before the data.) An end-or-identify (EOI) is sent with the last byte of data. See “P Format” above for more info about measurement units.

**Example of Using the B Format**

This example sends trace data from the spectrum analyzer in B format. The trace data format must be changed to A-block format to return the trace data to the spectrum analyzer. See following example.

```
NOTE:
It is not possible to return data to the spectrum analyzer using binary format. You must use either A-block, I-block, or M-block format to return the trace data to the spectrum analyzer.
```

**Program Example:**

```
10 ! This example uses trace data in B-format
20 !
30 INTEGER Tra_binary(1:401) ! Declare array for trace data
40 ASSIGN @Sa TO 718;FORMAT OFF
50 OUTPUT @Sa;"IP;TS;RFCALIB ON;" ! Turns 50 MHz signal on
60 OUTPUT @Sa;"CF 50 MZ;SP 10 MZ;SNGLS;TS;" ! Take meas sweep
70 OUTPUT @Sa;"MDS W;TDF B;TRA?;" ! Output trace A data
80 ENTER @Sa;Tra_binary(*) ! Change data to A-block format
90 OUTPUT @Sa;"TDF A;" ! Change data to A-block format
100 OUTPUT @Sa;"MOV TRA, 0;" ! Verify transfer;set trace=0's
110 DISP 'Press CONTINUE when ready'
120 PAUSE
130 DISP
140 OUTPUT @Sa USING ";K,W","TRA#A",802 ! Prepare SA for data
150 OUTPUT @Sa;Tra_binary(*) ! Transfer data to SA
160 OUTPUT @Sa;"VIEW TRA;" "
170 LOCAL @Sa
180 END
```

The result is transmitted as binary information. The MDS command can be used to change the data format from two 8-bit bytes to one 8-bit byte. For more detailed information about the B format and the MDS command, see the descriptions for TDF and MDS in Chapter 5.
Binary data can be converted to dBm or volts. For example, use the following equation to change the trace data (in measurement units) to a real logarithmic number (dBm):

\[
\text{dBm} = (\text{Trace data} - \text{MU}_{\text{ref level}}) \times \text{MU}_{\text{res}} + \text{Ref level (in dBm)}
\]

To change the trace data (in measurement units) to linear data (volts):

\[
\text{volts} = \frac{\text{reference level}}{\text{MU}_{\text{ref level}}} \times \text{trace data}
\]

Where:

- \(\text{MU}_{\text{ref level}}\) = Measurement Units for the Reference Level
- \(\text{MU}_{\text{res}}\) = Measurement Unit Resolution

The following programming converts binary data to dBm.

### Program Example:

10 ! This example converts binary data to dBm
20  
30 ASSIGN @Sa-bin TO 718;FORMAT OFF
40 ASSIGN @Sa TO 718
50 INTEGER Trace-a(1:401)
60 OUTPUT @Sa;"AUNITS DBM;"
70 OUTPUT @Sa,"RL?;"*
80 ENTER @Sa;Ref_lev
90 OUTPUT @Sa;"MDU RL_LOG?;"*
100 ENTER @Sa;Mdu_rl
110 OUTPUT @Sa;"MDU LOG-RES?;"
120 ENTER @Sa;Mdu_res
130 OUTPUT @Sa;"TDF B;TRA?;"
140 ENTER @Sa;bin;Trace_a(*)
150 ! Now the spectrum analyzer has all the data to determine
160 ! the measured trace data
170 REAL Trace_a_real(1:401)
180 MAT Trace-a= Trace_a-(Mdu_rl) ! Results in hundreths of
190  ! dB below ref level
200 MAT Trace-a-real= Trace_a*(Mdu_res) ! Now in dB below ref lev
220 MAT Trace-a-real= Trace_a_real+(Ref_lev)
230 FOR I=1 TO 401
240 PRINT Trace-a-real(I)
250 NEXT I
260 END

The following programming converts binary data to volts.

### Program Example:

10 ! Program converts binary data to volts
20  
30 ASSIGN @Sa_bin TO 718;FORMAT OFF
40 ASSIGN @Sa TO 718
A-Block Format

The A-block format is similar to binary format in that each data point is sent as two 8-bit bytes. This format is in the internal representation of measurement units. See “P Format” above for more information about measurement units. A-block format also transfers a four-byte header before the 401 points (802 bytes) of trace data. These bytes are the ASCII character ‘#’, ‘A’, and two-byte number representing the length of the trace data, followed by the data bytes.

Using A-block format allows you to write your program so that it reads the header data first and determines how many bytes will follow. It can then allocate the proper sized array for the data based on that information.

Example of Using the A-Block Format

This example sends trace data from the spectrum analyzer to the computer and back to the spectrum analyzer in A-block format.

Program Example:

```plaintext
50 INTEGER Trace_a(l:401)
60 OUTPUT @$sa;"LN;AUNITS V;"
70 OUTPUT @$sa;"RL?;"
80 ENTER @$sa;Ref_lev
90 OUTPUT @$sa;"MDU RL_LIN?;"
100 ENTER @$sa;Mdu_rl
110 Ref_lev_factor=Ref_lev/Mdu_rl
120 OUTPUT @$sa;"TDF B;TRA?;"
130 ENTER @$sa_bin;Trace_a(*)
140 ! Now the spectrum analyzer has all the data to determine
150 ! the measured trace data
160 REAL Trace_a_real(l:401)
170 MAT Trace_a-real= Trace_a*(Ref_lev_factor)
180 FOR I=1 TO 401
190 PRINT Trace_a-real(I)
200 NEXT I
210 END
```

A-Block Format

The A-block format is similar to binary format in that each data point is sent as two 8-bit bytes. This format is in the internal representation of measurement units. See “P Format” above for more information about measurement units. A-block format also transfers a four-byte header before the 401 points (802 bytes) of trace data. These bytes are the ASCII character ‘#’, ‘A’, and two-byte number representing the length of the trace data, followed by the data bytes.

Using A-block format allows you to write your program so that it reads the header data first and determines how many bytes will follow. It can then allocate the proper sized array for the data based on that information.

Example of Using the A-Block Format

This example sends trace data from the spectrum analyzer to the computer and back to the spectrum analyzer in A-block format.

Program Example:

```plaintext
50 INTEGER Trace_a(l:401)
60 OUTPUT @$sa;"LN;AUNITS V;"
70 OUTPUT @$sa;"RL?;"
80 ENTER @$sa;Ref_lev
90 OUTPUT @$sa;"MDU RL_LIN?;"
100 ENTER @$sa;Mdu_rl
110 Ref_lev_factor=Ref_lev/Mdu_rl
120 OUTPUT @$sa;"TDF B;TRA?;"
130 ENTER @$sa_bin;Trace_a(*)
140 ! Now the spectrum analyzer has all the data to determine
150 ! the measured trace data
160 REAL Trace_a_real(l:401)
170 MAT Trace_a-real= Trace_a*(Ref_lev_factor)
180 FOR I=1 TO 401
190 PRINT Trace_a-real(I)
200 NEXT I
210 END
```
Programming Examples

Different Formats for Trace Data Transfers

The transferred trace data consists of $A$, a two-byte number representing the most significant byte (MSB), and the least significant byte (LSB) of the length of the data to follow. Depending on the computer you are using, the data bytes may appear as symbols instead of numbers or they may not appear at all since some data bytes have values representing unprintable characters. Consult your computer documentation to determine the numeric value of the data bytes.

For more detailed information about the A-block format and the MDS command, see the descriptions for TDF and MDS in Chapter 5.

I-Block Format

NOTE: The I-block format is not recommended for use with the RS-232 interface (Option 1AX).

The I-block format transfers data points as two 8-bit bytes in the internal representation of measurement data. In addition to transferring trace data, I-block format also transfers the characters "#" and "I". These characters indicate that the trace data is in I-block format. The I-block format allows the spectrum analyzer to accept up to 401 points of trace data when using I-block format. Fewer than 401 points of trace data can be specified, and the spectrum analyzer will accept data until an EOI signal is sent to it. Therefore, returning the trace data to the spectrum analyzer requires the END instruction. END asserts the EOI line on HP-IB with the last data byte transferred. (See the following example.)

Example of Using the I-Block Format

This example sends trace data from the spectrum analyzer to the computer and back to the spectrum analyzer in I-block format.

Program Example:

```
10 ! Example using trace data in I-block format
20 !
30 INTEGER Tra_binary(1:401) ! Declare array for trace data
40 DIM Header$[2] ! Declare array for # and I header
50 ASSIGN @Sa TO 718
60 OUTPUT @Sa;"IP;SNGLS;RFCALIB ON;TS;" ! Turn 50 MHz sig on
61 OUTPUT @Sa;"CF 50 MZ;SP 10 MZ;TS;" ! Take meas sweep
70 OUTPUT @Sa;"TDF I;TRA;" ! Send trace A data in I format
71 ! Receive the header and the trace data from the analyzer
80 ENTER @Sa USING ":,2A,401(W);Header$;Tra_binary(*)
90 PRINT "Press CONTINUE to return data to the analyzer"
100 PAUSE
110 OUTPUT @Sa;"IP;TS;VIEW TRA;" ! View trace A
```

64
The END statement in line 130 sends the spectrum analyzer the last data byte stored in the array and sets the HP-IB EOI line “true,” as required by the I-block format.

The transferred trace data consists of \#1, followed by data bytes until the EOI line is set true.

For more detailed information about the I-block format and the MDS command, see the descriptions for TDF and MDS in Chapter 5.
Saving and Loading Instrument States

The spectrum analyzer’s control settings (or its “state”) can be saved with a computer and retrieved later to streamline test sequences or repeat manual measurements. Control settings can be stored in a state file in the spectrum analyzer (25 maximum), in computer memory, or on a computer disk.

The first program in this section demonstrates techniques for saving an instrument state, along with its current trace A data. The second program demonstrates how the state information and the trace data is read from the computer and returned to the spectrum analyzer.

If you wish to save states in the spectrum analyzer, see the descriptions of the save (SAVE) and load (LOAD) commands in Chapter 5.

Saving the Spectrum Analyzer’s State

The following program saves and loads a state and trace from the spectrum analyzer to a computer file.

**Program Example:**

```plaintext
10 ! Program saves a SA state and trace in the computer
20 !
30 ! Define 202 character string:
40 DIM Learn_string$[202]
50 ! Create 401-point array to store trace:
60 INTEGER Trace_a(l:401)
70 ASSIGN @Sa TO 718
80 OUTPUT @Sa;"IP;SNGLS;RFCALIB ON;TS;" ! Initialize
90 OUTPUT @Sa;"CF 50 MZ;SP 10 MZ;TS;"
100 ! Set output format for two byte integers:
110 OUTPUT @Sa;"TDF B;"
120 ! Ask spectrum analyzer for trace data:
130 OUTPUT @Sa;"TRA;"
140 ! Send trace to the computer:
150 ENTER @Sa USING ",W";Trace_a(*)
160 ! Get learnstring from spectrum analyzer:
170 OUTPUT @Sa;"OL;"
180 ENTER @Sa USING ",202A";Learn_string$
190 ! Create file to store trace:
200 CREATE BDAT "STATE",4
210 ! Assign path to the file:
220 ASSIGN @File TO "STATE"
230 ! Send trace to the file:
240 OUTPUT @File;Learn_string$,Trace_a(*)
250 ! Return output format to default mode:
260 OUTPUT @Sa;"TDF P;"
270 ! Close file:
280 ASSIGN @File TO *
```

66
This HP-IB program stores the trace in the variable called Trace-a(\textsuperscript{(*)}). The state of the spectrum analyzer is stored in the variable Learn-string$. These two variables are then saved in a file called STATE. Finally, the file is stored on a disk.

Using the data stored in STATE, the spectrum analyzer settings can be reset according to the saved state. Then, using the stored trace data, trace data can be viewed on the spectrum analyzer display.

Line 40 gives the dimensions of the learn string using the HP BASIC DIM command. Learn strings for the spectrum analyzer require 202 bytes of storage space. Also see the output learn string (OL) command.

Line 110 uses TDF B to format the output in binary. Binary provides the fastest data transfer and requires the least amount of memory to store data. Each data point is transferred in binary as two \textbf{8-bit} bytes. The data points are in the internal representation of measurement data. (See Table 5 in this chapter for more information about trace data formats.)

When the trace and state data are sent from the spectrum analyzer to the computer, they must be formatted. Lines 150 and 180 format trace data with the BASIC USING command. In the formatting statement, “\#” indicates that the statement is terminated when the last ENTER item is terminated. EOI (end-or-identify) and LF (line feed) are item terminators, and early termination will result in an inaccurate learn string. “W” specifies word format. “202A” indicates the size of the learn string.

Line 200 creates a file called STATE that is 4 records long. (To determine the number of records for the computer in our example, the \textbf{401-point} trace is multiplied by 2 bytes per point and the \textbf{202-byte} learn string is added to give 1004 bytes total. This total is divided by 256 bytes per record, resulting in 4 records.)

\textbf{NOTE:} If a program containing the CREATE command is run twice, the computer will report an error the second time because the tile already exists. To prevent this error, place an exclamation mark before the \textbf{CREATE} command to “comment out” the line after you run the program the first time. (See line 200.)

When using this program as part of a larger program, the “file exists” error should be trapped out and then the new file will overwrite the existing file.
Returning the Spectrum Analyzer to its Former State

The following program reads a state and trace stored in a file from the previous example and loads it into the spectrum analyzer.

Program Example:

```
10 !Program gets trace file from computer, puts SA back in
20 !
30 !define 202 character string:
40 DIM Learn_string$[202]
50 !create 401 point array to store trace:
60 INTEGER Trace_a(1:401)
70 ASSIGN @Sa TO 718
80 !assign path to the file:
90 ASSIGN @File TO "STATE"
100 !get values for Learn_string$ and Trace-a(*) from disk:
110 ENTER @File;Learn_string$,Trace_a(*)
120 !send learnstring to spectrum analyzer:
130 OUTPUT @Sa;"IP;DONE;"
140 ENTER @Sa
150 OUTPUT @Sa;Learn_string$
160 !set single sweep mode:
170 OUTPUT @Sa;"SNGLS;"
180 !prepare spectrum analyzer for a trace from
190 !the computer:
200 !the computer:
210 OUTPUT @Sa;"TRA #A"
220 !send trace to the spectrum analyzer
230 OUTPUT @Sa USING ";#;'W;802,Trace_a(*)
240 !view trace to see it was sent:
250 OUTPUT @Sa;"VIEW TRA;"
260 !close file:
270 ASSIGN @File TO *
280 END
```

This program reads the state and trace data stored in the file STATE, then loads it into the variables Learn-string$ and Trace-a(*).

First, the settings of the spectrum analyzer that were stored in the variable Learn-string$ are recalled. The spectrum analyzer state is changed to the same state as when the trace was stored. Then previously stored trace data is returned to the spectrum analyzer and the trace is viewed on the spectrum analyzer screen. Line 230 uses the HP BASIC USING command to format the trace data.
Measuring Harmonic Distortion

The harmonic distortion program presented here illustrates how the spectrum analyzer can be directed by a computer to make a complete measurement. Measuring the percent of total harmonic distortion is tedious when performed manually. It involves: tuning to the fundamental and to each harmonic of interest, recording the amplitude of each signal, converting these amplitudes to linear units (volts), and calculating the result using a formula. The following program measures percent of total harmonic distortion automatically, quickly, and accurately.

The program operates as if we were making the measurement manually.

**NOTE:**
This program is designed to measure harmonics of a signal that is greater than 20 MHz.

**Program Example for the HP-IB Interface:**

```plaintext
10 !Program measures total harmonic distortion
20 !
30 ASSIGN @Sa TO 718 ! assign IO path to spectrum analyzer
40 Variables: ! define variables:
50 REAL Fundamental,Fund_amptd_v,Fund_amptd_dbm
60 REAL Prcnt_distort,Sum_sqr
70 INTEGER Max_harmonic,I,Number
80 !allow user to change the number of harmonics:
90 Max_harmonic=4
100 ALLOCATE REAL Harm_v(2:Max_harmonic)
110 ALLOCATE REAL Harm_dbc(2:Max_harmonic)
120 GOSUB Clearsreen! clear the alpha screen
130 !ask for the frequency of the fundamental:
140 OUTPUT CRT USING "4/,10X,K,3/";"***HARMONIC  DISTORTION***"
150 OUTPUT CRT USING "1OX,K";'CONNECT  SOURCE TO INPUT"
160 OUTPUT CRT USING "1OX,K";'ENTER FUNDAMENTAL FREQUENCY IN MHZ"
170 OUTPUT CRT USING "1OX,K";'WHEN READY, PRESS ENTER n"
180 INPUT Fundamental
190 GOSUB Clearsreen! clear the alpha screen
200 Fundamental: ! write message on screen:
210 DISP "MEASURING FUNDAMENTAL"
220 !preset the spectrum analyzer, set single sweep mode, and
230 !take sweep:
240 OUTPUT @Sa;"IP; SNGLS; TS;"
250 !tune the spectrum analyzer to the fundamental freq and set
260 !20 MHz span:
270 OUTPUT @Sa,"CF ";Fundamental;"MZ;"
280 OUTPUT @Sa;"SP 20MZ; TS;"
290 !put a marker on signal peak, move marker to
300 !reference level:
310 OUTPUT @Sa;"MKPK HI; MKRL; TS;"
320 !find signal peak, activate signal track, and
330 !narrow span:
340 OUTPUT @Sa;"MKPK HI; TS;"
350 OUTPUT @Sa;"MKTRACK ON; SP 100KZ; TS;"
360 !turn off signal track:

```
Programming Examples

Measuring Harmonic Distortion

```plaintext
370  OUTPUT @Sa;"MKTRACK OFF;"
380  !find the peak of the signal; move peak to center
390  !of screen:
400  OUTPUT @Sa;"AUNITS V;"! MAKE READOUT UNITS VOLTS
410  !find peak of signal; send amplitude value to
420  !computer
430  !enter the amplitude of the fundamental:
440  OUTPUT @Sa;"MKPK HI; MKA?;"
450  ENTER @Sa;Fund_amptd_v
460  !send marker frequency to the computer, enter
470  !frequency value:
480  OUTPUT @Sa;"MKF?;"
490  ENTER @Sa;Fundamental
500  !make the fundamental frequency the center freq
510  !step size:
520  OUTPUT @Sa;"MKSS;"
530  !set the fundamental frequency units to MHz:
540  Fundamental=Fundamental/1.E+6
550  !harmonics: !measure the amplitudes of the harmonics:
560  FOR Number=2 TO Max-harmonic
570    DISP 'MEASURING HARMONIC #";Number
580    OUTPUT @Sa;"SP 20MZ;"!set span to 20 MHz
590    OUTPUT @Sa;"CF UP; TS;"!tune to next harmonic
600  !take second sweep to allow spectrum analyzer to move to the
610  !center frequency; find the signal peak; activate
620  !signal track:
630    OUTPUT @Sa;"TS;"
640    OUTPUT @Sa;"MKPK HI; MKTRACK ON; SP 100KZ; TS;"
650  !turn off signal track:
660    OUTPUT @Sa;"MKTRACK OFF;"
670  !find signal peak; send amplitude value to computer
680  !enter the amplitude of the harmonic:
690    OUTPUT @Sa;"MKPK HI; MKA?;"
700    ENTER @Sa;Harm_v(Number)
710  NEXT Number
720  !set amplitude units to dBm:
730  OUTPUT @Sa;"AUNITS DBM;"
740  !calculate the fundamental amplitude in dBm because
750  !it was measured in volts:
760    Fund_amptd_dbm=10*LGT(Fund_amptd_v^2/.05)
770  !calculate the sum of the squares of the amplitudes
780  !of the harmonics; calculate amplitudes of
790  !harmonics (dBm):
800  Sum-sqr=0
810  FOR I=2 TO Max-harmonic
820    Sum_sqr=Sum_sqr+Harm_v(I)^2
830    Harm_dbc(I)=20*LGT(Fund_amptd_v/Harm_v(I))
840  NEXT I
850  !calculate the percent distortion:
860    Prcnt_distort=SQR(Sum_sqr/Fund_amptd_v^100)
870  GOSUB Clearscreen !clear the alpha screen:
880  !Output data:
890  !send data to the screen of the computer:
900  OUTPUT CRT USING "7/,1X,K";"HARMONIC DISTORTION RESULTS"
910  OUTPUT CRT USING "11X,K,DDD.D,K";"FREQ = ";Fundamental;"MHz"
920  OUTPUT CRT USING "11X,K,DDD.D,K";"AMP = ";Fund_amptd_dbm;"dBm"
930  OUTPUT CRT USING "11X,K,DDD.D.K";"2nd Harm= -";Harm_Db(2);"dBc"
940  OUTPUT CRT USING "11X,K,DDD.D.K";"3rd Harm= -";Harm_dbc(3);"dBc"
950  FOR I=4 TO Max-harmonic
960  OUTPUT CRT USING "10X,DD,D,K,DDD.D,K";I;"th Harm= -";
```
The program prompts the user to connect a source to the spectrum analyzer INPUT and enter the source frequency. It sets the spectrum analyzer center frequency to the value of the source, or fundamental, frequency. It measures and records the frequency and amplitude of the fundamental, then measures and records the amplitude of the second, third, and fourth harmonics. These values are used to compute percent of harmonic distortion. The result of the harmonic distortion percentage computation, plus harmonic amplitudes in dBc (decibels relative to the carrier), are displayed on the computer display. Comments have been added (after the exclamation points) to help clarify the program.

If necessary, change the number of harmonics in line 90.
Monitoring System Operation

The following information only applies to remote operation using the HP-IB interface.

The programming techniques discussed so far describe communication between the analyzer and the computer, where the sequence of all data transfer is controlled by a computer program. This section describes how the analyzer can interrupt computer operation when it has attained a particular state.

The interrupting process is called a service request. Service requests have many applications. They facilitate economical use of computer-processing time when the analyzer is part of a large measurement system. For example, after the computer initiates an analyzer measurement, the computer can make calculations or control other devices via HP-IB while the analyzer is measuring. When the analyzer is through, it signals the computer with a service request. The computer service-request subprogram then determines what the computer will do next. Service requests can also be used to report analyzer errors and other analyzer events, such as end-of-sweep.

**Programming Steps Using Service Requests**

The main points to using service requests are highlighted below:

1. Choose the conditions for generating service requests.
2. Set a bit mask that enables only these chosen conditions.
3. Prepare the computer to accept service requests. Use the `ON INTR..GO SUB` and `ENABLE INTR` statements.
4. Once an interrupt is triggered, use the analyzer `STB` command or the `SPOLL` statement to read the interrupt.

**Interrupt Process**

The interrupt process begins when the analyzer “requests” attention by setting the HP-IB service-request line (SRQ) true. The computer must be programmed to respond to this event. Typically, the computer is programmed to interrupt normal program execution and call a user-specified subroutine when the service request occurs. If multiple instruments are being controlled remotely, this subroutine determines which instrument or instruments on HP-IB caused the service request. Then, it may be necessary to call another subroutine that determines why a particular instrument requested service (because there can be more than one possible cause). Finally, one or more subroutines will be called to respond to the indicated
events. Note that more than one instrument can request service simultaneously, and each instrument can have more than one event to report. These steps are summarized below.

2. Analyzer requests service by setting the SRQ line true.
3. Computer branches to routine that determines the instrument or instruments that caused the SRQ.
4. Computer branches to routine that determines why a particular instrument is requesting service.
5. Computer branches to routine to process a specific event in a particular instrument.

Some of the routines (that are shown above) can be omitted, if only one instrument has been instructed to use the SRQ line, or if a particular instrument has been instructed to use the SRQ line for only one event.

Several system-level statements are required to make the computer respond to service requests. The HP BASIC statement, ENABLE INTR (enable interrupt), tells the computer to monitor the service-request line. The on-interrupt statement, ON INTR, specifies where the computer program will branch when a service request occurs. If more than one instrument could cause the service request, or if an instrument can cause a service request for more than one reason, the serial-poll statement, SPOLL, is used. The serial-poll statement is always required to clear the service request being generated by the instrument.

**Analyzer Status Byte**

The analyzer status byte indicates the status or occurrence of certain analyzer functions. The status byte contains eight bits, numbered 0 to 7, with bit 0 being the least significant bit.

Bits 0, 1, 2, 3, 4, 5, and 7 represent specific conditions or events. These bits are referred to as condition bits and event bits. Condition bits reflect a condition in the analyzer that can be present or absent at any given moment. Event bits reflect the occurrence of a transition or event within the analyzer.

Bit 6 is set by the analyzer to indicate whether or not it is requesting service.

When the analyzer is instructed not to use the service request line on HP-IB, the status bits always reflect the current condition of the analyzer. In this situation, the event bits in the status byte should not be used. Event bits are only true at the exact instant of a transition in the analyzer and as such are not reliable when service requests are disabled.
When the analyzer is instructed to generate service requests for one or more conditions or events, the status bits reflect the current condition of the analyzer until a service request is generated. Then, the bits that are generating the service request are held true until the status byte is read out of the analyzer (by the HP BASIC SPOOL system-level statement), by the STB? command, or until an HP-IB DEVICE CLEAR (HP BASIC CLEAR system-level statement) is received. These actions clear the status byte to once again reflect the current conditions and events within the analyzer.

**The Service-Request Mask**

The service-request mode is enabled and controlled by the request service condition command, RQS. It defines a service-request mask that specifies which of the status byte bits can generate a service request. Below, RQS specifies the ILLEGAL-COMMAND and COMMAND-COMPLETE states (bits 5 and 4, respectively) for service requests.

```plaintext
OUTPUT 0Sa;"RQS 48;";
```

Selects bit 5 and 4 (32+16=48) to enable service request mode for ILLEGAL-COMMAND and COMMAND-COMPLETE.

Also selects bits 5 and 4 as above, but is somewhat easier to read.

Once RQS is executed, the analyzer requests service by setting the SRQ line true when the desired conditions or events occur.

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Decimal Equivalent</th>
<th>Analyzer State</th>
<th>Description</th>
<th>(E)vent or (C)ondition</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>128</td>
<td></td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>RQS</td>
<td>Requests service</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>ILLEGAL COMMAND</td>
<td>Set when illegal command is received</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>COMMAND COMPLETE</td>
<td>Any command is completed</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>HARDWARE BROKEN</td>
<td>Set when hardware error occurs</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>END OF SWEEP</td>
<td>Set when any sweep is completed</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>UNIT KEY PRESSED</td>
<td>Set when any units key is pressed</td>
<td></td>
</tr>
</tbody>
</table>
Table 6 Status Register

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Decimal Equivalent</th>
<th>Analyzer State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>OPERATOR NOTIFICATION</td>
<td>Set if an overload is detected on the analyzer RF input, if excessive reverse power is detected on the RF output (options 1DN or 1DQ only), or if the tracking generator becomes unleveled (options 1DN or 1DQ only).</td>
</tr>
</tbody>
</table>

**Computer Interrupt Statements**

Now that the spectrum analyzer is prepared to trigger service requests, you must prepare the computer to accept this type of interrupt. Use the BASIC statements `ON INTR...GOTO` or `CALL` or `GOSUB` and `ENABLE INTR`. `ON INTR...GOTO` causes the computer to branch to a subroutine or some other part of the program when an interrupt is generated. `ENABLE INTR` enables the computer to accept an interrupt. These two commands appear below.

```
10 OUTPUT @Sa;"RQS 16;"
20 ON INTR 7 GOSUB Srq
30 ENABLE INTR 7;2
```

In this example, Line 20 indicates that if an interrupt appears (ON INTR 7), the computer is to go to the subroutine Srq (GOSUB Srq). The 7 specifies the interface select code; in this case, it refers to the Hewlett-Packard Interface Bus (HP-IB). Line 30 enables the computer to accept an interrupt. Here, the 7 again specifies the HP-IB select code. The semicolon is part of the BASIC statement `ENABLE INTR`. The 2 indicates that the interrupt is specifically a service request interrupt, which is asserted from the SRQ line of the HP-IB. From Line 10, you know the interrupt will be a service request triggered from a "command complete" condition.

Now that the spectrum analyzer and computer can assert and accept service-request interrupts, choose an event that will trigger the service request and create a subroutine to handle the interrupt. In the example below, the take sweep command (TS) is used to trigger a command complete signal. (Because 10 video-averaging sweeps are desired, this signal does not occur until after the selected number of averages is complete.) This service request will cause the computer to go to the subroutine Srq. The subroutine identifies the type of service request and prints it on the computer screen. See Example 1.
Example 1:

```
10 ASSIGN @Sa TO 718
20 OUTPUT @Sa;"IP;SNGLS;RFCALIB ON;TS;" ! Initialize
30 OUTPUT @Sa;"CF 50 MZ;RB .3 MZ;SP 10 MZ;TS;" ! Set analyzer
50 OUTPUT @Sa;"RQS 16;" ! Set mask for command complete
60 ON INTR 7 GOTO Srq
   ENABLE INTR 7;2
70 OUTPUT @Sa;"VAVG 30;TS;" END ! Video avg on, assert EOI
80 Idle:GOTO Idle
90 Srq:Sbyte=SPOLL(718)
100 PRINT Sbyte
110 PRINT "VIDEO AVERAGING IS COMPLETE"
120 OUTPUT @Sa;"RQS 0;"
130 END
```

Line 20 sets the desired instrument state. Note that the instrument is set to single-sweep mode. This allows the video averages to happen only when the take-sweep command is sent.

Line 50 directs that the computer should execute the Srq routine when an interrupt occurs.

Line 60 enables the computer to accept the interrupt.

Line 70 selects the number of video averages desired.

Line 80 sends the take-sweep command; during the 30 video averages that will now occur, the computer remains on line 80. When the video averaging is complete, TS is complete and the “command complete” condition is satisfied. The computer then branches to the subroutine Srq.

Lines 90 and 100 causes the computer to read the decimal equivalent of the generated service request into the variable Sbyte. The computer then prints the value, alerting you that the interrupt has occurred.

Line 120 returns the status register to its initial state (that is, no conditions are masked).

Reading Service Request Data

In the above example, you used the serial-poll statement (SPOLL) to read the service request data into a variable. The STB (status byte query) command also reads service request data. Example 2 shows how.

Example 2:

```
10 ASSIGN @Sa TO 718
20 OUTPUT @Sa;"IP;SNGLS;RFCALIB ON;TS;"
30 OUTPUT @Sa;"CF 50 MZ;RB .3 MZ;SP 10 MZ;TS;"
40 OUTPUT @Sa;"RQS 16;"
```
Line 40 sets the bit mask so that only the “command complete” condition is set. On Line 120, once the “command complete” condition is satisfied (in this case, after 30 video averages), the STB command queries the spectrum analyzer for the service-request data. The data is then entered into variable Sbyte and printed. The value returned is the decimal equivalent of the generated service request.

**Reading Service Requests From More Than One Instrument**

Most instruments that can be controlled remotely have service request capability similar to that in the spectrum analyzer. You may want to take advantage of this capability in other instruments also. If you have more than one instrument on a bus that can generate a service request, you need to modify the above program to look for interrupts from more than one instrument. See Example 3.

**Example 3:**

```
10   ASSIGN @Sa TO 718
20   OUTPUT @Sa;"IP;SNGLS;RFCALIB ON;TS;"
30   OUTPUT @Sa;"CF 50 MZ;RB .3 MZ;SP 10 MZ;TS;"
40   OUTPUT @Sa;"RQS 16;"
50   ON INTR 7 GOSUB Srq
60   ENABLE INTR 7;2
70   Done=0
80   OUTPUT @Sa;"VAVG 30;TS;";END
90   Idle:IF Done=0 THEN
100  GOTO Idle
110  STOP
120  Srq:OUTPUT @Sa;"STB?;"
130  ENTER @Sa;Sbyte
140  PRINT Sbyte
150  PRINT "VIDEO AVERAGING IS COMPLETE"
160  OUTPUT @Sa;"RQS 0;"
170  Done=1
180  RETURN
190  END IF
200  END
```

```
120 Srq:Sbyte_1=SPOLL(718)
130 Sbyte_2=SPOLL(705)
140 IF BIT(Sbyte_1,6)=1 THEN
150 PRINT "SERVICE REQUEST",Sbyte_1,"ON ADDRESS 18"
160 OUTPUT @Sa;"RQS 0;"
170 STOP
180 END IF
190 IF BIT(Sbyte_2,6)=1 THEN
200 PRINT "SERVICE REQUEST",Sbyte_2,"ON ADDRESS 5"
210 END IF
220 ENABLE INTR 7;2
```
In this example, you execute the SPOLL command for each instrument that can cause a service request interrupt; in this case, the analyzer or an instrument that is set to address 5. Once the instruments are queried for interrupts, the `IF...THEN` statements provide a way to branch to the appropriate routine.

**Testing Service Request Routines**

In the previous programming examples, you knew that a service request would be generated when the VAVG command was completed. You could easily test the program and make sure that it worked. However, service requests may not always be so predictable; this can make a program difficult to test. The spectrum analyzer SRQ command automatically triggers any service request you choose. Of course, as with other service requests, you must set the bit mask before executing the SRQ command. See Example 4.

**Example 4:**

```
10 ASSIGN @Sa TO 718
20 OUTPUT @Sa;"IP;SNGLS;RFCALIB ON;TS;"
30 OUTPUT @Sa;"CP 50 MZ;RB .3 MZ;SF 10 MZ;TS;"
40 OUTPUT @Sa;"RQS 16;"
50 ON INTR 7 GOSUB Srq
60 ENABLE INTR 7;2
70 Done=0
80 OUTPUT @Sa;"SRQ 16;"
90 Idle: IF Done=0 THEN
100 GOTO Idle
110 STOP
120 Srq:Sbyte=SPOLL(718)
130 PRINT Sbyte
140 PRINT "RQS 0;"
150 OUTPUT @Sa;"INTERRUPT GENERATED"
160 Done=1
170 RETURN
180 END IF
190 END
```

On Line 80 a “command complete” service request is immediately generated, and you can be sure that the routine will work.
Using Markers

Markers can be used to locate signal peaks. A markers position is defined by two values, the amplitude and the frequency (or time.) The frequency, amplitude, or time values of a marker can be queried. This information can be used to identify signals and to redefine the analyzer’s displayed frequency span or amplitude. The following program lines show several different marker commands being used.

Program Example:

```
10 !Program fragment outputs different marker types
20 OUTPUT @Sa; "TDF P;MKREAD FRQ;MKF?;" !Freq, if not 0 span
30 OUTPUT @Sa; "TDF P;MKREAD PER;MKF?;" !Marker time(s), if 0 span
40 OUTPUT @Sa; "TDF P;MKREAD SWT;MKF?;" !Marker time (in seconds)
50 OUTPUT @Sa; "TDF P;MKREAD IST;MKF?;" !Freq of 1/marker time
60 OUTPUT @Sa; "TDF P;MKREAD FFT;MKF?;" !Marker freq (Hz)
```

**MKF results with TDF set to A or I:**

If the trace data format is set to trace data format A, the result depends on the setting of the MDS command.

Program Example:

```
10 OUTPUT @Sa; "TDF A;MDS B;MKF?;" !One byte of marker position
20 OUTPUT @Sa; "TDF A;MDS W;MKF?;" !Two byte binary word (1 to 401)
```

Using the trace data format I is equivalent to the TDF A format.

**MKF results with TDF set to M:**

If the trace data format is used with trace data format M, the result is the marker horizontal position value, from 1 to 401, in ASCII.

Program Example:

```
10 OUTPUT @Sa; "TDF M;MKF?;" !Marker horizontal position (ASCII)
```

Program Example:

```
10 !Finds signal peak using markers
20 !
30 ASSIGN @Sa TO 718 ! assign IO path to spectrum analyzer
40 OUTPUT @Sa; "TP;RFCALIB ON;SNGLS;"
50 OUTPUT @Sa; "CF 50 MZ;SP 50 MZ;TS;" ! Initialize, start/stop
60 INPUT "ENTER IN PEAK EXCURSION, IN DB ", Excursion
70 OUTPUT @Sa; "MKPX ";Excursion;"DB;" ! Change peak excursion lvel
80 OUTPUT @Sa; "TS;MKPK HI;" ! Search for highest peak
```
Programming Examples

Using Markers

Program Example:

10 !Delta marker example
20 !
30 ASSIGN @Sa TO 718 ! Assign IO path to spectrum analyzer
40 OUTPUT @Sa;"IP;SNGLS;" ! Initialize SA, goto single sweep
50 INPUT "Enter the Start Frequency, in MHz",Start_freq
60 INPUT 'Enter the Stop Frequency, in MHz",Stop_freq
70 OUTPUT @Sa;"FA ";Start_freq;"MHZ;" ! Set start freq
80 OUTPUT @Sa;"FB ";Stop_freq;"MHZ;" ! Set stop freq
90 ! Update trace, find peak, and minimum. Return delta frequency
100 OUTPUT @Sa;"TS;MKPK HI;MKD;MKMIN;MKF;"!
110 ENTER @Sa;Delta_freq!Get results
120 PRINT "The difference in Frequency is ",Delta_freq,"Hz"
130 END
Using Limit Lines

Example 1

This example enters segments into the upper limit-line table, then enters a segment into the lower limit-line table. (Upper and lower limit lines are treated as separate tables). Line 50 demonstrates deleting a limit-line table.

Program Example:

```
10 ! Limit line example where the upper & lower limits are
treated as separate tables
30 !
40 ASSIGN @Sa TO 718 ! assign IO path to spectrum analyzer
50 OUTPUT @Sa;"LIMIDEL;" ! Del current table, sets fixed type
60 OUTPUT @Sa;"LIMIHALF UPPER;" ! Selects upper limit table
70 OUTPUT @Sa;"LIMIFT FREQ;" ! Limit lines based on freq
80 OUTPUT @Sa;"LIMISEG 300MHZ,-30DB,FLAT;" ! Enter segment
90 OUTPUT @Sa;"LIMIHALF LOWER;" ! Selects lower limit table
100 OUTPUT @Sa;"LIMISEG 300MHZ,-70DB,FLAT;" ! Enter segment
110 OUTPUT @Sa;"SEGDEL 1;" ! Delete segment from lower table
120 END
```
Measure Signal Bandwidth

It is often necessary to characterize a bandpass filter. A common figure of merit is its shape factor, that is the ratio of the bandwidth 3 dB down from the peak to the bandwidth 60 dB down from the peak. The following program measures shape factor.

This example finds the shape factor of a filter.

Program Example:

```cpp
! Example finds the shape factor of a filter
10  ASSIGN @Sa TO 718 ! assign IO path to spectrum analyzer
20  OUTPUT @Sa;"IP;RFCA LIB ON;SNGLS;TS;" ! Turn 50 MHz sig on
30  OUTPUT @Sa;"CF 50 MHz;SP 10 MZ;RL -25 DM;RB .3 MZ;TS;"
40  OUTPUT @Sa;"NDBPNT 1;" ! Turns on NdB points measurement
50  OUTPUT @Sa;"NDB -3DB;TS;"  ! Set to measure 3 dB bandwidth
60  OUTPUT @Sa;"NDBPNTR;" ! Query for results
70  ENTER @Sa;Three ! Store results in variable
80  OUTPUT @Sa;"NDB -60DB;TS;" ! Set to measure 60 dB bandwidth
90  OUTPUT @Sa;"NDBPNTR;" ! Query for results
100 ENTER @Sa;Sixty ! Store results in variable
110 IF Three<>-100 AND Sixty<>-100 THEN
120 PRINT "Shape factor is ",Sixty/Three ! then print shape factor
130 ELSE ! If either result couldn't be found
140 END IF
150 PRINT "Error, bandwidth could not be determined"
160 END IF
170 OUTPUT @Sa;"NDBPNT 0;" ! Turn off NdB points measurement
180 END
```

Line 130 checks the 3 dB and 60 dB bandwidth measurement results. A value of -100 indicates that the function did not find a valid signal to measure.
Measuring Noise

The system noise level often needs to be measured. This signal to noise ratio is used to quantify the noise. In communications systems, the power of the carrier signal is measured and compared to the power level of the noise floor. See the following example.

Program Example:

```
10 !Program measures carrier to noise ratio, uses noise marker
20 !
30 ASSIGN @Sa TO 718
40 OUTPUT @Sa;"IP;TS;RFCALIB ON;" ! Initialize, turn source on
50 OUTPUT @Sa;"CF 50 MZ;SF 10 MZ;SNGLS;TS;"
60 ! Marker to highest peak
70 OUTPUT @Sa;"MKPK HI;TS;"
80 ! Move marker off peak and measure noise
90 OUTPUT @Sa;"MKD UP;UP;MKNOISE ON;TS;MKA?;"
100 ENTER @Sa;C_to_n ! Put marker value into variable
110 OUTPUT @Sa;"MKNOISE OFF;" ! Turn off noise marker
120 PRINT 'Carrier to Noise Ratio in 1 Hz BW is ";C_to_n;"dB"
130 END
```
Using Amplitude Correction

You can correct the amplitude of the displayed trace data for known system conditions. Amplitude correction values can be entered for known frequencies. The trace data will be corrected at these values, with interpolation done on the values between the given points.

Compensate for frequency-dependent amplitude inaccuracies at the input.

Program Example:

```
10 ! This program uses amplitude correction
20 !
30 ASSIGN @Sa TO 718
40 DIM A$[200]
50 OUTPUT @Sa;"CF 1GHz;SP 200MHz;"
60 ! Store freq/ampl pairs, in ascending order
70 OUTPUT @Sa;"AMPCOR 100MHZ,5DB,1GHz,-5DB,1.5GHz,10DB;"
80 OUTPUT @Sa;"AMPCOR?;" ! Get amp COR value from SA
90 ENTER @Sa;A$
100 PRINT "AMPCOR ON:",A$ ! Display freq/ampl pairs
110 OUTPUT @Sa;"AMPCOR OFF;"
120 OUTPUT @Sa;"AMPCOR?;"
130 ENTER @Sa;A$
140 PRINT "AMPCOR OFF:",A$ ! AMPCOR is off, "0,0" is displayed
150 END
```
Examples Using the RS-232 Interface

The examples in this section are written for the RS-232 interface using Quick BASIC.

Reading Trace Data

The following program, which has been annotated with comments, reads a trace from the spectrum analyzer and stores the trace data in a variable.

Program Example:

```
10 'File = 232PROG5
20 OPEN "COM1:9600,N,8,1" FOR RANDOM AS #1
30 'create a 401-point trace array
40 DIM TRCA(401)
50 PRINT #1, 'IP;
60 'set output format of spectrum analyzer for real numbers
70 PRINT #1, 'TDF P;
80 'set spectrum analyzer parameters
90 PRINT #1, "SNGLS;"
100 PRINT #1, 'CF 300MZ;"
110 PRINT #1, 'SP 200MZ;"
120 PRINT #1, 'TS;"
130 PRINT #1, 'MKPK;"
140 'move peak to center of spectrum analyzer screen
150 PRINT #1, "MKCF;"
160 PRINT #1, 'TS;"
170 'ask spectrum analyzer for trace data
180 PRINT #1, 'TRA?;"
190 'retrieve trace data from spectrum analyzer
200 FOR I = 1 TO 401
210 INPUT #1, TRCA(I)
220 NEXT I
230 'set continuous-sweep mode
240 PRINT #1, 'CONTS;"
250 END
```

The program creates a 401-point trace array, called TRCA, in which the trace data will be stored. Then it uses the TRA command to request trace A data. The spectrum analyzer sends trace A data to the variable, TRCA.

Saving Trace Data

The trace data in the previous program can be stored on a computer disk. See the following example.
Examples Using the M-232 Interface

Program Example:

10 'File = 232PROG6
20 OPEN "COM1:9600,N,8,1" FOR RANDOM AS #1
30 'create a 401-point trace array
40 DIM TRCA(401)
50 PRINT #1, 'IP;
60 'set output format of spectrum analyzer for real numbers
70 PRINT #1, 'TDF P;
80 'set spectrum analyzer parameters
90 PRINT #1, "SNGLS;"
100 PRINT #1, 'CF 300MZ;"
110 PRINT #1, 'SP 200MZ;"
120 PRINT #1, "TS;"
130 PRINT #1, "MKPK;"
140 'move peak to center of spectrum analyzer screen
150 PRINT #1, "MKCF;"
160 PRINT #1, "TS;"
170 'ask spectrum analyzer for trace data
180 PRINT #1, "TRA?;"
190 'input the trace data to the BASIC program
200 FOR I = 1 TO 401
210 INPUT #1, TRCA(I) 'data input in dBm
220 NEXT I
230 'create file to store trace on disk
240 OPEN "TRACEA" FOR OUTPUT AS #2
250 'print the trace data to the disk
260 FOR I = 1 TO 401
270 PRINT #2, TRCA(I)
280 NEXT I
290 'put spectrum analyzer into continuous-sweep mode
300 PRINT #1, "CONTS;"
310 END

Using the OPEN command, we create an empty file on the disk for storing the trace and assign an output path to the file TRACEA. Then we send the trace data to the file. (See lines 260 through 280.)

Reading Trace Data from a Computer Disk

If we want to return trace data to the spectrum analyzer for later viewing, we must work the “saving” process in reverse. The following program reads a trace previously stored on a computer disk and stores the trace in an array variable.

Program Example:

10 'File = 232PROG7
20 OPEN "COM1:9600,N,8,1" FOR RANDOM AS #1
30 'create a 401-point trace array
40 DIM TRCA(401)
50 'assign number to file with trace data in it
60 OPEN "TRACEA" FOR INPUT AS #2
70 'enter the trace into the array
80 FOR I = 1 TO 401

---

86
First, in line 40, the program creates a 401-point trace array. Then, in lines 60 through 100, the program reads the disk file TRACEA and stores data in the array variable TRCA.

**Saving and Recalling Instrument States**

The spectrum analyzer’s control settings (or its “state”) can be saved with a computer and retrieved later to streamline test sequences or repeat manual measurements. Control settings can be stored in files in the spectrum analyzer, in computer memory, or on a computer disk. This program demonstrates how the state information and the trace data is read from the spectrum analyzer.

Program Example:

```plaintext
90 INPUT #2, TRCA(I)
100 NEXT I
110 CLOSE
120 END
```

First, in line 40, the program creates a 401-point trace array. Then, in lines 60 through 100, the program reads the disk file TRACEA and stores data in the array variable TRCA.
This program reads a trace and state from the spectrum analyzer. The trace information is stored in five string variables. These variables are then converted to a numeric array. The state of the spectrum analyzer is stored in the string variable LEARN$. This variable is also converted to numeric array. These two numeric arrays are then saved to a file on a disk called TRACEA.

Line 40 gives the dimensions of the learn string using the Quick BASIC DIM command. Learn strings for the spectrum analyzer require 210 bytes of storage space. Refer to the output learn string (OL) command description in Chapter 5 for more information.

Line 90 uses TDF B to format the output in binary. Binary provides the fastest data transfer and requires the least amount of memory to store data. Each data point is transferred in binary as two 8-bit bytes. The data points are in the internal representation of measurement data. (See Different Formats for Trace Data Transfers on page 55 for more information about trace data formats.)

When the trace and state data is sent from the spectrum analyzer to the computer, it must be formatted. Lines 270 through 320 format the state data.

Lines 330 to 510 format the trace data.
Returning the Spectrum Analyzer to its Former State

The following program reads a state and trace stored in a file and loads it into the spectrum analyzer.

Program Example:

10 'File = 232PROG9
20 OPEN "COM1:9600,N,8,1" FOR RANDOM AS #1
30 DEFINT I, X-Y'integer variable
40 'define 210-character string
50 DIM LEARN$(210), X1(210)
60 'create an 802-character string to store disk data
70 DIM TR1$(200), TR2$(200), TR3$(200), TR4$(200)
80 DIM TR5$(2), TR6$(200), Y1(802)
90 'open disk file "TRACEA"
100 OPEN "TRACEA" FOR INPUT AS #2
110 'enter learn array from disk
120 FOR I = 1 TO 210
130 INPUT #2, X1(I) 'get integer variable from disk
140 NEXT I
150 'enter trace data from disk
160 FOR I = 1 TO 802
170 INPUT #2, Y1(I)
180 NEXT I
190 'close the disk file
200 CLOSE #2
210 'format the integer data into strings
220 'for the spectrum analyzer. See 232PROG8 for explanation
230 LEARN$ = "" 'null out the learn string
240 FOR I = 1 TO 210 'format learn string first
250 LEARN$ = LEARN$ + CHR$(X1(I))
260 NEXT I
270 'format the trace data
280 I2 = 1 'set the counter
290 GOSUB 500 'do the conversion
300 TR1$ = TR6$ 'set the string
310 I2 = 201
320 GOSUB 500 'do the conversion
330 TR2$ = TR6$
340 I2 = 401
350 GOSUB 500
360 TR3$ = TR6$
370 I2 = 601
380 GOSUB 500
390 TR4$ = TR6$
400 'format last two characters
410 TR5$ = ""
420 TR5$ = TR5$ + CHR$(Y1(801)) + CHR$(Y1(802))
430 'write to spectrum analyzer
440 PRINT #1, LEARN$
445 SLEEP 3
450 'output trace data
460 PRINT #1, "TRA"; TR1$; TR2$; TR3$; TR4$; TR5$;
470 PRINT #1, 'VIEW TRB;'
480 GOTO 560 'end program
490 'subroutine for converting integer data to ASCII
500 TR6$ = "" 'set the string to a null value
Programming Examples
Examples Using the RS-232 Interface

510 FOR I = 1 TO 200
520 TR6$ = TR6$ + CHR$(Y1(I2))
530 I2 = I2 + 1
540 NEXT I
550 RETURN 'done with conversion
560 END

The program reads a state and trace stored in the file TRACEA. This program assumes that trace data is stored on the disk using the previous program example.

First, the settings of the spectrum analyzer that were stored in the variable LEARN$ are recalled. The spectrum analyzer state is changed to the same state as when the trace was stored. Then previously stored trace data is returned to the spectrum analyzer and the trace is viewed on the spectrum analyzer screen.

Measuring Harmonic Distortion

The harmonic distortion program presented here illustrates how the spectrum analyzer can be directed by a computer to make a complete measurement. Measuring the percent of total harmonic distortion is tedious when performed manually: it involves tuning to the fundamental and to each harmonic of interest, recording the amplitude of each signal, converting these amplitudes to linear units (volts), and calculating the result using a formula. The following program measures percent of total harmonic distortion automatically, quickly, and accurately.

The program operates as if we were making the measurement manually.

NOTE:
This program is designed to measure harmonics of a signal that is greater than 20 MHz.

Program Example:

10 'File = THDTEST
20 OPEN "COM1:9600,N,8,1" FOR RANDOM AS #1
30 'allow user to change the number of harmonics
40 MAXHARMONIC = 4
50 DIM HARMONICV(10), HARMONICDBC(10)
60 'clear the screen
70 CLS
80 'ask for the frequency of the fundamental
90 PRINT "******** HARMONIC DISTORTION ********"
100 PRINT
110 PRINT "CONNECT SOURCE TO ANALYZER INPUT, THEN"
120 PRINT "ENTER FREQUENCY OF THE FUNDAMENTAL IN MHZ"
130 PRINT
140 INPUT FUNDAMENTAL
150 CLS
160 'print measuring fundamental on screen
170 PRINT "MEASURING FUNDAMENTAL"
180 'preset the spectrum analyzer, set single-sweep
185 'take sweep
190 PRINT #1, "IP;SNGLS;TS;"
200 PRINT #1, "DONE;"
210 INPUT #1, DONE
220 'tune the spectrum analyzer to the fundamental freq and set
225 '20 MHz span
230 PRINT #1, "CF "; FUNDAMENTAL; "MHZ"
240 PRINT #1, "SP 20MZ;TS;"
250 PRINT #1, "DONE;"
260 INPUT #1, DONE
270 'put a marker on signal peak, move marker to
275 'reference level
280 PRINT #1, "MKPK HI;MKRL;TS;"
290 'find signal peak, activate signal track, and
295 'narrow span
300 PRINT #1, "MKPK HI;TS;"
310 PRINT #1, "MKTRACK ON;SP 100KZ;TS;"
320 PRINT #1, "DONE;"
330 INPUT #1, DONE
340 'turn off signal track
350 PRINT #1, "MKTRACK OFF;"
360 'find peak of signal, move peak to center of screen
370 'make units in volts
380 PRINT #1, "AUNITS V;"
390 'find peak of signal, send amplitude value to
395 'computer
400 PRINT #1, "MKPK HI;MKA?;"
410 INPUT #1, FUNDAMPTDV
420 'send marker frequency to computer, enter frequency
425 'value
430 PRINT #1, "MKF?;"
440 INPUT #1, FUNDAMENTAL
450 'make the fundamental frequency the center freq
455 'step size
460 PRINT #1, "MKSS;"
470 'set the fundamental frequency units to MHZ
480 FUNDAMENTAL = FUNDAMENTAL / 1000000!
490 FOR NUMBER = 2 TO MAXHARMONIC
500 PRINT "MEASURING HARMONIC # "; NUMBER
510 'set span and tune to next harmonic
520 PRINT #1, "SP 20MZ;"
530 PRINT #1, "CF UP;TS;"
540 PRINT #1, "DONE;"
550 INPUT #1, DONE
560 'take a second sweep to allow spectrum analyzer to move to
570 'the center frequency, find the signal peak,
580 'activate the signal track
590 PRINT #1, "TS;"
600 PRINT #1, "MKPK HI;MKRACK ON;SP 100KZ;TS;"
610 PRINT #1, "MKTRACK OFF;"
620 'find signal peak, send amplitude value to computer
630 'enter amplitude of harmonic
640 PRINT #1, "MKA?;"
650 INPUT #1, HARMONICV(NUMBER)
660 NEXT NUMBER
670 'set amplitude units to dBm
670 PRINT #1, "AUNITSDB;"
680 'calculate the fundamental amplitude in dBm because
690 'it was measured in volts
700 FUNDAMPTDDBM = 10 * (LOG(FUNDAMPTDV ^ 2 / .05) / 2.3026)
710 'calculate the sum of the squares of the amplitudes
720 'of the harmonics, calculate amplitudes of
725 'harmonics (dBm)
The program prompts the operator to connect a source to the spectrum analyzer INPUT and enter the source frequency. It sets the spectrum analyzer center frequency to the value of the source, or fundamental, frequency. It measures and records the frequency and amplitude of the fundamental, then measures and records the amplitude of the second, third, and fourth harmonics. These values are used to compute percent of harmonic distortion. The results of the harmonic distortion percentage computation, plus harmonic amplitude in dBc (decibels relative to the carrier), are displayed on the computer display.

If necessary, change the number of harmonics in line 40.
Programming Command Cross References

Command to Front-Panel on page 94
Front-Panel to Command on page 106
Functional Index on page 113
Command Backwards Compatibility on page 120
### Command to Front-Panel

<table>
<thead>
<tr>
<th>Command</th>
<th>Corresponding Key Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALIGN</td>
<td>Align Now or Auto Align</td>
<td>Controls automatic alignment of measurement systems.</td>
</tr>
<tr>
<td>AMPCOR</td>
<td>Ampcor</td>
<td>Applies amplitude corrections at specified frequencies.</td>
</tr>
<tr>
<td>ANNOT</td>
<td>Annotation On Off</td>
<td>Turns on or off the screen annotation.</td>
</tr>
<tr>
<td>APB</td>
<td>none</td>
<td>Adds trace A to trace B and sends the result to trace A.</td>
</tr>
<tr>
<td>AT</td>
<td>Attenuation Auto Man</td>
<td>Specifies RF input attenuation.</td>
</tr>
<tr>
<td>ATC</td>
<td>A → C</td>
<td>Transfers trace A into trace C.</td>
</tr>
<tr>
<td>AUNIT</td>
<td>Amptd Units</td>
<td>Specifies amplitude units for input, output, and display.</td>
</tr>
<tr>
<td>AUTO</td>
<td>Auto Couple</td>
<td>Couples the active functions automatically.</td>
</tr>
<tr>
<td>AXB</td>
<td>A ↔ B</td>
<td>Exchanges trace A and trace B.</td>
</tr>
<tr>
<td>BAUDRATE</td>
<td>Baud Rate</td>
<td>Specifies the baud rate of a spectrum analyzer with Option IAX installed in it.</td>
</tr>
<tr>
<td>BLANK</td>
<td>Blank A, Blank B, or Blank C</td>
<td>Blanks trace A, trace B, or trace C and stops taking new data into the specified trace.</td>
</tr>
<tr>
<td>BML</td>
<td>B – DL → B</td>
<td>Subtracts display line from trace B and places the result in trace B.</td>
</tr>
<tr>
<td>BRIGHT</td>
<td>(two front panel keys)</td>
<td>The two keys adjust the brightness of the display, up and down.</td>
</tr>
<tr>
<td>BTC</td>
<td>B → C</td>
<td>Transfers trace B into trace C.</td>
</tr>
<tr>
<td>BXC</td>
<td>B ↔ C</td>
<td>Exchanges trace B and trace C.</td>
</tr>
<tr>
<td>CAT</td>
<td>none</td>
<td>Displays/returns directory information from memory.</td>
</tr>
<tr>
<td>Command</td>
<td>Corresponding Key Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CF</td>
<td>Center Freq</td>
<td>Specifies center frequency.</td>
</tr>
<tr>
<td>CLRAVG</td>
<td>Video Average On Off</td>
<td>Restarts video averaging.</td>
</tr>
<tr>
<td>CLRW</td>
<td>Clear Write A, Clear Write B, or Clear Write C.</td>
<td>Clears the specified trace and enables trace data acquisition.</td>
</tr>
<tr>
<td>CLS</td>
<td>none</td>
<td>Clears all status bits.</td>
</tr>
<tr>
<td>CMDERRQ</td>
<td>none</td>
<td>Allows query of error queue</td>
</tr>
<tr>
<td>CONTRAST</td>
<td>Contrast</td>
<td>Adjusts the display contrast.</td>
</tr>
<tr>
<td>CONTS</td>
<td>Sweep Cont Single (Cont)</td>
<td>Sets the spectrum analyzer to the continuous sweep mode.</td>
</tr>
<tr>
<td>CORREK</td>
<td>none</td>
<td>Query the box for the state of corrections</td>
</tr>
<tr>
<td>DATASTAT</td>
<td>none</td>
<td>Indicates certain conditions of the instrument.</td>
</tr>
<tr>
<td>DATEMODE</td>
<td>Datemode MDY DMY</td>
<td>Allows you to set the format for displaying the real-time clock.</td>
</tr>
<tr>
<td>DEFCONFIG</td>
<td>Default Config</td>
<td>Resets the analyzer to the user configuration originally set at the factory.</td>
</tr>
<tr>
<td>DELETE</td>
<td>Delete</td>
<td>Deletes a file from memory.</td>
</tr>
<tr>
<td>DEMOD</td>
<td>Demod</td>
<td>Turns the demodulator on or off, and selects between AM, FM, or quasi-peak demodulation.</td>
</tr>
<tr>
<td>DEMODT</td>
<td>Dwell Time On Off</td>
<td>Sets demodulation time.</td>
</tr>
<tr>
<td>DET</td>
<td>Detector</td>
<td>Selects the spectrum analyzer detection mode.</td>
</tr>
<tr>
<td>DL</td>
<td>Display Line On Off</td>
<td>Defines the level of the display line in the active amplitude units and displays the display line on the spectrum analyzer screen.</td>
</tr>
<tr>
<td>DN</td>
<td>↓ key</td>
<td>Reduces the active function by the applicable step size.</td>
</tr>
<tr>
<td>Command</td>
<td>Corresponding Key Function</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DONE</td>
<td>Done</td>
<td>Allows you to determine when the <strong>spectrum analyzer</strong> has started to execute all commands prior to and including DONE.</td>
</tr>
<tr>
<td>ERASE</td>
<td>none</td>
<td>Purges all state and trace registers and deletes limit-lines and ampcor information.</td>
</tr>
<tr>
<td>ERR</td>
<td>none</td>
<td>Returns power-on test results.</td>
</tr>
<tr>
<td>EXITSHOWSYS</td>
<td>none</td>
<td>Exits show system information. See SHOWSYS.</td>
</tr>
<tr>
<td>FA</td>
<td>Start Freq</td>
<td>Specifies the start frequency.</td>
</tr>
<tr>
<td>FB</td>
<td>Stop Freq</td>
<td>Specifies the stop frequency.</td>
</tr>
<tr>
<td>FOFFSET</td>
<td>Freq Offset</td>
<td>Specifies the frequency offset for all absolute frequency readouts such as center frequency.</td>
</tr>
<tr>
<td>FREF</td>
<td>none</td>
<td>Returns the source of the 10 MHz frequency reference.</td>
</tr>
<tr>
<td>FS</td>
<td>Full Span</td>
<td>Sets the frequency span of the spectrum analyzer to full span.</td>
</tr>
<tr>
<td>GETPRNT</td>
<td>Print</td>
<td>Initiates output of the spectrum analyzer display to a printer.</td>
</tr>
<tr>
<td>GRAT</td>
<td>Graticule On Off</td>
<td>Turns the graticule on or off.</td>
</tr>
<tr>
<td>HD</td>
<td>Esc</td>
<td>Disables data entry via the spectrum analyzer numeric keypad, knob, or step keys. The active function readout is blanked, and any active function is deactivated.</td>
</tr>
<tr>
<td>ID</td>
<td>Show System</td>
<td>Returns the spectrum analyzer model number.</td>
</tr>
<tr>
<td>INVERTLCD</td>
<td>Invert</td>
<td>Inverts the display. (Use with an external display.)</td>
</tr>
<tr>
<td>INZ</td>
<td>Input Z</td>
<td>Specifies the value of input impedance expected at the active input port.</td>
</tr>
<tr>
<td>IP</td>
<td>Preset</td>
<td>Performs an instrument preset.</td>
</tr>
</tbody>
</table>

---

96
<table>
<thead>
<tr>
<th>Command</th>
<th>Corresponding Key Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG</td>
<td>Scale Type Log Lin (Log)</td>
<td>Specifies the vertical graticule divisions as logarithmic units, without changing the reference level.</td>
</tr>
<tr>
<td>LIMIDEL</td>
<td>Delete Limits</td>
<td>Deletes all segments in the current limit-line table.</td>
</tr>
<tr>
<td>LIMIDISP</td>
<td>Limit Display Y N Auto</td>
<td>Controls when the limit line (or limit lines) are displayed.</td>
</tr>
<tr>
<td>LIMIFAIL</td>
<td>Limit Test On Off</td>
<td>Returns a &quot;0&quot; if the last measurement sweep of trace A is equal to or within the limit-line bounds.</td>
</tr>
<tr>
<td>LIMIFT</td>
<td>X Axis Units Freq Time</td>
<td>Selects how the limit-line segments are placed on the spectrum analyzer display, according to frequency, or according to the sweep time setting of the spectrum analyzer.</td>
</tr>
<tr>
<td>LIMIHALF</td>
<td>Select Line Upper Lower</td>
<td>Selects the upper or lower limit-line for editing.</td>
</tr>
<tr>
<td>LIMILINE</td>
<td>Edit Limits</td>
<td>Accesses limit-line functions.</td>
</tr>
<tr>
<td>LIMIREL</td>
<td>Limits Fixed Rel</td>
<td>Specifies the current limit lines as fixed or relative.</td>
</tr>
<tr>
<td>LIMISEG</td>
<td>Edit Line</td>
<td>Adds new segments to the current frequency limit line in either the upper limit line or the lower limit line.</td>
</tr>
<tr>
<td>LIMISEGT</td>
<td>Edit Line (time limit lines)</td>
<td>Adds new segments to the current sweep time limit line in either the upper limit line or the lower limit line.</td>
</tr>
<tr>
<td>LIMITEST</td>
<td>Limit Test On Off</td>
<td>Compares trace A with the current limit-line data.</td>
</tr>
<tr>
<td>LN</td>
<td>Scale Type Log Lin (Lin)</td>
<td>Specifies the vertical graticule divisions as linear units, without changing the reference level.</td>
</tr>
<tr>
<td>LOAD</td>
<td>Load</td>
<td>to load a trace, amp, limit, state</td>
</tr>
<tr>
<td>LSPAN</td>
<td>Last Span</td>
<td>Changes the spectrum analyzer’s span to the previous span setting.</td>
</tr>
<tr>
<td>Command</td>
<td>Corresponding Key Function</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MDS</td>
<td>none</td>
<td>Specifies measurement data size as byte or word.</td>
</tr>
<tr>
<td>MDU</td>
<td>none</td>
<td>Returns values for the spectrum analyzer's baseline and reference level.</td>
</tr>
<tr>
<td>MEAN</td>
<td>none</td>
<td>Returns the mean value of the given trace in measurement units.</td>
</tr>
<tr>
<td>MEANTH</td>
<td>none</td>
<td>Returns the mean value of the given trace above the threshold, in measurement units.</td>
</tr>
<tr>
<td>MF</td>
<td>Mkr Readout</td>
<td>Returns the frequency (or time) of the on-screen active marker.</td>
</tr>
<tr>
<td>MINH</td>
<td>Min Hold C</td>
<td>Updates trace C elements with minimum level detected.</td>
</tr>
<tr>
<td>MKA</td>
<td>none</td>
<td>Specifies amplitude of the active marker.</td>
</tr>
<tr>
<td>MKACT</td>
<td>Select Marker 12 3 4</td>
<td>Specifies the active marker.</td>
</tr>
<tr>
<td>MKBW</td>
<td>N dB Points On Off (On)</td>
<td>Returns the bandwidth at the specified power level relative to an on-screen marker (if present) or the signal peak (if no on-screen marker is present).</td>
</tr>
<tr>
<td>MKCF</td>
<td>Mkr → CF</td>
<td>Sets the center frequency equal to the marker frequency and moves the marker to the center of the screen.</td>
</tr>
<tr>
<td>MKD</td>
<td>Marker A</td>
<td>Activates the delta marker.</td>
</tr>
<tr>
<td>MKF</td>
<td>none</td>
<td>Specifies the frequency value of the active marker.</td>
</tr>
<tr>
<td>MKFA</td>
<td>Mkr → Start</td>
<td>Sets the start frequency to the frequency of the active marker.</td>
</tr>
<tr>
<td>MKFB</td>
<td>Mkr → Stop</td>
<td>Sets the stop frequency to the frequency of the active marker.</td>
</tr>
<tr>
<td>MKFC</td>
<td>Marker Count On Off</td>
<td>Turns on or off marker frequency counter.</td>
</tr>
<tr>
<td>Command</td>
<td>Corresponding Key Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MKFCR</td>
<td>Resolution Auto Man</td>
<td>Sets the resolution of the marker frequency counter.</td>
</tr>
<tr>
<td>MKMIN</td>
<td>Min Search</td>
<td>Moves active marker to minimum signal detected.</td>
</tr>
<tr>
<td>MKN</td>
<td>Marker Normal</td>
<td>Activates and moves the marker to the specified frequency.</td>
</tr>
<tr>
<td>MKNOISE</td>
<td>Marker Noise On Off (On)</td>
<td>Displays the average noise level at the marker.</td>
</tr>
<tr>
<td>MKOFF</td>
<td>Marker All Off</td>
<td>Turns off either the active marker or all the markers.</td>
</tr>
<tr>
<td>MKP</td>
<td>none</td>
<td>Places the active marker at the given x-coordinate.</td>
</tr>
<tr>
<td>MKPAUSE</td>
<td>Dwell Time On Off</td>
<td>Pauses the sweep at the active marker for the duration of the delay period.</td>
</tr>
<tr>
<td>MKPK</td>
<td>Peak Search</td>
<td>Positions the active marker on a signal peak.</td>
</tr>
<tr>
<td>MKPP</td>
<td>Pk-Pk Search</td>
<td>Finds and displays the frequency and amplitude differences between the highest and lowest trace points.</td>
</tr>
<tr>
<td>MKPX</td>
<td>Peak Excursn</td>
<td>Specifies the minimum signal excursion for the spectrum analyzer’s internal peak-identification routine.</td>
</tr>
<tr>
<td>MKREAD</td>
<td>Mkr Readout</td>
<td>Selects the type of active trace information displayed by the spectrum analyzer marker readout.</td>
</tr>
<tr>
<td>MKRL</td>
<td>Mkr (\rightarrow) Ref Lvl</td>
<td>Sets the reference level to the amplitude value of the active marker.</td>
</tr>
<tr>
<td>MKSP</td>
<td>Mkr A (\rightarrow) Span</td>
<td>Sets the start and stop frequencies to the values of the delta markers.</td>
</tr>
<tr>
<td>MKSS</td>
<td>Mkr (\rightarrow) CF Step</td>
<td>Sets the center-frequency step-size to the marker frequency.</td>
</tr>
<tr>
<td>MKTH</td>
<td>Pk Threshold</td>
<td>Sets a lower boundary to the active trace.</td>
</tr>
<tr>
<td><strong>Command</strong></td>
<td><strong>Corresponding Key Function</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>MKTRACE</td>
<td>Marker Trace Auto A B C</td>
<td>Moves the active marker to a corresponding position in trace A, trace B, or trace C.</td>
</tr>
<tr>
<td>MKTRACK</td>
<td>Signal Track On Off (On)</td>
<td>Moves the signal with an active marker to the center of the spectrum analyzer display and keeps the signal peak at center screen.</td>
</tr>
<tr>
<td>ML</td>
<td>Max Mixer Lvl</td>
<td>Specifies the maximum signal level that is applied to the input mixer for a signal that is equal to or below the reference level.</td>
</tr>
<tr>
<td>MOV</td>
<td>none</td>
<td>Copies the source values into the destination.</td>
</tr>
<tr>
<td>MXMH</td>
<td>Max Hold A or Max Hold B</td>
<td>Updates trace elements with maximum level detected.</td>
</tr>
<tr>
<td>NDB</td>
<td>N dB Points On Off (On)</td>
<td>Specifies the distance (in dB) from the signal peak for the N dB points measurement (NDBPNT).</td>
</tr>
<tr>
<td>NDBPNT</td>
<td>N dB Points On Off</td>
<td>Turns on or off the N dB points measurement.</td>
</tr>
<tr>
<td>NDBPNTR</td>
<td>none</td>
<td>Returns the bandwidth measured by the N dB points measurement (NDBPT).</td>
</tr>
<tr>
<td>NRMCLZ</td>
<td>Normalize</td>
<td>Normalizes the current data with the values previously stored in Trace A.</td>
</tr>
<tr>
<td>NRL</td>
<td>Norm Ref Lvl</td>
<td>Sets the normalized reference level.</td>
</tr>
<tr>
<td>NRPOS</td>
<td>Normal Position</td>
<td>Sets the position of the normalized reference.</td>
</tr>
<tr>
<td>OL</td>
<td>none</td>
<td>Output current state in learn string format.</td>
</tr>
<tr>
<td>PCTAM</td>
<td>%AM On Off</td>
<td>Turns on or off the percent AM measurement.</td>
</tr>
<tr>
<td>PCTAMR</td>
<td>%AM On Off (On)</td>
<td>Returns the percent AM measured by the percent AM measurement (PCTAM).</td>
</tr>
<tr>
<td>Command</td>
<td>Corresponding Key Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PEAKS</td>
<td>none</td>
<td>Sorts signal peaks by frequency or amplitude, stores the results in the destination trace, and returns the number of peaks found.</td>
</tr>
<tr>
<td>POWERON</td>
<td>Power On IP Last</td>
<td>Selects the state the spectrum analyzer will be in when it is turned on: IP(instrument preset) or LAST state.</td>
</tr>
<tr>
<td>PREAMPG</td>
<td>Ext Amp Gain</td>
<td>Subtracts a positive or negative preamplifier gain value from the displayed signal.</td>
</tr>
<tr>
<td>PRINT</td>
<td>none</td>
<td>Initiates output of the spectrum analyzer display to a controller.</td>
</tr>
<tr>
<td>PRNPRT</td>
<td>Port</td>
<td>Directs the printer output to HP-IB, serial or parallel ports.</td>
</tr>
<tr>
<td>PRNTADRS</td>
<td>Printer Addr</td>
<td>Allows you to set the HP-IB address of the printer.</td>
</tr>
<tr>
<td>PRNTMARGB</td>
<td>Bottom Margin</td>
<td>Adjusts the white space below a print.</td>
</tr>
<tr>
<td>PRNTMARGT</td>
<td>Top Margin</td>
<td>Adjusts the white space above a print.</td>
</tr>
<tr>
<td>PRNTRES</td>
<td>Resolution</td>
<td>Selects the resolution number to be sent to the printer.</td>
</tr>
<tr>
<td>PSTATE</td>
<td>Internal Lock On Off (On)</td>
<td>Protects all of the spectrum analyzer’s user state and trace registers from being changed.</td>
</tr>
<tr>
<td>PWRBW</td>
<td>none</td>
<td>Computes the bandwidth around the trace center, which includes signals whose total power is a specified percentage of the total trace signal power.</td>
</tr>
<tr>
<td>PWRUPTIME</td>
<td>none</td>
<td>Returns the number of milliseconds that have elapsed since the spectrum analyzer was turned on.</td>
</tr>
<tr>
<td>RB</td>
<td>Resolution BW Auto Man</td>
<td>Specifies the resolution bandwidth.</td>
</tr>
<tr>
<td>Command</td>
<td>Corresponding Key Function</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>RCLS</td>
<td>Load</td>
<td>Recalls spectrum analyzer state data from one of nine state registers in spectrum analyzer memory. These registers do not appear in a FILE catalog.</td>
</tr>
<tr>
<td>RCLT</td>
<td>Load</td>
<td>Recalls previously saved trace data, amplitude factors, or limit-line data from the trace registers in spectrum analyzer memory. These registers are specially mapped to named files.</td>
</tr>
<tr>
<td>REV</td>
<td>Show System</td>
<td>Returns the date code of the firmware revision number in YYYYMMDD format.</td>
</tr>
<tr>
<td>RFCALIB</td>
<td>50 MHz osc On Off</td>
<td>Turns the internal 50 MHz alignment signal on or off.</td>
</tr>
<tr>
<td>RL</td>
<td>Ref Level</td>
<td>Specifies the amplitude value of the reference level.</td>
</tr>
<tr>
<td>RMS</td>
<td>none</td>
<td>Returns the root mean square value of the trace in measurement units.</td>
</tr>
<tr>
<td>ROFFSET</td>
<td>Ref Lvl Offst</td>
<td>Offsets all amplitude readouts without affecting the trace.</td>
</tr>
<tr>
<td>RQS</td>
<td>none</td>
<td>Sets a bit mask for service requests.</td>
</tr>
<tr>
<td>SAVE</td>
<td>Save</td>
<td>Saves a specified file in spectrum analyzer memory.</td>
</tr>
<tr>
<td>SAVES</td>
<td>Save</td>
<td>Saves the currently displayed instrument state in spectrum analyzer memory. These registers do not appear in a FILE catalog.</td>
</tr>
<tr>
<td>SEGDEL</td>
<td>Del Segment</td>
<td>Deletes the specified segment from the limit-line tables.</td>
</tr>
<tr>
<td>SER</td>
<td>Show System</td>
<td>Returns the serial number suffix of the spectrum analyzer.</td>
</tr>
<tr>
<td>SETDATE</td>
<td>Set Date</td>
<td>Sets the date of the real-time clock.</td>
</tr>
<tr>
<td>SET-TIME</td>
<td>Set Time</td>
<td>Sets the time of the real-time clock.</td>
</tr>
<tr>
<td>Command</td>
<td>Corresponding Key Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SHOWSYS</td>
<td>Show System</td>
<td>Displays the model number, product number, serial number, firmware revision, and options that are installed in the spectrum analyzer.</td>
</tr>
<tr>
<td>SMOOTH</td>
<td>Video</td>
<td>Smooths the trace according to the number of points specified for the running average.</td>
</tr>
<tr>
<td>SNGLS</td>
<td>Sweep Cont Single (Single)</td>
<td>Selects single-sweep mode.</td>
</tr>
<tr>
<td>SP</td>
<td>Span</td>
<td>Changes the total displayed frequency range symmetrically about the center frequency.</td>
</tr>
<tr>
<td>SPEAKER</td>
<td>Speaker On Off</td>
<td>Turns on or off the internal speaker.</td>
</tr>
<tr>
<td>SPZOOM</td>
<td>Span Zoom</td>
<td>Places a marker on the highest on-screen signal (if an on-screen marker is not present), turns on the signal track function, and activates the span function.</td>
</tr>
<tr>
<td>SQR</td>
<td>none</td>
<td>Places the square root of the source into the destination.</td>
</tr>
<tr>
<td>SRCAT</td>
<td>Attenuation Auto Man</td>
<td>Attenuates the source output level.</td>
</tr>
<tr>
<td>SRCPOFS</td>
<td>Amptd Offset</td>
<td>Offsets the source power level readout.</td>
</tr>
<tr>
<td>SRCPSTP</td>
<td>Amptd Step Auto Man</td>
<td>Selects the source-power step size.</td>
</tr>
<tr>
<td>SRCPSWP</td>
<td>Power Sweep On Off</td>
<td>Selects sweep range of the source output.</td>
</tr>
<tr>
<td>SRCPWR</td>
<td>Amplitude On Off</td>
<td>Selects the source power level.</td>
</tr>
<tr>
<td>SRQ</td>
<td>none</td>
<td>Used by an external controller to simulate interrupts from the spectrum analyzer.</td>
</tr>
<tr>
<td>SS</td>
<td>CF Step Auto Man</td>
<td>Specifies center-frequency step size.</td>
</tr>
<tr>
<td>ST</td>
<td>Sweep Time Auto Man</td>
<td>Specifies the time in which the spectrum analyzer sweeps the displayed frequency range.</td>
</tr>
<tr>
<td>Command</td>
<td>Corresponding Key Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>STB</td>
<td>none</td>
<td>Returns to the controller the decimal equivalent of the status byte.</td>
</tr>
<tr>
<td>STDEV</td>
<td>none</td>
<td>Returns the standard deviation of the trace amplitude in measurement units.</td>
</tr>
<tr>
<td>SUM</td>
<td>none</td>
<td>Returns the sum of the amplitudes of the trace elements in measurement units.</td>
</tr>
<tr>
<td>SUMSQR</td>
<td>none</td>
<td>Returns the sum of the squares of the amplitude of each trace element.</td>
</tr>
<tr>
<td>SWPCPL</td>
<td>Swp Coupling SR SA</td>
<td>Selects a stimulus-response (SR) or spectrum-analyzer (SA) auto-coupled sweep time.</td>
</tr>
<tr>
<td>TDF</td>
<td>none</td>
<td>Formats trace information for return to the controller.</td>
</tr>
<tr>
<td>TH</td>
<td>Threshold On Off</td>
<td>Clips signal responses below the threshold level.</td>
</tr>
<tr>
<td>TIMEBASEC</td>
<td>Timebase Coarse</td>
<td>Controls coarse tuning of the timebase frequency.</td>
</tr>
<tr>
<td>TIMEBASEF</td>
<td>Timebase Fine</td>
<td>Controls fine tuning of the timebase frequency.</td>
</tr>
<tr>
<td>TIMEDATE</td>
<td>Time/Date</td>
<td>Sets the time and date of the real-time clock.</td>
</tr>
<tr>
<td>TIMEDSP</td>
<td>Time/Date On Off</td>
<td>Turns on or off the display of the real-time clock.</td>
</tr>
<tr>
<td>TITLE</td>
<td>Change Title</td>
<td>Activates the screen title mode.</td>
</tr>
<tr>
<td>TM</td>
<td>Trig</td>
<td>Specifies trigger mode.</td>
</tr>
<tr>
<td>TOI</td>
<td>TOI On Off</td>
<td>Turns on or off the third-order intermodulation (TOI) measurement.</td>
</tr>
<tr>
<td>TOIR</td>
<td>none</td>
<td>Returns the highest third-order intermodulation product measured by the third-order intermodulation measurement(TOI).</td>
</tr>
<tr>
<td>TRA, TRB, TRC</td>
<td>Trace A, Trace B, or Trace C</td>
<td>Controls trace data input or output.</td>
</tr>
<tr>
<td>Command</td>
<td>Corresponding Key Function</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>TRSTAT</td>
<td>none</td>
<td>Returns the status of traces A, B, and C: clear write, blank, view, minimum hold, or maximum hold.</td>
</tr>
<tr>
<td>TS</td>
<td>Sweep Cont Single (Single)</td>
<td>Starts and completes one full sweep before the next command is executed.</td>
</tr>
<tr>
<td>UP</td>
<td>↑ key</td>
<td>Increases the active function by the applicable step size.</td>
</tr>
<tr>
<td>VARIANCE</td>
<td>none</td>
<td>Returns the amplitude variance of the specified trace, in measurement units.</td>
</tr>
<tr>
<td>VAVG</td>
<td>Video Average On Off</td>
<td>Enables the video-averaging function, which averages trace points to smooth the displayed trace.</td>
</tr>
<tr>
<td>VB</td>
<td>Video BW Auto Man</td>
<td>Specifies the video bandwidth.</td>
</tr>
<tr>
<td>VBR</td>
<td>VBW/RBW Ratio</td>
<td>Specifies coupling ratio of video bandwidth to resolution bandwidth.</td>
</tr>
<tr>
<td>VIEW</td>
<td>View</td>
<td>Displays trace A, trace B, or trace C, and stops taking new data into the viewed trace.</td>
</tr>
<tr>
<td>XCH</td>
<td>A ↔ B, B ↔ C</td>
<td>Exchanges traces.</td>
</tr>
</tbody>
</table>
## Front-Panel to Command

<table>
<thead>
<tr>
<th>Key Label</th>
<th>Remote Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>% AM On Off</td>
<td>PCTAM</td>
</tr>
<tr>
<td>50 MHz osc On Off</td>
<td>RFCALIB</td>
</tr>
<tr>
<td>Align Now, RF</td>
<td>RFCALIB</td>
</tr>
<tr>
<td>A ↔ B</td>
<td>AXB</td>
</tr>
<tr>
<td>A → C</td>
<td>ATC</td>
</tr>
<tr>
<td>Align Now</td>
<td>ALIGN</td>
</tr>
<tr>
<td>Alignments</td>
<td>ALIGN</td>
</tr>
<tr>
<td>All Memory</td>
<td>ERASE</td>
</tr>
<tr>
<td>AM</td>
<td>DEMOD</td>
</tr>
<tr>
<td>Ampcor</td>
<td>AMPCOR</td>
</tr>
<tr>
<td>Amplitude On Off</td>
<td>SRCPWR</td>
</tr>
<tr>
<td>Amptd Offset</td>
<td>SRCPOFS</td>
</tr>
<tr>
<td>Amptd Step Auto Man</td>
<td>SRCPSTP</td>
</tr>
<tr>
<td>Amptd Units</td>
<td>AUNITS</td>
</tr>
<tr>
<td>Annotation On Off</td>
<td>ANNOT</td>
</tr>
<tr>
<td>Attenuation Auto Man</td>
<td>AT, SRCAT</td>
</tr>
<tr>
<td>Auto Align</td>
<td>ALIGN</td>
</tr>
<tr>
<td>Auto Couple</td>
<td>AUTO</td>
</tr>
<tr>
<td>B ↔ C</td>
<td>BXC</td>
</tr>
<tr>
<td>B → C</td>
<td>BTC</td>
</tr>
<tr>
<td>B → DL → B</td>
<td>BML</td>
</tr>
<tr>
<td>Baud Rate</td>
<td>BAUDRATE</td>
</tr>
<tr>
<td>Blank A, B, C</td>
<td>BLANK</td>
</tr>
<tr>
<td>Key Label</td>
<td>Remote Command</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Bottom Margin</td>
<td>PRNTMARGB</td>
</tr>
<tr>
<td>brightness keys</td>
<td>BRIGHT</td>
</tr>
<tr>
<td>BW/Avg</td>
<td>RB, VAVG</td>
</tr>
<tr>
<td>Center Freq</td>
<td>CF</td>
</tr>
<tr>
<td>CF Step Auto Man</td>
<td>SS</td>
</tr>
<tr>
<td>Change Title</td>
<td>TITLE</td>
</tr>
<tr>
<td>Clear Title</td>
<td>TITLE</td>
</tr>
<tr>
<td>Clear Write A, B, C</td>
<td>CLRW</td>
</tr>
<tr>
<td>Color Printing On Off</td>
<td>PRINT</td>
</tr>
<tr>
<td>Contrast</td>
<td>CONTRAST</td>
</tr>
<tr>
<td>Datemode MDY DMY</td>
<td>DATEMODE</td>
</tr>
<tr>
<td>Default Config</td>
<td>DEFCONFIG</td>
</tr>
<tr>
<td>Define Printer</td>
<td>PRNTMARGB, PRNTMARGT, PRNTRES</td>
</tr>
<tr>
<td>Del Segment</td>
<td>SEGDEL</td>
</tr>
<tr>
<td>Delete</td>
<td>DELETE</td>
</tr>
<tr>
<td>Delete Limits</td>
<td>LIMIDEL</td>
</tr>
<tr>
<td>Demod</td>
<td>DEMOD</td>
</tr>
<tr>
<td>Detector</td>
<td>DET</td>
</tr>
<tr>
<td>Display Line On Off</td>
<td>DL</td>
</tr>
<tr>
<td>Dwell Time On Off</td>
<td>DEMODT, MKPAUSE</td>
</tr>
<tr>
<td>Edit Limits</td>
<td>LIMILINE</td>
</tr>
<tr>
<td>Edit Line</td>
<td>LIMILINE</td>
</tr>
<tr>
<td>Erase Internal</td>
<td>ERASE</td>
</tr>
<tr>
<td>Esc</td>
<td>HD</td>
</tr>
<tr>
<td>Execute Title</td>
<td>TITLE</td>
</tr>
<tr>
<td>Key Label</td>
<td>Remote Command</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Expand On Off</td>
<td>none</td>
</tr>
<tr>
<td>Ext Amp Gain</td>
<td>PREAMPG</td>
</tr>
<tr>
<td>External</td>
<td>TM</td>
</tr>
<tr>
<td>Form Feed</td>
<td>none</td>
</tr>
<tr>
<td>Free Run</td>
<td>TM</td>
</tr>
<tr>
<td>Freq Correct On Off</td>
<td>ALIGN</td>
</tr>
<tr>
<td>Freq Count</td>
<td>MKFC</td>
</tr>
<tr>
<td>Freq Offset</td>
<td>FOFFSET</td>
</tr>
<tr>
<td>Frequency</td>
<td>CF, FA, FB</td>
</tr>
<tr>
<td>Full Span</td>
<td>FS</td>
</tr>
<tr>
<td>Graticule On Off</td>
<td>GRAT</td>
</tr>
<tr>
<td>HP-IB</td>
<td>PRNPRT, PRNTADRS</td>
</tr>
<tr>
<td>HP Color Mode PJ DJ Off</td>
<td>PRINT</td>
</tr>
<tr>
<td>Input Z Corr 50 75</td>
<td>INZ</td>
</tr>
<tr>
<td>Internal Lock On Off</td>
<td>PSTATE</td>
</tr>
<tr>
<td>Inverse Time</td>
<td>MKREAD</td>
</tr>
<tr>
<td>Inverse Video On Off</td>
<td>INVERTLCD</td>
</tr>
<tr>
<td>Last Span</td>
<td>LSPAN</td>
</tr>
<tr>
<td>Limit Display Y N Auto</td>
<td>LIMIDISP</td>
</tr>
<tr>
<td>Limit Test On Off</td>
<td>LIMIEST</td>
</tr>
<tr>
<td>Limits</td>
<td>LIMILINE</td>
</tr>
<tr>
<td>Limits Fixed Rel</td>
<td>LIMIREL</td>
</tr>
<tr>
<td>Line</td>
<td>TM</td>
</tr>
<tr>
<td>Load</td>
<td>LOAD</td>
</tr>
<tr>
<td>Load Defaults</td>
<td>CORREK</td>
</tr>
<tr>
<td>Marker Δ</td>
<td>MKD</td>
</tr>
<tr>
<td>Key Label</td>
<td>Remote Command</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Marker # On Off</td>
<td>MKACT, MKOFF</td>
</tr>
<tr>
<td>Marker All Off</td>
<td>MKOFF</td>
</tr>
<tr>
<td>Marker Count On Off</td>
<td>MKFC</td>
</tr>
<tr>
<td>Marker Noise On Off</td>
<td>MKNOISE</td>
</tr>
<tr>
<td>Marker Normal</td>
<td>MKN</td>
</tr>
<tr>
<td>Marker Trace Auto A B C</td>
<td>MKTRACE</td>
</tr>
<tr>
<td>Max Hold A, B</td>
<td>MXMH</td>
</tr>
<tr>
<td>Max Mixer Lvl</td>
<td>ML</td>
</tr>
<tr>
<td>Max Pk → CF</td>
<td>none</td>
</tr>
<tr>
<td>Min Hold C</td>
<td>MINH</td>
</tr>
<tr>
<td>Min Search</td>
<td>MKMIN</td>
</tr>
<tr>
<td>Mkr △ → Span</td>
<td>MKSP</td>
</tr>
<tr>
<td>Mkr → CF</td>
<td>MKCF</td>
</tr>
<tr>
<td>Mkr → CF Step</td>
<td>MKSS</td>
</tr>
<tr>
<td>Mkr → Ref Lvl</td>
<td>MKRL</td>
</tr>
<tr>
<td>Mkr → Start</td>
<td>MKFA</td>
</tr>
<tr>
<td>Mkr → Stop</td>
<td>MKFB</td>
</tr>
<tr>
<td>Mkr Readout</td>
<td>MKREAD</td>
</tr>
<tr>
<td>N dB Points On Off</td>
<td>NDB, NDBPNT</td>
</tr>
<tr>
<td>Negative Peak</td>
<td>DET</td>
</tr>
<tr>
<td>New Filename</td>
<td>SAVE</td>
</tr>
<tr>
<td>Next Peak</td>
<td>MKPK</td>
</tr>
<tr>
<td>Next Pk Left</td>
<td>MKPK</td>
</tr>
<tr>
<td>Next Pk Right</td>
<td>MKPK</td>
</tr>
<tr>
<td>Norm Ref Lvl</td>
<td>NRL</td>
</tr>
<tr>
<td>Norm Ref Posn</td>
<td>NRPOS</td>
</tr>
<tr>
<td>Key Label</td>
<td>Remote Command</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Normalize On Off</td>
<td>NORMLIZE</td>
</tr>
<tr>
<td>Parallel</td>
<td>PRNPRT</td>
</tr>
<tr>
<td>Peak Excursn</td>
<td>MKPX</td>
</tr>
<tr>
<td>Peak Search</td>
<td>MKPK</td>
</tr>
<tr>
<td>Period</td>
<td>MKREAD</td>
</tr>
<tr>
<td>Pk-Pk Search</td>
<td>MKPP</td>
</tr>
<tr>
<td>Pk Threshold</td>
<td>MKTH</td>
</tr>
<tr>
<td>Port</td>
<td>PRNPRT</td>
</tr>
<tr>
<td>Power On IP Last</td>
<td>POWERON</td>
</tr>
<tr>
<td>Power Sweep On Off</td>
<td>SRCPSWP</td>
</tr>
<tr>
<td>Preset</td>
<td>IP</td>
</tr>
<tr>
<td>Print</td>
<td>PRINT</td>
</tr>
<tr>
<td>Print Softkeys On Off</td>
<td>none</td>
</tr>
<tr>
<td>Printer Addr</td>
<td>PRNTADRS</td>
</tr>
<tr>
<td>Purge Ampcor</td>
<td>AMFCOR</td>
</tr>
<tr>
<td>Ref Lvl Offst</td>
<td>ROFFSET</td>
</tr>
<tr>
<td>Ref Level</td>
<td>RL</td>
</tr>
<tr>
<td>Resolution</td>
<td>MKFCR, PRNTRES</td>
</tr>
<tr>
<td>Resolution Auto Man</td>
<td>MKFCR</td>
</tr>
<tr>
<td>Resolution BW Auto Man</td>
<td>RB</td>
</tr>
<tr>
<td>Sample</td>
<td>DET</td>
</tr>
<tr>
<td>Save</td>
<td>SAVE</td>
</tr>
<tr>
<td>Scale/Div</td>
<td>LG</td>
</tr>
<tr>
<td>Scale Type Log Lin</td>
<td>LG, LN</td>
</tr>
<tr>
<td>Search</td>
<td>MKPK</td>
</tr>
<tr>
<td>Select Line Upper Lower</td>
<td>LIMIHALF</td>
</tr>
<tr>
<td>Key Label</td>
<td>Remote Command</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Select Marker 1 2 3 4</td>
<td>MKACT</td>
</tr>
<tr>
<td>Serial</td>
<td>PRNPRT, BAUDRATE</td>
</tr>
<tr>
<td>Set Date</td>
<td>SETDATE</td>
</tr>
<tr>
<td>Set Time</td>
<td>SETTIME</td>
</tr>
<tr>
<td>Show System</td>
<td>ID, REV, SER, EXITSHOWSYS</td>
</tr>
<tr>
<td>Signal Track On Off</td>
<td>MKTRACK</td>
</tr>
<tr>
<td>Single Sweep</td>
<td>SNGLS</td>
</tr>
<tr>
<td>Source Amptd</td>
<td>SRCPWR</td>
</tr>
<tr>
<td>Span</td>
<td>SP</td>
</tr>
<tr>
<td>Span Zoom</td>
<td>SPZOOM</td>
</tr>
<tr>
<td>Speaker On Off</td>
<td>SPEAKER</td>
</tr>
<tr>
<td>Start Freq</td>
<td>FA</td>
</tr>
<tr>
<td>Stop Freq</td>
<td>FB</td>
</tr>
<tr>
<td>Sweep Cont Single</td>
<td>CONTS, SNGLS, TS</td>
</tr>
<tr>
<td>Sweep Time Auto Man</td>
<td>ST</td>
</tr>
<tr>
<td>Swp Coupling SR SA</td>
<td>SWPCPL</td>
</tr>
<tr>
<td>Threshold On Off</td>
<td>TH</td>
</tr>
<tr>
<td>Time</td>
<td>MKREAD</td>
</tr>
<tr>
<td>Time/Date</td>
<td>SETTIME, SETDATE</td>
</tr>
<tr>
<td>Time/Date On Off</td>
<td>TIMEDSP</td>
</tr>
<tr>
<td>Timebase</td>
<td>TIMEBASEC, TIMEBASEF</td>
</tr>
<tr>
<td>Title</td>
<td>TITLE</td>
</tr>
<tr>
<td>Top Margin</td>
<td>PRNTMARGT</td>
</tr>
<tr>
<td>Trace A B C</td>
<td>TRA, TRB, TRC</td>
</tr>
<tr>
<td>Trig</td>
<td>TM</td>
</tr>
<tr>
<td>Key Label</td>
<td>Remote Command</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>User Defined</td>
<td>PRNTMARGB, PRNTMARGT, PRNTRES</td>
</tr>
<tr>
<td>VBW/RBW Ratio</td>
<td>VB</td>
</tr>
<tr>
<td>Video</td>
<td>TM</td>
</tr>
<tr>
<td>Video Average On Off</td>
<td>VAVG</td>
</tr>
<tr>
<td>Video BW Auto Man</td>
<td>VB</td>
</tr>
<tr>
<td>View A, B, C</td>
<td>VIEW</td>
</tr>
<tr>
<td>X Axis Units Freq Time</td>
<td>LIMIFT</td>
</tr>
<tr>
<td>Zero Span</td>
<td>SP</td>
</tr>
</tbody>
</table>
**Functional Index**

This functional index categorizes the programming commands by the type of function that the command performs. The functional index contains the following information: the programming command mnemonic, the softkey or front-panel key that corresponds to the command’s function, and a brief definition of the command. Once the desired command is found, refer to the alphabetical listing of commands later in this chapter for more information about the command.

<table>
<thead>
<tr>
<th>Function Category</th>
<th>Command</th>
<th>Corresponding Key Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALIGNMENT</td>
<td>ALIGN</td>
<td>Alignments functions</td>
</tr>
<tr>
<td></td>
<td>RFCALIB</td>
<td>Align Now, RF</td>
</tr>
<tr>
<td>AMPLITUDE</td>
<td>AMPCOR</td>
<td>Ampcor functions</td>
</tr>
<tr>
<td></td>
<td>AT</td>
<td>Attenuation Auto Man</td>
</tr>
<tr>
<td></td>
<td>AUNITS</td>
<td>Amptd Units</td>
</tr>
<tr>
<td></td>
<td>LG</td>
<td>Scale Type Log Lin (Log)</td>
</tr>
<tr>
<td></td>
<td>LN</td>
<td>Scale Type Log Lin (Lin)</td>
</tr>
<tr>
<td></td>
<td>ML</td>
<td>Max Mixer Lvl</td>
</tr>
<tr>
<td></td>
<td>NRL</td>
<td>Norm Ref Lvl</td>
</tr>
<tr>
<td></td>
<td>PREAMPG</td>
<td>Ext Amp Gain</td>
</tr>
<tr>
<td></td>
<td>RL</td>
<td>Ref Level</td>
</tr>
<tr>
<td></td>
<td>ROFFSET</td>
<td>Ref Lvl Offst</td>
</tr>
<tr>
<td>BANDWIDTH</td>
<td>RB</td>
<td>Resolution BW Auto Man</td>
</tr>
<tr>
<td></td>
<td>VAVG</td>
<td>Video Average On Off</td>
</tr>
<tr>
<td></td>
<td>VB</td>
<td>Video BW Auto Man</td>
</tr>
<tr>
<td></td>
<td>VBR</td>
<td>VBW/RBW Ratio</td>
</tr>
<tr>
<td>CONFIGURATION</td>
<td>BAUDRATE</td>
<td>Baud Rate</td>
</tr>
<tr>
<td></td>
<td>DATEMODE</td>
<td>Datemode MDY DMY</td>
</tr>
<tr>
<td></td>
<td>DEFCONFIG</td>
<td>Default Config</td>
</tr>
<tr>
<td>Function Category</td>
<td>Command</td>
<td>Corresponding Key Function</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>COUPLING</td>
<td>AUTO</td>
<td>Auto Couple</td>
</tr>
<tr>
<td>DELETE, LOAD or SAVE</td>
<td>DELETE</td>
<td>Delete</td>
</tr>
<tr>
<td></td>
<td>LOAD</td>
<td>Load</td>
</tr>
<tr>
<td></td>
<td>PSTATE</td>
<td>Internal Lock On Off</td>
</tr>
<tr>
<td></td>
<td>SAVE</td>
<td>Save</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>ANNOT</td>
<td>Annotation On Off</td>
</tr>
<tr>
<td></td>
<td>DL</td>
<td>Display Line On Off</td>
</tr>
<tr>
<td></td>
<td>BRIGHT</td>
<td>Graticule On Off</td>
</tr>
<tr>
<td></td>
<td>HD</td>
<td>Esc</td>
</tr>
<tr>
<td></td>
<td>TH</td>
<td>Threshold On Off</td>
</tr>
<tr>
<td></td>
<td>TITLE</td>
<td>Change Title</td>
</tr>
<tr>
<td>FREQUENCY</td>
<td>CF</td>
<td>Center Freq</td>
</tr>
<tr>
<td></td>
<td>FA</td>
<td>Start Freq</td>
</tr>
<tr>
<td></td>
<td>FB</td>
<td>Stop Freq</td>
</tr>
<tr>
<td></td>
<td>FOFFSET</td>
<td>Freq Offset</td>
</tr>
<tr>
<td>Function Category</td>
<td>Command</td>
<td>Corresponding Key Function</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>INFORMATION</td>
<td>s s</td>
<td>CF StepAutoMan</td>
</tr>
<tr>
<td></td>
<td>CLS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CORREK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ERR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FREF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MDU</td>
<td></td>
</tr>
<tr>
<td></td>
<td>REV</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RQS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SRQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STB</td>
<td></td>
</tr>
<tr>
<td>INPUT and OUTPUT</td>
<td>OL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TDF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRAT</td>
<td>Trace A, Trace B, Trace C</td>
</tr>
<tr>
<td></td>
<td>TRC</td>
<td></td>
</tr>
<tr>
<td>LIMIT LINES</td>
<td>LIMIDEL</td>
<td>Delete Limits</td>
</tr>
<tr>
<td></td>
<td>LIMIDISP</td>
<td>Limit Display Y N Auto</td>
</tr>
<tr>
<td></td>
<td>LIMIFAIL</td>
<td>Limit Test On Off</td>
</tr>
<tr>
<td></td>
<td>LIMIFT</td>
<td>X Axis Units Freq Time</td>
</tr>
<tr>
<td></td>
<td>LIMIHALF</td>
<td>Select Line Upper Lower</td>
</tr>
<tr>
<td></td>
<td>LIMILINE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LIMIREL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LIMITTEST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SEGDEL</td>
<td></td>
</tr>
<tr>
<td>MARKER</td>
<td>MDS</td>
<td>Mkr Readout</td>
</tr>
<tr>
<td></td>
<td>MF</td>
<td></td>
</tr>
<tr>
<td>Function Category</td>
<td>Command</td>
<td>Corresponding Key Function</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td>----------------------------</td>
</tr>
<tr>
<td></td>
<td>MKA</td>
<td>Select Marker 1 2 3 4</td>
</tr>
<tr>
<td></td>
<td>MKACT</td>
<td>N dB Points On Off</td>
</tr>
<tr>
<td></td>
<td>MKBW</td>
<td>Marker A</td>
</tr>
<tr>
<td></td>
<td>MKCF</td>
<td>Marker Count On Off</td>
</tr>
<tr>
<td></td>
<td>MKD</td>
<td>Resolution Auto Man</td>
</tr>
<tr>
<td></td>
<td>MKF</td>
<td>Min Search</td>
</tr>
<tr>
<td></td>
<td>MKFC</td>
<td>Marker Normal</td>
</tr>
<tr>
<td></td>
<td>MKFCR</td>
<td>Marker Noise On Off</td>
</tr>
<tr>
<td></td>
<td>MKMIN</td>
<td>Marker All Off</td>
</tr>
<tr>
<td></td>
<td>MKN</td>
<td>Peak Search, Search, Next Peak, Next Pk Right, Next Pk Left</td>
</tr>
<tr>
<td></td>
<td>MKNOISE</td>
<td>Peak Excursn</td>
</tr>
<tr>
<td></td>
<td>MKOFF</td>
<td>Mkr Readout</td>
</tr>
<tr>
<td></td>
<td>MKP</td>
<td>Mkr → Ref Lvl</td>
</tr>
<tr>
<td></td>
<td>MKPAUSE</td>
<td>Mkr A → Span</td>
</tr>
<tr>
<td></td>
<td>MKPK</td>
<td>Mkr → CF Step</td>
</tr>
<tr>
<td></td>
<td>MKPX</td>
<td>Mkr → Start</td>
</tr>
<tr>
<td></td>
<td>MKREAD</td>
<td>Mkr → Stop</td>
</tr>
<tr>
<td></td>
<td>MKRL</td>
<td>Marker Trace Auto A B C</td>
</tr>
<tr>
<td></td>
<td>MKSP</td>
<td>Signal Track On Off</td>
</tr>
<tr>
<td></td>
<td>MKSS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MKFA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MKFB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MKTRACE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MKTRACK</td>
<td></td>
</tr>
<tr>
<td>Function Category</td>
<td>Command</td>
<td>Corresponding Key Function</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>MEASURE</td>
<td>DEMOD</td>
<td>Demod</td>
</tr>
<tr>
<td></td>
<td>NDB</td>
<td>\text{N dB Points} On Off</td>
</tr>
<tr>
<td></td>
<td>NDBPNT</td>
<td>\text{N dB Points} On Off</td>
</tr>
<tr>
<td></td>
<td>NDBPNTR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NRL</td>
<td>Norm Ref Lvl</td>
</tr>
<tr>
<td></td>
<td>PCTAM</td>
<td>% AM On Off</td>
</tr>
<tr>
<td></td>
<td>PCTAMR</td>
<td>% AM On Off</td>
</tr>
<tr>
<td></td>
<td>TOI</td>
<td>TOI On Off</td>
</tr>
<tr>
<td></td>
<td>TOIR</td>
<td></td>
</tr>
<tr>
<td>OPERATOR ENTRY</td>
<td>DN</td>
<td>\downarrow key</td>
</tr>
<tr>
<td></td>
<td>HD</td>
<td>Esc</td>
</tr>
<tr>
<td></td>
<td>UP</td>
<td>\uparrow key</td>
</tr>
<tr>
<td>PRESET</td>
<td>IP</td>
<td>Preset</td>
</tr>
<tr>
<td></td>
<td>POWERON</td>
<td>Power On IP Last</td>
</tr>
<tr>
<td></td>
<td>PRINT</td>
<td></td>
</tr>
<tr>
<td>PRINTING</td>
<td>PRINT</td>
<td>Printer Addr</td>
</tr>
<tr>
<td></td>
<td>PRNTADRS</td>
<td>Bottom Margin</td>
</tr>
<tr>
<td></td>
<td>PRNTMARGB</td>
<td>Top Margin</td>
</tr>
<tr>
<td></td>
<td>PRNTMARGT</td>
<td>Resolution</td>
</tr>
<tr>
<td></td>
<td>PRNTRES</td>
<td></td>
</tr>
<tr>
<td>SOURCE</td>
<td>SRCAT</td>
<td>Attenuation Auto Man</td>
</tr>
<tr>
<td></td>
<td>SRCPOFS</td>
<td>Amptd Offset</td>
</tr>
<tr>
<td></td>
<td>SRCPSTP</td>
<td>Amptd Step Auto Man</td>
</tr>
<tr>
<td></td>
<td>SRCPSWP</td>
<td>Power Sweep On Off</td>
</tr>
<tr>
<td></td>
<td>SRCPWR</td>
<td>Amplitude On Off</td>
</tr>
<tr>
<td></td>
<td>SWPCPL</td>
<td>Swp Coupling SR SA</td>
</tr>
<tr>
<td>Function Category</td>
<td>Command</td>
<td>Corresponding Key Function</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>SPAN</td>
<td>FS</td>
<td>Full Span</td>
</tr>
<tr>
<td></td>
<td>LSPAN</td>
<td>Last Span</td>
</tr>
<tr>
<td></td>
<td>SP</td>
<td>Span</td>
</tr>
<tr>
<td></td>
<td>SPZOOM</td>
<td>Span Zoom</td>
</tr>
<tr>
<td>SPEAKER</td>
<td>SPEAKER</td>
<td>Speaker On Off</td>
</tr>
<tr>
<td>SWEEP</td>
<td>CONTS</td>
<td>Sweep Cont Single (Cont)</td>
</tr>
<tr>
<td></td>
<td>s s</td>
<td>CF Step Auto Man</td>
</tr>
<tr>
<td></td>
<td>ST</td>
<td>Sweep Time Auto Man</td>
</tr>
<tr>
<td>SYNCHRONIZATION</td>
<td>DONE</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>Sweep Cont Single (Single)</td>
</tr>
<tr>
<td>TRACE</td>
<td>AXB</td>
<td>A ↔ B</td>
</tr>
<tr>
<td></td>
<td>BLANK</td>
<td>Blank A B C</td>
</tr>
<tr>
<td></td>
<td>BML</td>
<td>B → DL → B</td>
</tr>
<tr>
<td></td>
<td>BTC</td>
<td>B → C</td>
</tr>
<tr>
<td></td>
<td>BXC</td>
<td>B ↔ C</td>
</tr>
<tr>
<td></td>
<td>CLRW</td>
<td>Clear Write A, B, or C</td>
</tr>
<tr>
<td></td>
<td>DET</td>
<td>Detector</td>
</tr>
<tr>
<td></td>
<td>MINH</td>
<td>Min Hold C</td>
</tr>
<tr>
<td></td>
<td>MOV</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MXMH</td>
<td>Max Hold A or B</td>
</tr>
<tr>
<td></td>
<td>TRA TRB TRC</td>
<td>Trace A, Trace B, Trace C</td>
</tr>
<tr>
<td></td>
<td>TRSTAT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VAVG</td>
<td>Video Average On Off</td>
</tr>
<tr>
<td></td>
<td>VIEW</td>
<td>View A B C</td>
</tr>
<tr>
<td>TRACE MATH</td>
<td>APB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CLRAVG</td>
<td>Video Average On Off</td>
</tr>
<tr>
<td>Function Category</td>
<td>Command</td>
<td>Corresponding Key Function</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>TRIGGER</td>
<td>SNGLS</td>
<td>Sweep Cont Single (Single)</td>
</tr>
<tr>
<td></td>
<td>TM</td>
<td>Trig</td>
</tr>
<tr>
<td></td>
<td>TS</td>
<td>Sweep Cont Single (Single)</td>
</tr>
<tr>
<td></td>
<td>MEAN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MEANTH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PEAKS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RMS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SMOOTH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SQR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STDEV</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUMSQR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VARIANCE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XCH</td>
<td></td>
</tr>
</tbody>
</table>

*Video*  

A ↔ B, B ↔ C
Command Backwards Compatibility

The alternate commands provide compatibility with commands used by the HP 8566A/B, HP 8568A/B, and HP 70000 Series instruments. The equivalent commands for the HP ESA-L1500A spectrum analyzer are listed in the right column.

<table>
<thead>
<tr>
<th>Alternate Commands</th>
<th>Description</th>
<th>HP ESA-L1500A Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Clear write trace A</td>
<td>CLRW TRA</td>
</tr>
<tr>
<td>A2</td>
<td>Max hold trace A</td>
<td>MXMH TRA</td>
</tr>
<tr>
<td>A3</td>
<td>Store and view trace A</td>
<td>VIEW TRA</td>
</tr>
<tr>
<td>A4</td>
<td>Store and blank trace A</td>
<td>BLANK TRA</td>
</tr>
<tr>
<td>B1</td>
<td>Clear write trace B</td>
<td>CLRW TRB</td>
</tr>
<tr>
<td>B2</td>
<td>Max hold trace B</td>
<td>MXMH TRB</td>
</tr>
<tr>
<td>B3</td>
<td>Store and view trace B</td>
<td>VIEW TRB</td>
</tr>
<tr>
<td>B4</td>
<td>Store and blank trace B</td>
<td>BLANK TRB</td>
</tr>
<tr>
<td>BL</td>
<td>B = DL -&gt; B</td>
<td>BML</td>
</tr>
<tr>
<td>CA</td>
<td>Coupled input attenuation</td>
<td>AT AUTO</td>
</tr>
<tr>
<td>CR</td>
<td>Coupled resolution bandwidth</td>
<td>RB AUTO</td>
</tr>
<tr>
<td>CS</td>
<td>Coupled step size</td>
<td>SS AUTO</td>
</tr>
<tr>
<td>CT</td>
<td>Coupled sweep time</td>
<td>ST AUTO</td>
</tr>
<tr>
<td>CV</td>
<td>Coupled video bandwidth</td>
<td>VB AUTO</td>
</tr>
<tr>
<td>E1</td>
<td>Peak search</td>
<td>MKPK HI</td>
</tr>
<tr>
<td>E2</td>
<td>Enter marker into center frequency</td>
<td>MKCF</td>
</tr>
<tr>
<td>E3</td>
<td>Enter marker delta into center frequency stepsize</td>
<td>MKSS</td>
</tr>
<tr>
<td>E4</td>
<td>Enter marker amplitude into reference level</td>
<td>MKRL</td>
</tr>
<tr>
<td>EX</td>
<td>Exchange trace A and B</td>
<td>AXB</td>
</tr>
<tr>
<td>Alternate Commands</td>
<td>Description</td>
<td>HP ESA-L1500A Command</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>LO</td>
<td>Display line off</td>
<td>DL OFF</td>
</tr>
<tr>
<td>M1</td>
<td>Marker off</td>
<td>MKOFF</td>
</tr>
<tr>
<td>M2</td>
<td>Marker normal</td>
<td>MKN</td>
</tr>
<tr>
<td>M3</td>
<td>Marker delta</td>
<td>MKD</td>
</tr>
<tr>
<td>MA</td>
<td>Marker amplitude</td>
<td>MKA</td>
</tr>
<tr>
<td>MC</td>
<td>Marker count</td>
<td>MKFC</td>
</tr>
<tr>
<td>MT</td>
<td>Signal track on off</td>
<td>MKTRACK ON/OFF</td>
</tr>
<tr>
<td>01</td>
<td>Output format, in real number format</td>
<td>TDF P</td>
</tr>
<tr>
<td>02</td>
<td>Output format, in binary format, two bytes (word)</td>
<td>TDF B;MDS W</td>
</tr>
<tr>
<td>03</td>
<td>Output format, in measurement data format</td>
<td>TDFM</td>
</tr>
<tr>
<td>04</td>
<td>Output format, in binary format, 1 byte per element</td>
<td>TDF B;MDS B</td>
</tr>
<tr>
<td>R1</td>
<td>Activates illegal command service requestonly</td>
<td>RQS 32</td>
</tr>
<tr>
<td>R2</td>
<td>Activates end-of-sweep, illegal command</td>
<td>RQS 36</td>
</tr>
<tr>
<td>R3</td>
<td>Activates broken hardware, illegal command</td>
<td>RQS 40</td>
</tr>
<tr>
<td>R4</td>
<td>Activates units-key pressed, illegal command</td>
<td>RQS 34</td>
</tr>
<tr>
<td>RC</td>
<td>Recall state</td>
<td>RCLS</td>
</tr>
<tr>
<td>S1</td>
<td>Sweep continuous</td>
<td>CONTS</td>
</tr>
<tr>
<td>S2</td>
<td>Sweep single</td>
<td>SNGLS</td>
</tr>
<tr>
<td>TA</td>
<td>Transfers trace A to the controller in display units.</td>
<td>TDF M:TRA</td>
</tr>
<tr>
<td>Alternate Commands</td>
<td>Description</td>
<td>HP ESA-L1500A Command</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>TB</td>
<td>Transfers trace B to the controller in display units.</td>
<td>TDF M;TRB</td>
</tr>
<tr>
<td>TO</td>
<td>Threshold off</td>
<td>TH OFF</td>
</tr>
<tr>
<td>T1</td>
<td>Trigger mode free run</td>
<td>TM FREE</td>
</tr>
<tr>
<td>T2</td>
<td>Trigger mode line</td>
<td>TM LINE</td>
</tr>
<tr>
<td>T3</td>
<td>Trigger mode external</td>
<td>TM EXT</td>
</tr>
<tr>
<td>T4</td>
<td>Trigger mode video</td>
<td>TM VID</td>
</tr>
</tbody>
</table>
Programming Commands
Command Syntax Conventions

Command syntax is represented pictorially.

**Figure 14**  
Command Syntax Figure

- Ovals enclose command mnemonics. The command mnemonic must be entered exactly as shown.
- Circles and ovals surround secondary keywords or special numbers and characters. The characters in circles and ovals are considered reserved words and must be entered exactly as shown. See Table 8.
- Rectangles contain the description of a syntax element defined in Table 7.
- A loop above a syntax element indicates that the syntax element can be repeated.
- Solid lines represent the recommended path.
- Dotted lines indicate an optional path for bypassing secondary keywords or using alternate units.
- Arrows and curved intersections indicate command path direction.
- Semicolons are the recommended command terminators. Using semicolons makes programs easier to read, prevents command misinterpretation, and is recommended by IEEE Standard 728.

**NOTE:**  
Uppercase is recommended for entering all commands unless otherwise noted.

Syntax Elements are shown in the syntax diagrams as elements within rectangles. In the syntax diagrams, characters and secondary keywords are shown within circles or ovals. Characters and secondary keywords must be entered exactly as shown.
<table>
<thead>
<tr>
<th>Syntax Component</th>
<th>Definition/Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyzer command</td>
<td>Any spectrum analyzer command in this chapter, with required parameters and terminators.</td>
</tr>
<tr>
<td>Character</td>
<td>$a_b_c_d_e_f_g_h_i_j_k_l_m_o_n_p_q_r_s_t_u_v_w_x_y_z$ data byte.</td>
</tr>
<tr>
<td>Character &amp; EOI</td>
<td>8-bit byte containing only character data and followed by end-or-identify (EOI) condition, where the EOI control line on HP-IB is asserted to indicate the end of the transmission. END signifies the EOI condition.</td>
</tr>
<tr>
<td>Character string</td>
<td>A list of characters.</td>
</tr>
<tr>
<td>Data byte</td>
<td>S-bit byte containing numeric or character data.</td>
</tr>
<tr>
<td>Data byte &amp; EOI</td>
<td>8-bit byte containing numeric or character data followed by end-or-identify (EOI) condition, where the EOI control line on HP-IB is asserted to indicate the end of the transmission. END signifies the EOI condition.</td>
</tr>
<tr>
<td>Delimiter</td>
<td>$123456789$</td>
</tr>
<tr>
<td>Digit</td>
<td>Represents the least significant byte of a two-byte word that describes the number of bytes returned or transmitted. See msb length.</td>
</tr>
<tr>
<td>lsb length</td>
<td>Represents the most significant byte of a two-byte word that describes the number of bytes returned or transmitted. See msb length.</td>
</tr>
<tr>
<td>Number</td>
<td>Expressed as integer, decimal, or in exponential (E) form.</td>
</tr>
<tr>
<td></td>
<td>Real Number Range: $\pm 1.797693 \times 10^{308}$, including 0.</td>
</tr>
<tr>
<td></td>
<td>Up to 15 significant figures allowed.</td>
</tr>
<tr>
<td></td>
<td>Numbers may be as small as $\pm 2.225073858507202 \times 10^{-308}$</td>
</tr>
<tr>
<td></td>
<td>Integer Number Range: $-32,768$ through $+32,767$</td>
</tr>
<tr>
<td>Output termination</td>
<td>Carriage return ($C_R$) and line feed ($L_F$), with end-or-identify (EOI) condition. ASCII codes 13 (carriage return) and 10 (line feed) is sent via HP-IB, then the end-or-identify control line on HP-IB sets to indicate the end of the transmission.</td>
</tr>
<tr>
<td>Trace element</td>
<td>Value contained in one trace point. Notated as TRA[N] where N specifies the point position in the trace array. Values for N are 1 to 401 (for traces A, B, C). The same values apply to trace B(TRB[N]), and trace C(TRC[N]).</td>
</tr>
</tbody>
</table>
### Table 7  Syntax Elements

<table>
<thead>
<tr>
<th>Syntax Component</th>
<th>Definition/Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace range</td>
<td>Values contained in trace segment. Multi-point segments are notated as TRA[N,M], where N and M are end points of a segment and specify point positions in trace array. Values for N or M are 1 to 401 (for traces A, B, C). The same values apply to trace B(TRB[N,M]) and trace C (TRC[N,M]). Single-point segments are notated the same as the trace element above.</td>
</tr>
<tr>
<td>Units</td>
<td>Represent standard scientific units.</td>
</tr>
<tr>
<td></td>
<td>Frequency Units: GHZ or GZ, MHZ or MZ, KHZ or KZ, HZ</td>
</tr>
<tr>
<td></td>
<td>Amplitude Units: DB, DM, DBMV, DBUV, V, MV, UV, W, MW, UW</td>
</tr>
<tr>
<td></td>
<td>Time Units: SC, MS, US</td>
</tr>
<tr>
<td></td>
<td>Current Units: A, MA, UA</td>
</tr>
<tr>
<td></td>
<td>Impedance Units: OHM</td>
</tr>
</tbody>
</table>

### Table 8  Characters and Secondary Keywords (Reserved Words)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Amp (unit) or A-block data field</td>
</tr>
<tr>
<td>ABHZ</td>
<td>Absolute Hz (unit)</td>
</tr>
<tr>
<td>AC</td>
<td>Alternating current</td>
</tr>
<tr>
<td>ALL</td>
<td>All</td>
</tr>
<tr>
<td>AM</td>
<td>Amplitude modulation</td>
</tr>
<tr>
<td>AMP</td>
<td>Amplitude</td>
</tr>
<tr>
<td>AMPCOR</td>
<td>Amplitude correction</td>
</tr>
<tr>
<td>AUTO</td>
<td>Auto couple or set to automatic</td>
</tr>
<tr>
<td>AVG</td>
<td>Average</td>
</tr>
<tr>
<td>B</td>
<td>8-bit byte or binary format</td>
</tr>
<tr>
<td>BW</td>
<td>Black and white</td>
</tr>
<tr>
<td>CNT</td>
<td>Counter-lock</td>
</tr>
<tr>
<td>COLOR</td>
<td>Color</td>
</tr>
</tbody>
</table>
### Table 8

**Characters and Secondary Keywords (Reserved Words)**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB</td>
<td>Decibel (unit)</td>
</tr>
<tr>
<td>DBM</td>
<td>Absolute decibel milliwatt (unit)</td>
</tr>
<tr>
<td>DBMV</td>
<td>Decibel millivolt (unit)</td>
</tr>
<tr>
<td>DBUV</td>
<td>Decibel microvolt (unit)</td>
</tr>
<tr>
<td>DC</td>
<td>Direct current</td>
</tr>
<tr>
<td>DELTA</td>
<td>Delta</td>
</tr>
<tr>
<td>DISP</td>
<td>Display</td>
</tr>
<tr>
<td>DM</td>
<td>Absolute decibel milliwatt (unit)</td>
</tr>
<tr>
<td>DMY</td>
<td>Day, month, year format</td>
</tr>
<tr>
<td>DN</td>
<td>Decreases parameter one step size</td>
</tr>
<tr>
<td>DUMP</td>
<td>Dump</td>
</tr>
<tr>
<td>EXT</td>
<td>External trigger</td>
</tr>
<tr>
<td>FIXED</td>
<td>Fixed</td>
</tr>
<tr>
<td>FLAT</td>
<td>Flat</td>
</tr>
<tr>
<td>FMD</td>
<td>Frequency modulation demodulator</td>
</tr>
<tr>
<td>FM</td>
<td>Frequency modulation</td>
</tr>
<tr>
<td>FMV</td>
<td>Frequency modulation detection</td>
</tr>
<tr>
<td>FREE</td>
<td>Free run</td>
</tr>
<tr>
<td>FREQ or FRQ</td>
<td>Frequency</td>
</tr>
<tr>
<td>GHZ</td>
<td>Gigahertz (unit)</td>
</tr>
<tr>
<td>GZ</td>
<td>Gigahertz (unit)</td>
</tr>
<tr>
<td>HI</td>
<td>Highest</td>
</tr>
<tr>
<td>HPIB</td>
<td>HP-IB</td>
</tr>
<tr>
<td>HZ</td>
<td>Hertz (unit)</td>
</tr>
</tbody>
</table>
### Table 8

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I-block data field</td>
</tr>
<tr>
<td>INT</td>
<td>Internal or integer</td>
</tr>
<tr>
<td>P</td>
<td>Instrument preset</td>
</tr>
<tr>
<td>IST</td>
<td>Inverse sweep time</td>
</tr>
<tr>
<td>KHZ</td>
<td>Kilohertz (unit)</td>
</tr>
<tr>
<td>KZ</td>
<td>Kilohertz (unit)</td>
</tr>
<tr>
<td>LAST</td>
<td>Last state</td>
</tr>
<tr>
<td>LIMILINE</td>
<td>Limit line</td>
</tr>
<tr>
<td>LINE</td>
<td>Line trigger</td>
</tr>
<tr>
<td>LOAD</td>
<td>Load operation</td>
</tr>
<tr>
<td>LOWER</td>
<td>Lower limit line</td>
</tr>
<tr>
<td>M</td>
<td>Measurement units</td>
</tr>
<tr>
<td>MA</td>
<td>Milliamp (unit)</td>
</tr>
<tr>
<td>MDY</td>
<td>Month, day, year format</td>
</tr>
<tr>
<td>MHZ</td>
<td>Megahertz (unit)</td>
</tr>
<tr>
<td>MS</td>
<td>Millisecond (unit)</td>
</tr>
<tr>
<td>MTR</td>
<td>Meter</td>
</tr>
<tr>
<td>MV</td>
<td>Millivolts (unit)</td>
</tr>
<tr>
<td>MW</td>
<td>Milliwatt (unit)</td>
</tr>
<tr>
<td>MZ</td>
<td>Megahertz (unit)</td>
</tr>
<tr>
<td>NEG</td>
<td>Negative</td>
</tr>
<tr>
<td>NH</td>
<td>Next highest peak</td>
</tr>
<tr>
<td>NL</td>
<td>Next peak left</td>
</tr>
<tr>
<td>NONE</td>
<td>No units</td>
</tr>
<tr>
<td>NR</td>
<td>Next peak right</td>
</tr>
</tbody>
</table>
### Characters and Secondary Keywords (Reserved Words)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRM or NORMAL</td>
<td>Normal</td>
</tr>
<tr>
<td>OA</td>
<td>Output amplitude</td>
</tr>
<tr>
<td>OFF</td>
<td>Turns off function</td>
</tr>
<tr>
<td>ON</td>
<td>Turns on function</td>
</tr>
<tr>
<td>P</td>
<td>Parameter units</td>
</tr>
<tr>
<td>PER</td>
<td>Period</td>
</tr>
<tr>
<td>PKAVG</td>
<td>Peak average</td>
</tr>
<tr>
<td>PKPIT</td>
<td>Peak pit</td>
</tr>
<tr>
<td>POINT</td>
<td>Point</td>
</tr>
<tr>
<td>POS</td>
<td>Positive</td>
</tr>
<tr>
<td>PSN</td>
<td>Position</td>
</tr>
<tr>
<td>RS232</td>
<td>RS-232 interface</td>
</tr>
<tr>
<td>SA</td>
<td>Signal analysis</td>
</tr>
<tr>
<td>SC</td>
<td>Seconds (unit)</td>
</tr>
<tr>
<td>SLOPE</td>
<td>Slope</td>
</tr>
<tr>
<td>SMP</td>
<td>Sample detection mode</td>
</tr>
<tr>
<td>SP</td>
<td>Space</td>
</tr>
<tr>
<td>SR</td>
<td>Stimulus response</td>
</tr>
<tr>
<td>STATE</td>
<td>State register</td>
</tr>
<tr>
<td>STEP</td>
<td>Step key ability</td>
</tr>
<tr>
<td>STORE</td>
<td>Store</td>
</tr>
<tr>
<td>SWT</td>
<td>Sweep time</td>
</tr>
<tr>
<td>TG</td>
<td>Tracking generator</td>
</tr>
<tr>
<td>TRA</td>
<td>Trace A</td>
</tr>
</tbody>
</table>
### Table 8

**Characters and Secondary Keywords (Reserved Words)**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRB</td>
<td>Trace B</td>
</tr>
<tr>
<td>TRC</td>
<td>Trace C</td>
</tr>
<tr>
<td>UA</td>
<td>Microamp (unit)</td>
</tr>
<tr>
<td>UP</td>
<td>Increases the parameter one step size</td>
</tr>
<tr>
<td>UPLOW</td>
<td>Upper and lower limit lines</td>
</tr>
<tr>
<td>UPPER</td>
<td>Upper limit line</td>
</tr>
<tr>
<td>US</td>
<td>Microseconds (unit)</td>
</tr>
<tr>
<td>UV</td>
<td>Microvolts (unit)</td>
</tr>
<tr>
<td>UW</td>
<td>Microwatt (unit)</td>
</tr>
<tr>
<td>V</td>
<td>Volts (unit)</td>
</tr>
<tr>
<td>VID</td>
<td>Video trigger</td>
</tr>
<tr>
<td>W</td>
<td>Watts or word (for MDS command)</td>
</tr>
<tr>
<td>*</td>
<td>Asterisk (used as a wildcard)</td>
</tr>
<tr>
<td>;</td>
<td>Semicolon (ASCII code 59)</td>
</tr>
<tr>
<td>,</td>
<td>Comma (ASCII code 44)</td>
</tr>
<tr>
<td>0</td>
<td>Off (command argument)</td>
</tr>
<tr>
<td>1</td>
<td>On (command argument)</td>
</tr>
<tr>
<td>50</td>
<td>50 Ω</td>
</tr>
<tr>
<td>75</td>
<td>75 Ω</td>
</tr>
<tr>
<td>?</td>
<td>Returns a query response containing the value or state of the associated parameter. The query response is followed by a carriage-return/line-feed.</td>
</tr>
</tbody>
</table>
Programming Command Descriptions

To find a programming command that performs a particular function refer to the chapter of cross-reference information where commands are categorized by function. Once the desired command is found in the functional index, refer to the description for the command in this chapter.
ALIGN Self-alignment Routines

Initiates self-alignment routines.

Syntax

Preset State: ALIGN AUTOON
Equivalent Key: The softkeys accessed by Auto Align and Align Now
Related Command: CORREK
Example

10 OUTPUT 718; "ALIGN ALL;"

Performs an immediate alignment of all sub-assemblies.

Description

The ALIGN command controls alignment functions. ALIGN initiates action according to the ALIGN parameters. The various parameters correspond to spectrum analyzer softkeys as follows:

- **AUTOON** enables the automatic alignment of all assemblies within the instrument. AUTOON corresponds to **ALL** under the **Auto Align** key.

- **AUTOFF** disables the automatic alignment of all assemblies within the instrument. AUTOFF corresponds to Off under the **Auto Align** key.

**NOTE:** When auto alignment is disabled, instrument calibration may no longer be valid. Refer to the specifications chapter in the User’s and Calibration Guide, for conditions under which the spectrum analyzer will meet its specifications when the auto alignment routine is disabled.

- **AUTONRF** enables the automatic alignment of all assemblies except the RF assembly. AUTONRF corresponds to **All but RF** under the **Auto Align** key.

- Eliminating the automatic alignment of the RF prevents changes in the input impedance between sweeps, which could cause input device instability.

- **ALL** performs an alignment of all assemblies within the instrument. ALL corresponds to **All** under the **Align Now** key.

- **ADC** performs an alignment of the instrument ADC circuitry. ADC corresponds to **ADC** under the **Align Now** key.

- **IF** performs an alignment of the instrument IF assembly. IF corresponds to under the **Align Now** key.

- **LO** performs an alignment of the instrument LO assembly. LO corresponds to **LO** under the **Align Now** key.

- **RF** performs an alignment of the instrument RF assembly. RF corresponds to **RF** under the **Align Now** key.

**NOTE:** ALIGN ALL and ALIGN RF use the internal 50 MHz oscillator. When this is done, the spectrum analyzer input impedance becomes an “open” between sweeps. This impedance mismatch could cause input device instability.

- **DEFAULTS** initializes the alignment data. DEFAULTS corresponds to **Load Defaults** under the **Alignments** key. The ALIGN DEFAULTS command must be followed by
an ALIGN ALL command.

FRQCOROFF disables the application of frequency corrections generated by the instrument alignment. FRQCOROFF corresponds to Freq Correct On Off under the Alignments key.

**NOTE:**
When auto alignment is disabled, instrument calibration may no longer be valid. Refer to the specifications chapter in the User’s and Calibration Guide, for conditions under which the spectrum analyzer will meet its specifications when the auto alignment routine is disabled.

FRQCORON enables the application of frequency corrections generated by the instrument alignment. FRQCORON corresponds to Freq Correct On Off under the Alignments key.

**NOTE:**
The frequency correction application is enabled whenever the ac power is cycled, or the instrument is preset.

**Query Response**

```
output termination

AUTOON
AUTOOFF
AUTONRF
```

QALIGN
**AMPCOR Amplitude Correction**

Applies amplitude corrections at specified frequencies.

### Syntax

```
AMPCOR 7

@ frequency, amplitude

kHZ OFF

Item Description/Default

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is dB.</td>
<td>Frequency: 0 to 1000 GHz Amplitude: ±327 dB.</td>
</tr>
</tbody>
</table>

Equivalent Keys: the softkeys accessed by Ampcor
Preset State: AMPCOR OFF
Related Commands: AT, RL
```
Example
Compensate for frequency dependent amplitude inaccuracies at the input.

10  DIM A$[200]
20  OUTPUT 718;“CF 1GHZ;SP 200MHZ;”
Sets center frequency and span.
30  OUTPUT 718;“AMPCOR 100MHZ,5DB,1GHZ,-5DB,1.5GHZ,10DB;”
Stores frequency-amplitude pairs in spectrum analyzer. Notice that frequencies are in ascending order.
40  OUTPUT 718;“AMPCOR?;”
Returns correction values to computer.
50  ENTER 718;A$
60  PRINT A$
Displays the frequency-amplitude pairs.
70  OUTPUT 718;“AMPCOR OFF;”
Turns off the amplitude correction constants.
80  OUTPUT 718;“AMPCOR?;”
90  ENTER 718;A$
100 PRINT A$
Because AMPCOR is off, “0,0” is displayed.
110 END

Description
Use AMPCOR to compensate for frequency dependent amplitude variations at the spectrum analyzer input. Up to 80 pairs of amplitude correction points can be entered. The frequency values entered must either be equal or in increasing order, or an error condition results. Whenever AMPCOR is on, the correction values are applied to all measurement results. Executing “AMPCOR ON;” or entering frequency and amplitude corrections, turns on the amplitude correction factors. Performing an instrument preset (IP) or turning off the spectrum analyzer sets AMPCOR to OFF. (Setting AMPCOR to OFF does not change the frequency amplitude correction factors that have been entered.)
The values of the correction points are applied across the active measurement range. Between points, the correction values are interpolated. When measuring at frequencies outside the first and last correction points, these values are used as the correction value. If you do not want any amplitude correction outside of the first and last correction points, set the amplitude correction to 0 at the frequencies that are outside of the first and last correction values.

Amplitude correction factors can be stored in spectrum analyzer memory with the SAVE command. The amplitude correction factors can be edited and viewed with the ampcor softkey functions.

**Query Response**

AMPCOR? returns the frequency and amplitude correction pairs.

```
frequency amp I i titude
```

Returned values are 0,0 when AMPCOR is set to OFF.
**ANNOT Annotation**

Turns on or off the display annotation.

**Syntax**

```
ANNOT 2-i-e OFF / ON
```

Equivalent Key: **Annotation On Off**

Preset State: **ANNOT ON**

Related Commands: **GRAT, TITLE**

**Example**

10 OUTPUT 718;"ANNOT ON;"

Turns on the annotation.

20 OUTPUT 718;"ANNOT?;"

Queries state of the annotation function.

30 ENTER 718;Reply$

Places response in a variable.

40 DISP Reply$

Displays response on the computer screen.

50 END
Description

The `ANNOT` command turns on or off all the words and numbers (annotation) on the spectrum analyzer display (except for the `softkey` labels).

Query Response
APB Trace A Plus Trace B

Adds trace A to trace B and sends the result to trace A.

**Syntax**

```
( APB
```

Related Commands: CLRW, SNGLS, TS, VIEW.

**Example**

10 OUTPUT 718;“IP;SNGLS;”
Initializes spectrum analyzer, changes to single-sweep mode.

20 OUTPUT 718;“TS;”
Updates the trace.

30 OUTPUT 718;“VIEW TRA;RL -20DM;CLRW TRB;”
Changes the reference level.

40 OUTPUT 718;“TS;VIEW TRB;”
Takes a measurement sweep.

50 OUTPUT 718;“APB;”
Activates APB command.

60 OUTPUT 718;“BLANK TRB;VIEW TRA;”
Displays the result of APB.

70 END

**Description**

The traces are added as 16-bit integers. Negative numbers are represented in two’s complement format. The two’s complement representation of a negative number is obtained by changing the 1s to 0s in the binary representation of the number, and then adding 1.
AT Attenuation

Specifies the RF input attenuation.

Syntax

```
AT attenuation.
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer. Default units are dB.</td>
<td>Input attenuator range of spectrum analyzer</td>
</tr>
</tbody>
</table>

Equivalent Key: **Attenuation** 
Auto Man is similar
Preset State: 10 dB
Step Increment: in 5 dB steps
Related Commands: AUTO, ML, RL

Example

```
10 OUTPUT 718;"AT 40DB;"
Sets the attenuation to 40 dB.
```

```
20 OUTPUT 718;"AT UP;"
Increases the attenuation to 45 dB.
```
Description

The AT command specifies the input attenuation in 5 dB steps. Normally, the input attenuator is coupled to the reference level. When a continuous wave signal is displayed with its peak at or below the reference level, the coupling keeps the mixer input level at or below the specified level (also see the command “ML”). The AT command allows less than the specified value at the mixer input.

When the attenuation is increased with the AT command, the reference level does not change. If the attenuation is decreased from the coupled value using the AT command, the reference level will be decreased. When the reference level is changed using the RL command, the input attenuation changes to maintain a constant signal level on the screen if attenuation is auto-coupled. Using auto-coupling resets the attenuation value so that a continuous wave signal displayed at the reference level yields -10 dBm (or the specified mixer level) at the mixer input.

CAUTION: Signal levels above +30 dBm will damage the spectrum analyzer.

Query Response
ATC Transfer Trace A to Trace C

Transfers trace A into trace C.

Syntax

\[ \text{ATC} \]

Equivalent Key: \( A \rightarrow C \).

Related Commands: BLANK, CLRW, SNGLS, TS, VIEW.

Example

10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"SNGLS;"
Activates single-sweep mode.

40 OUTPUT 718;"CF 300 MHZ;SP 1MHZ;"
Sets up measurement range.

50 OUTPUT 718;"CLRW TRA;TS;"
Takes measurement sweep.

60 OUTPUT 718;"ATC;"
Moves trace A to trace C.

70 OUTPUT 718;"BLANK TRA;VIEW TRC;"
Displays result in trace C.

80 END

Description

The ATC command moves trace A into trace C, then stops updating trace C by placing it in the view mode. Trace A is unchanged by ATC. Trace A must contain a complete sweep of measurement information.
AUNITSAmplitude Units

Specifies the amplitude units for input, output, and display.

Syntax

```
AUNITS
```

Equivalent Key: Amptd Units
Related Commands: DL, MKA, RL, TH

Example

```
10 OUTPUT 718; "LN;"
Changes spectrum analyzer to linear mode.

20 OUTPUT 718; "AUNITS DBMV;"
Changes the linear amplitude units to DBMV.

30 OUTPUT 718; "AUNITS?;"
Queries current amplitude units.

40 ENTER 718; Reply$
Puts response in a variable.

50 DISP Reply$
Displays response on the computer screen.
```

Description

The AUNITS command sets the amplitude readouts (reference level, marker, display line, and threshold) to the specified units. Different amplitude units can be set for log and linear amplitude scales.
When watts are selected as the units of measure, values below $1 \text{ pW}$ are rounded to $0 \text{ pw}$.

**Query Response**

The query response returns the current amplitude units for the current amplitude scale.
**AUTO Auto Couple**

Couples functions which have an AUTO parameter and sets the sweep coupling mode.

**Syntax**

```
active function

KEYWORD AUTO

AUTO ALL
```

Equivalent Key: **Auto Couple**

Related Commands: AT, LIMIDISP, MKFCR, RB, SRCAT, SRPCSTP, SWPCPL, SS, ST, VB, VBR

**Example #1**

```
10 OUTPUT 718; "AT AUTO;"
```

Couples the attenuation.

**Example #2**

```
10 OUTPUT 718; "AUTO ALL;"
```

Couples all functions.

**Description**

The result of the AUTO command depends on the active function it acts upon. The following are the functions that are affected by the AUTO parameter:

- **AT**
  - couples attenuation to the reference level
- **MKFCR**
  - deactivates use of user-supplied counter resolution value without changing the value of resolution
allows LIMITEST to control the display of limit lines

couples resolution bandwidth to frequency span

couples the source attenuator to the source output level

couples the source power level step size to the reference level

couples step size to frequency span

couples sweep time to frequency span

selects spectrum analysis auto-coupled sweeptimes

couples video bandwidth to resolution bandwidth

sets the video to bandwidth ratio to 0.3

Individual functions can be coupled by entering the keyword for the command before AUTO, (for example, “AT AUTO;”).

Executing “AUTO ALL;” couples all functions.
**AXB Exchange Trace A and Trace B**

Exchanges trace A and trace B.

**Syntax**

```
AXB
```

Equivalent Key: A ↔ B

Related Commands: CLRW, SNGLS, TS, VIEW

**Example**

```plaintext
10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"SNGLS;"
Activates single-sweep mode.

30 OUTPUT 718;"DET POS;TS;"
Activates positive-peak detection of trace A.

40 OUTPUT 718;"VIEW TRA;"
Stores results, displays trace A.

50 OUTPUT 718;"DET SMP;"
Activates sample detection.

60 OUTPUT 718;"CLRW TRB;TS;"
Clear-writes trace B and takes sweep.

70 OUTPUT 718;"VIEW TRB;"
Stores results of sweep in trace B.

80 OUTPUT 718;"AXB;"
Exchanges trace A with trace B.

90 OUTPUT 718;"BLANK TRB;"
Blanks trace B, leaving only trace A on screen.

100 END
```
Description

The AXB command exchanges trace A and trace B, point by point. AXB sets trace A and trace B to the view mode.
BAUDRATE Baud Rate of Spectrum Analyzer

Specifies the baud rate of a spectrum analyzer with Option 1AX (the RS-232 and parallel interface) installed in it.

Syntax

```
BAUDRATE
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any valid integer number</td>
<td>110 to 115200</td>
</tr>
</tbody>
</table>

Equivalent Key: Baud Rate
Option Required: Option 1AX

Example

The following example allows you to use a softkey to change the baud rate of the spectrum analyzer to 2400 baud.

```
10 OUTPUT 718; "BAUDRATE 2400;"
```

Description

The BAUDRATE command changes the baud rate of the spectrum analyzer to one of the standard baud rates. The standard baud rates are as follows: 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200. If you specify a baud rate other than one of the standard baud rates, the nearest standard baud rate will be used.

To communicate with the computer, the baud rates of the spectrum analyzer and the computer must be the same. Because changing the baud rate of the spectrum analyzer within a program ends communication with the computer, you should
only use **BAUDRATE** within a remote program or when using the external keyboard to enter programming commands. To reestablish communication with the computer, you must set the baud rate back to the baud rate of the computer.

**Query Response**

```
  baud rate
    number
    output termination
```
BLANK Blank Trace

Blanks trace A, trace B, or trace C and stops taking new data into the specified trace.

Syntax

Equivalent Keys: Blank A, Blank B, and Blank C
Preset State: BLANK TRB, BLANK TRC
Related Commands: CLRW, MXMH, TRDSP, VIEW

Example

10 OUTPUT 718; "BLANK TRA;"
BML Trace B Minus Display Line

Subtracts display line from trace B and places the result in trace B.

Syntax

```
BML
```

Equivalent Key: B →DL → B
Related Commands: BLANK, CLR, SUB, TS

Example

```
10 OUTPUT 718;"IP;SNGL;S;"
Initializes spectrum analyzer, activates single-sweep mode.

20 OUTPUT 718;"BLANK TRA;"
Blanks trace A.

30 OUTPUT 718;"CLR TRB;TS;"
Clear-writes trace B, takes sweep.

40 OUTPUT 718;"DL -70DM;"
Sets the display line to -70 dBm.

50 OUTPUT 718;"BML;"
Activates BML function.

60 END
```

Description

The BML command subtracts the display line from trace B (point by point), and sends the difference to trace B.
BRIGHT Display Brightness

Specifies the brightness of the display backlight.

Syntax

```
BRIGHT c number.
```

Table: Item Description/Default/Range

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any integer number</td>
<td>Values between 0 and 255</td>
</tr>
</tbody>
</table>

Equivalent Key: Hardkeys located to the upper left of the display
Related Commands: CONTRAST

Example

```
10 OUTPUT 718;"BRIGHT 88;"
```

Sets the brightness to a value of 88.

Description

The BRIGHT command specifies the amount of display backlight brightness using whole number values from 0 to 255. The two hardkeys at the upper left portion of the display vary the brightness.
BTC Transfer Trace B to Trace C

Transfers trace B into trace C.

Syntax

```
BTC XBTC
```

Equivalent Key: B -> C

Related Commands: BLANK, CLRW, SNGLS, TS, VIEW

Example

```
10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"SNGLS;"
Activates single-sweep mode.

30 OUTPUT 718;"BLANK TRA;"
Blanks trace A.

40 OUTPUT 718;"CF 300 MHZ;SP 1MHZ;"
Sets up measurement range.

50 OUTPUT 718;"CLRW TRB;TS;"
Takes measurement sweep.

60 OUTPUT 718;"BTC;"
Moves trace B to trace C.

70 OUTPUT 718;"BLANK TRB;VIEW TRC;"
Displays result in trace C.

END
```

Description

The BTC command moves trace B into trace C, then stops updating trace C by placing it in the view mode. Trace B is unchanged by BTC. Trace B must contain a complete sweep of measurement information.
BXC Trace B Exchange Trace C

Exchanges trace B and trace C.

**Syntax**

```
BXC
```

Equivalent Key: \( B \leftrightarrow C \)
Related Commands: BLANK, CLRW, SNGLS, TS

**Example**

10 OUTPUT 718;"IP;BLANK TRA"
Initializes spectrum analyzer.

20 OUTPUT 718;"SNGLS;"
Activates single-sweep mode.

30 OUTPUT 718;"DET POS;CLRW TRB;TS;"
Activates positive-peak detection of trace B.

40 OUTPUT 718;"VIEW TRB;"
Stores results and displays trace B.

50 OUTPUT 718;"DET SMP;CLRW TRC;"
Activates sample detection.

60 OUTPUT 718;"TS;"

70 OUTPUT 718;"VIEW TRC;"
Stores results of sweep in trace C.

80 OUTPUT 718;"BXC;"
Exchanges trace B with trace C.

90 OUTPUT 718;"BLANK TRB;"
Blanks trace B leaving only trace C on screen.

100 END
Description

The BXC command exchanges the contents of traces B and C, then places both traces in the view mode.

To retain all data, trace B and trace C should contain a complete sweep of measurement data before BXC is executed.
CAT Catalog

Catalogs spectrum analyzer memory and returns the catalog information to the controller.

Syntax

Equivalent Keys: the functions accessed by *Save* or *Load*
Related Commands: LOAD, SAVE

Example

This example returns the catalog information for the states stored in the memory. Catalog information is sent as individual catalog lines that are separated by a carriage return and a line feed. A carriage return, a line feed, and a line feed with an EOI (equivalent to a carriage return, a line feed, and a line feed) is asserted after the last item.

10 DIM User$(2000), Catalog$ (1:100)(80)

Dimensions strings to store the catalog information. *User$* stores the entire string of catalog information. *Catalog*$ stores the catalog information line by line (up to 80 lines and 100 characters long).

20 INTEGER I, Pos_lf

I and *Pos_lf* are used to search through *User$* string.

30 OUTPUT 718; "CAT SREG*;"

The spectrum analyzer sends catalog information for all the states stored in
memory.

40 ENTER 718 USING ",",-K;User$

Reads the catalog information into the User$ string.

50 I=0

60 WHILE LEN(User$)>1

Loops until the User$ string is empty.

70 I=I+1

80 Pos lf=POS(User$,CHR$(10))

Checks for line feeds. CHR$(10) represents the line feed, the ASCII code for a line feed is “10.”

90 Catalog$(I)=User$[I,Pos lf-2]

Extracts catalog line.

100 OUTPUT CRT;Catalog$(I)

Displays catalog line.

120 User$=User$[Pos lf+1]

130 END WHILE

140 END

**Description**

To use the CAT command, you must specify the type of information to be cataloged. After the spectrum analyzer has sent the catalog information to the controller, the spectrum analyzer sends two line feed characters to the controller. The register types are:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Register Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SREG</td>
<td>State</td>
</tr>
<tr>
<td>TTREG</td>
<td>Trace Data</td>
</tr>
<tr>
<td>ATREG</td>
<td>Ampcpor Data</td>
</tr>
<tr>
<td>LTREG</td>
<td>Limit Line Data</td>
</tr>
</tbody>
</table>
The zero in the command syntax indicates that the catalog information comes from the spectrum analyzer. If the zero is not specified when the command is sent, the analyzer will assume it.
**CF Center Frequency**

Specifies the center frequency.

**Syntax**

```
frequency value
```

- **Syntax**: `CF number HZ`
- **Equivalent Key**: `Center Freq`
- **Step Increment**: If uncoupled, step size is determined by the SS command. If coupled, step size is 10% of span.
- **Related Commands**: FA, FB, FOFFSET, FS, MKCF, MKSS, SP, SS

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is Hz.</td>
<td>Frequency range of the spectrum analyzer</td>
</tr>
</tbody>
</table>

Equivalent Key: **Center Freq**

Step Increment: If uncoupled, step size is determined by the SS command. If coupled, step size is 10% of span.

Related Commands: FA, FB, FOFFSET, FS, MKCF, MKSS, SP, SS
Programming Commands
Programming Command Descriptions

NOTE:

Although the spectrum analyzer allows entry of frequencies not in the specified frequency range, using frequencies outside the frequency span of the spectrum analyzer is not recommended and is not warranted to meet specifications.

Example

10 OUTPUT 718;"CF 300MHZ;"

Sets the center frequency to 300 MHz.

Description

The CF command specifies the value of the center frequency.

Query Response

Although the spectrum analyzer allows entry of frequencies not in the specified frequency range, using frequencies outside the frequency span of the spectrum analyzer is not recommended and is not warranted to meet specifications.
CLRAVG Clear Average

Restarts video averaging.

**Syntax**

```
CLRAVG
```

Equivalent Key: **Video Average**
Related Commands: CLRW, MINH, MXMH, VAVG

**Example**

```
10 OUTPUT 718;"IP;"
```
Initializes the spectrum analyzer.

```
20 OUTPUT 718;"VAVG 100;"
```
Initializes video averaging.

```
30 WAIT 30
```

```
40 OUTPUT 718;"CLRAVG;"
```
Restarts video averaging.

**Description**

The CLRAVG command restarts the VAVG command by resetting the number of averaged sweeps to one. The video averaging routine resets the number of sweeps, but does not stop video averaging. Use "VAVG OFF;" to stop video averaging.
CLRW Clear Write

Clears the specified trace and enables trace data acquisition.

Syntax

```
CLRW TRA
```

Equivalent Keys: **Clear Write A, Clear Write B, and Clear Write C**

Preset State: CLRW TRA

Related Commands: BLANK, DET, MINH, MXMH, VAVG, VIEW

Example

```
10 OUTPUT 718; "CLRW TRA;"
```

Description

The CLRW command places the indicated trace in the clear-write mode. Data acquisition begins at the next sweep. (See “TS” for more information about data acquisition.)
CLS Clear Status Byte

Clears all status bits.

Syntax

```
CLS
```

Related Commands: RQS, SRQ, STB

Example

```
10 OUTPUT 718,"CLS;"
```

Description

The CLS command clears all the status bits in the status byte. (See “SRQ” for more information on the status byte.)
CMDERRQ Command Error Query

The CMDERRQ command returns the current buffer of illegal commands and then clears the illegal-command buffer from the spectrum analyzer.

Syntax

Example

10 OUTPUT 718;“XYZ;”
20 OUTPUT 718;“CMDERRQ;”
Initiates the command
30 ENTER 718;A$
Displays the response
XYZ

Description

Executing the CMDERRQ command does the following:

- Returns the most recently stored illegal or unrecognized command characters.
- Returns up to 45 characters of an illegal command or commands.
- Erases the illegal command buffer.

Query Response
CONFIG Configuration

Returns the analyzer configuration information to the controller.

Syntax

```
CONFIG ? ;
```

Related Command: SHOWSYS

Example

This example returns the configuration information for the analyzer. Configuration information is sent as individual lines that are separated by a carriage return and a line feed. A carriage return, a line feed, and a line feed with an EOI (equivalent to a carriage return, a line feed, and a line feed) is asserted after the last item.

```
10 DIM User$(2000), Config$(1:100)(80)
Dimensions strings to store the configuration information. User$ stores the entire string of configuration information. Config$ stores the configuration information line by line (up to 80 lines and 100 characters long).
20 INTEGER I, Pos lf
I and Pos lf are used to search through User$ string.
30 OUTPUT 718; "CONFIG?;
The spectrum analyzer sends configuration information.
40 ENTER 718 USING ";K"; User$
Reads the catalog information into the User$ string.
50 I=0
60 WHILE LEN(User$)>1
Loops until the User$ string is empty.
70 I=I+1
```
Programming Commands
Programming Command Descriptions

\begin{verbatim}
80  Pos_lf=POS(User$,CHR$(10))
Checks for line feeds. CHR$(10) represents the line feed, the ASCII code
for a line feed is "10."

90  Config$(I)=User$[1,Pos_lf-2]
Extracts catalog line.

100  OUTPUT CRT;Config$(I)
Displays catalog line.

120  User$=User$[Pos_lf+1]

130  END WHILE

140  END
\end{verbatim}

Description

The \texttt{CONFIG} command is used to query the contents of the Show System screen. It contains analyzer configuration information such as the instrument model number, serial number, firmware revision, and options. After the analyzer has sent the configuration information to the controller, the analyzer sends two line feed characters to the controller.
CONTRAST Display Contrast

Specifies the amount of contrast for the display.

Syntax

```
CONTRAST > number ;
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any integer number.</td>
<td>Values between 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and 255.</td>
</tr>
</tbody>
</table>

Equivalent Key: Contrast.
Step Increment: 1.

Example

```
10 output 718;"CONTRAST 88;"
```

Sets the contrast to a value of 88.

Description

The CONTRAST command specifies the viewing angle of the LCD display in arbitrary units. Be careful not to program a value that results in a display that cannot be viewed; the last value set is used following a power cycle.
CONTS Continuous Sweep

Sets the spectrum analyzer to the continuous sweep mode.

Syntax

```
CONTS
```

Equivalent Key: **Sweep Cont Single** (when Cont is underlined)

Preset State: CONTS

Related Commands: SNGLS, ST, TM

Example

```
10 OUTPUT 718; “CONTS;”
```

Description

The CONTS command sets the spectrum analyzer to continuous sweep mode. In the continuous sweep mode, the spectrum analyzer takes its next sweep as soon as possible after the current sweep (as long as the trigger conditions are met). A sweep may temporarily be interrupted by data entries made from the front panel or over the remote interface.
CORREK Correction Factors On

Returns a “1” if the frequency correction factors are on, a “0” if they are off.

Syntax

Equivalent Key: **Freq Correct**
Related Commands: ALIGN FRQCORON, ALIGN FRQCOROFF

Example

10 OUTPUT 718;"CORREK FREQ?;"
20 ENTER 718;A
30 DISP A

Query Response
DATASTAT Data Status

Indicates the status of certain conditions of the analyzer.

Syntax

```
DATASTAT
```

Example

```
10 OUTPUT 718; "DATASTAT?;"
20 ENTER 718; Datastat
```

Description

The **DATASTAT** command is used to verify that the trace data is valid as well as checking on other analyzer conditions.

The command returns a number representing a 16 bit word. The presence of a 1 (one) indicates that the condition exists. The bit definitions are listed below:

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Decimal Equivalent</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Instrument corrections are off</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Span/frequency change is not finished</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Sweep time setting is not finished</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>Trace A invalid</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>Trace B invalid</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>Trace C invalid</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>Marker is not counted</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>Sweep time too fast for resolution BW or video BW</td>
</tr>
</tbody>
</table>

172
## Programming Commands

### Programming Command Descriptions

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Decimal Equivalent</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>256</td>
<td>TG is unleveled</td>
</tr>
<tr>
<td>9</td>
<td>512</td>
<td>Decrease span for valid frequency count measurement</td>
</tr>
<tr>
<td>10</td>
<td>1024</td>
<td>Degenerate limit line</td>
</tr>
<tr>
<td>11</td>
<td>2048</td>
<td>reserved</td>
</tr>
<tr>
<td>12</td>
<td>4096</td>
<td>Frequency corrections are off</td>
</tr>
<tr>
<td>13</td>
<td>8192</td>
<td>reserved</td>
</tr>
<tr>
<td>14</td>
<td>16384</td>
<td>reserved</td>
</tr>
<tr>
<td>15</td>
<td>32768</td>
<td>Signal track lost the signal</td>
</tr>
</tbody>
</table>

### Query Response

```
Query Response

| number | output termination |
```

001
DATEMODE Date Mode

Allows you to set the format for displaying the real-time clock in either the month, day, year format or the day, month, year format.

Syntax

```
DATEMODE
```

Equivalent Key: **Datemode MDY DMY**

Example

10 OUTPUT 718;"DATEMODE DMY;"
Sets the date mode to day, month, year format.

20 OUTPUT 718;"DATEMODE?;"
Queries the format of the display of the real-time clock.

30 ENTER 718;A$

40 DISP A$

Query Response
DEFCONFIG Default Configuration

Resets the user configuration to the defaults and performs an instrument preset.

Syntax

```
XDEFCONF
```

Equivalent Key: Default Config

Example

```
10 OUTPUT 718; "DEFCONFIG;"
```

Description

The DEFCONFIG command resets the spectrum analyzer to the user configuration originally set at the factory and performs an instrument preset, IP. The following table shows the default user configuration settings which result from executing the command:

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyzer address (Option A4H)²</td>
<td>18</td>
</tr>
<tr>
<td>BAUDRATE (Option 1AX)</td>
<td>9600</td>
</tr>
<tr>
<td>DATEMODE</td>
<td>MDY</td>
</tr>
<tr>
<td>PSTATE</td>
<td>OFF</td>
</tr>
<tr>
<td>POWERON</td>
<td>IP</td>
</tr>
<tr>
<td>ALIGN</td>
<td>AUTOON</td>
</tr>
<tr>
<td>PRNTADRS (Option A4H)</td>
<td>1</td>
</tr>
<tr>
<td>PRNPRT (Options A4H and 1AX)</td>
<td>4 (Parallel)</td>
</tr>
<tr>
<td>TIMEDSP</td>
<td>On</td>
</tr>
</tbody>
</table>

a. No remote command is associated with this command.
DELETE File

Deletes identified file.

Syntax

```
DELETE file
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delimiter</td>
<td>Matching characters marking the beginning and end of the filename</td>
<td>I@ ^ $ % ; !</td>
</tr>
<tr>
<td>Filename</td>
<td>Any valid character</td>
<td>Any valid file name</td>
</tr>
</tbody>
</table>

Equivalent Key: Delete and Delete Now

Related Commands: ERASE

Example

```
10 OUTPUT 718;"DELETE @MY.TRCE;"
```

Deletes MY.TRCE file from spectrum analyzer memory.

Description

The DELETE command is used to delete files from analyzer memory. There are several different types of file data. The type of data, and its corresponding data destination, are shown in the following table. To use the DELETE command you must specify the file name including the appropriate file extension.

Some of the characters that are available to use as delimiters for the file name are also allowed in the file name. Do not use any of the same characters for the delimiters as are used in the file name. The analyzer will interpret the second occurrence of the character as the end delimiter, rather than a part of the file name.
<table>
<thead>
<tr>
<th>Data Source</th>
<th>Type of Data Transferred</th>
<th>File Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRA</td>
<td>Trace</td>
<td>.TRC</td>
</tr>
<tr>
<td>TRB</td>
<td>Trace</td>
<td>.TRC</td>
</tr>
<tr>
<td>TRC</td>
<td>Trace</td>
<td>.TRC</td>
</tr>
<tr>
<td>STATE!</td>
<td>Instrument state</td>
<td>.STA</td>
</tr>
<tr>
<td>LIMILINE</td>
<td>Limit lines</td>
<td>.LIM</td>
</tr>
<tr>
<td>AMPCOR</td>
<td>Amplitude correction factors</td>
<td>.AMP</td>
</tr>
</tbody>
</table>
DEMOD  Demodulation

Turns the demodulator on or off, and selects AM demodulation.

Syntax

Equivalent Key: Demod On Off is similar
Related Commands: DEMODT, SPEAKER

Example

This example demonstrates AM demodulation in a span greater than zero.

10 OUTPUT 718; "IP;FA 500KHZ;"
Sets start frequency.

20 OUTPUT 718; "FB 1600KHZ;"
Sets stop frequency.

30 OUTPUT 718; "TS;MKPK HI;MKCF;"
Places marker on the highest peak and brings the peak to center frequency.

40 OUTPUT 718; "DEMOD ON;DEMOD AM;"
Turns on AM demodulation.

50 OUTPUT 718; "DEMODT 500MS;"
Turns on marker pause.

60 END
Description

Execute “DEMOD ON;” to turn on the demodulator. “DEMOD AM;” selects the demodulation mode, but does not turn on the demodulator.

For AM demodulation in nonzero frequency spans, use DEMODT to set the dwell time.
DEMOLDT Demodulation Time (Dwell Time)

Pauses the sweep at the active marker for the duration of the delay period.

**Syntax**

```
DEMOLDT, time value
```

**Range**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number</td>
<td>2 ms to 100 s</td>
</tr>
</tbody>
</table>

Restrictions: Not available with negative peak detection
Equivalent Key: **Dwell Time On Off**
Step Increment: 1, 2, 5, 10 sequence
Related Commands: DEMOD, DEMOLDT, MKA, MKF, MKFC, MKN, MKOFF, ST

**Example**

```
10 OUTPUT 718;"DEMOLDT 10SC;"
```
Changes the marker pause time to 10 seconds.

**Description**

To turn DEMOLDT off send “DEMOLDT 0;”.

180
Query Response

number \rightarrow output termination \rightarrow
DET Detection Mode

Selects the spectrum analyzer detection mode.

Syntax

```
DET

Equivalent Key: Detector
Preset State: DET POS
Related Commands: MEANTH
```

Example

```
10 OUTPUT 718; "IP;"
Initializes the spectrum analyzer.

20 OUTPUT 718; "SNGLS;"
Activates single-sweep mode.

30 OUTPUT 718; "DET POS;TS;"
Activates the positive-peak detection of trace A.

40 OUTPUT 718; "VIEW TRA;"
Stores results in trace A.

50 OUTPUT 718; "DET SMP;"
Activates sample detection for trace B.

60 OUTPUT 718; "CLRW TRB;TS;"
Measures with trace B.

70 OUTPUT 718; "VIEW TRB;"
Stores results in trace B.
```
80 OUTPUT 718; "AVG TRA, TRB, 2;"
Averages trace A and B with a ratio of 2, and stores the results in trace A.

90 OUTPUT 718; "BLANK TRB;"
Blanks trace B, leaving only averaged results on screen.

100 END

Description
The DET command selects the type of spectrum analyzer detection (positive-peak, sample, or negative) and accesses service-diagnostic detection functions.

POS enables positive-peak detection, which displays the maximum video signal detected over a number of instantaneous samples for a particular frequency.

SMP enables sample detection, which uses the instantaneous video signal value. Video averaging and noise-level markers, when activated, activate sample detection automatically.

NEG enables negative peak detection in sweep times of less than or equal to 200 ms.

Query Response
DL Display Line

Defines the level of the display line in the active amplitude units and displays the display line on the spectrum analyzer screen.

Syntax

```
DL [number] [AUTO|DN|UP|OFF|ON]
```

**Item** | **Description/Default** | **Range**
---|---|---
Number | Any real or integer number. Default units are dBm. | Dependent on the reference level

Equivalent Key: **Display Line On Off**

Preset State: DL OFF

Step Increment: 1 major division

Related Commands: AUNITS, AUTO, LG, LN, NRL, RL, ROFFSET, TH

Example

10 OUTPUT 718; "AUNITS DBM;"

Changes the active amplitude units to dBm.

20 OUTPUT 718; "DL ON;"
Turns on the display line.

30 OUTPUT 718;"DL -5DM;"

Changes display line to -5 dBm.

Description

Activating video trigger mode activates the display line. The AUTO command and “DL OFF;” turn off the display line. See “AUNITs” for more information on changing the active amplitude units.

Query Response

![Diagram showing query response flow](image1.png)
**DN Down**

Reduces the active function by the applicable step size.

**Syntax**

```
DN
```

Equivalent Key: ↓

Related Commands: See the list of active functions listed in the description for DN

**Example**

```
10 OUTPUT 718;"SS 1MHZ;CF 1GHZ;DN;"
```

Sets center frequency to 1 GHz.

```
20 OUTPUT 718;"SP 40MHZ;MKPK;DN;"
```

Decreases the frequency span.

**Description**

Before executing DN, be sure that the function to be decreased is the active function. For example, the second line of the programming example decreases the span, because marker peak (MKPK) is not an active function.

The active functions are AT, CF, CONTRAST, DEMODT, DL, FA, FB, LG, MKA, MKD, MKFCR, MKN, MKPAUSE, MKPX, MKTH, ML, NDB, NRL, NRPOS, PRNTMARB, PRNTMARGT, PRNTRES, RB, ROFFSET, RL, SET-DATE, SET-TIME, SP, SRCALC, SRCAT, SRCPOFS, SRCPSWP, SRCPWR, SS, ST, TH, TIMEBASEC, TIMEBASEF, VB, and VBR.
DONE Done

Allows you to determine when the spectrum analyzer has parsed the spectrum analyzer commands and has started to execute all commands prior to and including DONE. The spectrum analyzer returns a value of "1" when all commands in a command string entered before DONE have been started.

Syntax

\[ \text{DONE} \rightarrow ? \rightarrow ; \rightarrow \text{xDONE} \]

Related Commands: TS

Example 1

10 OUTPUT 718; "IP; SNGLS; CF 1GHz; SP 1GHz; TS; DONE;"

The take sweep (TS) must be completed before the DONE command is executed so that the auto-coupled functions and trace data have been changed before the DONE command is executed.

20 ENTER 718; Done

Stores 1 in computer variable, called Done.

30 DISP Done

Example 2

10 OUTPUT 718; "IP; SNGLS; CF 1GHz; SP 1GHz; DONE;"

There is no TS command before the DONE command in this example. Because of this, the center frequency and span values are set before DONE returns a "1." Functions coupled to span (SP), such as RB, have not been changed, and there is no trace data associated with the new frequency settings.

20 ENTER 718; Done

30 DISP Done
Description

As shown by the example, if a take sweep (TS) precedes the DONE command, DONE is executed after all the preceding commands have been completed. Use a take sweep (TS) to ensure all previous commands have completed before DONE returns a “1.”

Query Response

![Diagram](image-url)
ERASE Erase

Purges all state and trace registers, deletes limit lines and AMPCOR information.

Syntax

```
ERASE
```

Equivalent Key: `ALL MEMORY`.
Related Commands: `PSTATE`.

Example

```
10 OUTPUT 718;"ERASE;"
```
EXITSHOWSYS Exit Config/Show System Screen

Erases the display for the function that shows the analyzer’s current system configuration.

Syntax

```
EXITSHOWSYS ;
```

Related Softkey: **Show System, Done**
Related Commands: SHOWSYS

Example

```
10 OUTPUT 718;"EXITSHOWSYS;"
```

Description

The show system function prints the analyzer configuration on the display screen. EXITSHOWSYS is used to remove the configuration screen and return to normal instrument operation.
FA Start Frequency

Specifies the start frequency.

Syntax

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is Hz.</td>
<td>Frequency range of the spectrum analyzer</td>
</tr>
</tbody>
</table>

Equivalent Key: **Start Freq**

Step Increment: Frequency span divided by 10

Related Commands: CF, FB, FOFFSET, FS, MKF, SP

Example

10 OUTPUT 718;"FA 88MHZ;FB 108MHZ;"

Sets the start frequency to 88 MHz, the stop frequency to 108 MHz.

20 OUTPUT 718;"FA?;"

Returns the start frequency.
Programming Commands
Programming Command Descriptions

30 ENTER 718;Freq
Stores the response from the spectrum analyzer.

40 DISP Freq
Displays the frequency on the computer screen.

**Description**

The FA command specifies the start frequency value. The start frequency is equal to the center frequency minus the span divided by two (FA = CF - SP/2). Changing the start frequency changes the center frequency and span.

**Query Response**

![Diagram of query response process]

192
**FB Stop Frequency**

Specifies the stop frequency.

**Syntax**

```
FB Stop Frequency
```

```
XFE
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is Hz.</td>
<td>Frequency range of the spectrum analyzer</td>
</tr>
</tbody>
</table>

Equivalent Key: **Stop Freq**

Step Increment: Frequency span divided by 10

Related Commands: CF, FA, FOFFSET, FS, MKF, SP

**Example**

10 OUTPUT 718; "FA 88MHZ; FB 108MHZ;"

Sets the start frequency to 88 MHz, the stop frequency to 108 MHz.

20 OUTPUT 718; "FB?;"

193
Returns the stop frequency.

30 ENTER 718;Freq
Stores the response from the spectrum analyzer.

40 DISP Freq
Displays the frequency on the computer screen.

Description
The FB command specifies the stop frequency value. The stop frequency is equal to the center frequency plus the span divided by two (FB = CF + SP/2). Changing the stop frequency changes the center frequency and span.

Query Response

[Diagram showing flow of data: number → output termination]
**FOFFSET Frequency Offset**

Specifies the frequency offset for all absolute frequency readouts such as center frequency.

**Syntax**

```
FOFFSET [frequency offset value]
```

**Item** | **Description/Default** | **Range**
--- | --- | ---
Number | Any real or integer number. Default unit is Hz. | |

Equivalent Key: `Freq Offset`

Preset State: 0 Hz

Related Commands: CF, FA, FB, MKN, MKF, MKSP, MKSS

**Example**

```
10 OUTPUT 718;"IP;FA 200MZ;"
```

Initializes spectrum analyzer. Sets start frequency.

```
20 OUTPUT 718;"FB 1GZ;"
```

Sets stop frequency.

```
30 OUTPUT 718;"TS;MKPK HI;"
```

Places marker on signal peak.

```
40 OUTPUT 718;"MF;"
```

---

195
Finds frequency of marker.

50 ENTER 718;A

60 PRINT A
Prints frequency of marker.

70 OUTPUT 718;"FOFFSET 500MZ;"
Adds a frequency offset.

80 OUTPUT 718;"TS;MF;"
The frequency of the marker now is the frequency of the signal peak plus
the frequency offset.

90 ENTER 718;A

100 PRINT A
The displayed frequency is 500 MHz greater than the frequency displayed
in line 60.

110 END

Description
The FOFFSET command selects a value that offsets the frequency scale for all
absolute frequency readouts (for example, center frequency). Relative values such
as span and marker delta are not offset.

After execution, the FOFFSET command displays the frequency offset in the
active function readout. When an offset is in effect, it is displayed beneath the bot-
tom graticule line on the spectrum analyzer screen.

Execute “FOFFSET 0;” or “IP;” to turn off the offset.

Query Response

```
<table>
<thead>
<tr>
<th>number</th>
<th>output termination</th>
</tr>
</thead>
</table>
```

001
FREF Frequency Reference

Returns the source of the 10 MHz frequency reference.

Syntax

```
FREF
```

Example

```
10 OUTPUT 718;"FREF?;"
20 ENTER 718;SOURCE$
30 PRINT "THE FREQUENCY REF SOURCE IS";SOURCE IS";SOURCE$
40 END
```

Description

The FREF command is used to query the source of the frequency reference, which is supplied from an internal (INT) or external (EXT) 10 MHz source.

Query Response
FS Full Span

Sets the frequency span of the spectrum analyzer to full span.

Syntax

FS XFS

Equivalent Key: Full Span
Related Commands: CF, FA, FB, HNLOCK, SP, SS

Example

10 OUTPUT 718;"FS;"

Puts the spectrum analyzer in full-span mode.

Description

The FS command selects both the start frequency and the stop frequency according to the frequency span of the spectrum analyzer. Resolution bandwidth, video bandwidth, and sweep time are all set to auto-coupled.

<table>
<thead>
<tr>
<th>Spectrum Analyzer Model</th>
<th>Start Frequency</th>
<th>Stop Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP ESA-L1500A</td>
<td>0 Hz</td>
<td>1.5 GHz</td>
</tr>
</tbody>
</table>
GETPRNT Get Print

Initiates output of the spectrum analyzer display to a printer. GETPRNT is meant to be used within a remote program.

Syntax

```
GETPRNT number, color or number
```

Equivalent Key: Print

Related Commands: GETPLOT, PRNPRT

Example for the HP-IB Interface

This example illustrates how you can use a softkey to print the spectrum analyzer display, and then have the printer perform a page feed. This example assumes that the printer is at address 1 and the spectrum analyzer is at address 18. (This example is only valid for HP 9000 series 200 and 300 computers.)

```
10 OUTPUT 718; PRNPRT 4;
Prints the contents of the display in monochrome to an HP Deskjet 850 printer.

20 OUTPUT 718; "GETPRNT 14, 0;"
Prints the contents of the display in monochrome to an HP Deskjet 850 printer.

30 OUTPUT 718; "GETPRNT 14, 1;"
Prints the contents of the display in color to an HP Deskjet 850 printer.
```

Description

The data is output in HP raster graphics or Epson graphics format.
The following table lists available printers and their associated numbers, for use in the GETPRNT remote command. The printers listed in your spectrum analyzer may be different than those listed in the table due to differences in firmware versions.

<table>
<thead>
<tr>
<th>Printer Model</th>
<th>Printer Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP DeskJet 3 10</td>
<td>0</td>
</tr>
<tr>
<td>HP DeskJet 320</td>
<td>1</td>
</tr>
<tr>
<td>HP DeskJet 340</td>
<td>2</td>
</tr>
<tr>
<td>HP DeskJet 400</td>
<td>3</td>
</tr>
<tr>
<td>HP DeskJet 500</td>
<td>4</td>
</tr>
<tr>
<td>HP DeskJet 500C</td>
<td>5</td>
</tr>
<tr>
<td>HP DeskJet 520</td>
<td>6</td>
</tr>
<tr>
<td>HP DeskJet 540</td>
<td>7</td>
</tr>
<tr>
<td>HP DeskJet 550C</td>
<td>8</td>
</tr>
<tr>
<td>HP DeskJet 560C</td>
<td>9</td>
</tr>
<tr>
<td>HP DeskJet 600</td>
<td>10</td>
</tr>
<tr>
<td>HP DeskJet 660C</td>
<td>11</td>
</tr>
<tr>
<td>HP DeskJet 680C</td>
<td>12</td>
</tr>
<tr>
<td>HP DeskJet 690C</td>
<td>13</td>
</tr>
<tr>
<td>HP DeskJet 850C</td>
<td>14</td>
</tr>
<tr>
<td>HP DeskJet 870C</td>
<td>15</td>
</tr>
<tr>
<td>HP DeskJet 1200C</td>
<td>16</td>
</tr>
<tr>
<td>HP DeskJet 1600C</td>
<td>17</td>
</tr>
<tr>
<td>HP DeskJet Plus</td>
<td>18</td>
</tr>
<tr>
<td>HP DeskJet Portable</td>
<td>19</td>
</tr>
<tr>
<td>HP DeskJet</td>
<td>20</td>
</tr>
<tr>
<td>HP ThinkJet</td>
<td>21</td>
</tr>
<tr>
<td>HP QuietJet</td>
<td>22</td>
</tr>
<tr>
<td>Printer Model</td>
<td>Printer Number</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>HP PaintJet</td>
<td>23</td>
</tr>
<tr>
<td>HP LaserJet III</td>
<td>24</td>
</tr>
<tr>
<td>HP LaserJet 4</td>
<td>25</td>
</tr>
<tr>
<td>HP LaserJet 4L</td>
<td>26</td>
</tr>
<tr>
<td>HP LaserJet 4P</td>
<td>27</td>
</tr>
<tr>
<td>HP LaserJet 4Plus</td>
<td>28</td>
</tr>
<tr>
<td>HP LaserJet 5</td>
<td>29</td>
</tr>
<tr>
<td>HP LaserJet 5L</td>
<td>30</td>
</tr>
<tr>
<td>HP LaserJet 5P</td>
<td>31</td>
</tr>
<tr>
<td>Epson FX-85</td>
<td>32</td>
</tr>
<tr>
<td>Epson LQ-570</td>
<td>33</td>
</tr>
</tbody>
</table>
GRAT Graticule

Turns on or off the graticule.

Syntax

```
Equivalent Key: Graticule On Off
Preset State: GRAT ON
Related Commands: ANNOT
```

Example

```
10 OUTPUT 718; "GRAT OFF;"
Turns off the graticule.

20 OUTPUT 718; "GRAT?;"
Gets response from the spectrum analyzer.

30 DISP Grat$
Displays OFF on the computer screen.
```
Query Response

ON

OFF

output termination
HD Hold Data Entry

Disables data entry via the spectrum analyzer numeric keypad, knob, or step keys. The active function readout is blanked, and any active function is deactivated.

Syntax

```
HD
```

Equivalent Key: Esc

Related Commands: Any active function. See the description below for a list of the active functions

Example

```
10 OUTPUT 718;"HD;"
```

Disables the active function and clears the active function block area on the spectrum analyzer screen.

```
20 OUTPUT 718;"CF 600MHZ;HD;700MHZ;"
```

This will leave the center frequency at 600 MHz, because HD deactivates any current function.

Description

The active functions are AT, BAUDRATE, CF, CONTRAST, DEMODT, DL, FA, FB, INZ, LG, MKA, MKD, MKFC, MKFCR, MKN, MKPAUSE, MKPX, MKTH, ML, MODE, NDB, NRL, NRPOS, PREAMPG, PRNTMARGB, PRNTMARGT, PRNTRES, RB, ROFFSET, RL, SETDATE, SETTIME, SP, SRCAT, SRCPOFS, SRCSTP, SRCPSWP, SRCPWR, SS, ST, TH, TIMEBASEC, TIMEBASEF, TIMEDATE, VAVG, VB, and VBR.
ID Identify

Returns the spectrum analyzer model number to the controller (for example, “HP ESA-L1500A”).

Syntax

Equivalent Key: Show Config
Related Commands: REV, SER

Example

10 ALLOCATE A$[50]
Allocates string to hold model number.

20 OUTPUT 718;“ID;”
Gets model number.

30 ENTER 718;A$
Transfers number to computer.

40 DISP A$
Displays model number.

50 END

Query Response
INVERTLCD Inverse Video

Inverts the LCD display (black for white, or white for black).

Syntax

```
INVERTLCD
```

Equivalent Key: **Inverse Video On Off**

Preset State: Inverse Video Off

**Example**

```
10 OUTPUT 718;"INVERTLCD on;"
Inverts the display.

20 OUTPUT 718;"INVERTLCD?;"

30 ENTER 718;INVERTLCD$
Gets response from the spectrum analyzer.

40 DISP INVERTLCD$
Displays OFF on the computer screen.
```
Description

Inverts the LCD display and also affects the VGA monitor colors. The colors are changed by assigning a color setting number equal to 63 minus the original color number. For example, red 10, green 20, and blue 30 would be changed to red 53, green 43, and blue 33. INVERTLCD can be used in environments where viewing the normal display color is undesirable.

Query Response

![Diagram showing query response with 'ON' and 'OFF' as inputs and 'output termination' as output.]
INZ Input Impedance

Specifies the value of input impedance expected at the active input port.

Syntax

```
INZ
```

Equivalent Softkey: Input Z Corr 50 75

Preset Value: The value is unaffected by preset

Related Commands: AUNITS

Example

```
10 OUTPUT 718;"INZ 75;"
```

Changes input impedance to 75 ohms.

```
20 OUTPUT 718;"AUNITS V;"
```

Changes amplitude units to volts.

Description

The actual impedance can be affected only by internal hardware. With the exception of Option 1DP or 1DQ, the spectrum analyzer hardware supports 50 Ω only.

The INZ command is used for computation purposes during power or voltage conversions.

The INZ command affects only the amplitude results that are reported in absolute relative power (dBm units or watts). (See “AUNITS.”)
Query Response

output termination
IP Instrument Preset

Performs an instrument preset.

Syntax

```
IP ;
```

Equivalent Key: Preset is similar

Example

```
10 OUTPUT 718;"IP;"
```

Description

The instrument preset command, IP, executes the following commands:

- **AMPCOR**: Turns off amplitude correction factors.
- **ANNOT**: Turns on annotation.
- **AT**: Couples the attenuation to the reference level.
- **AUNITS**: Loads the amplitude units from a configuration location in spectrum analyzer memory.
- **BLANK B, BLANK C**: Blanks trace B and trace C.
- **CLRWA**: Clears and writes trace A.
- **CONTS**: Selects continuous sweep mode.
- **DEMODT**: Sets demodulation dwell time to 500 ms.
- **DET**: Selects positive peak detection.
- **DL**: Turns off the display line.
- **FOFFSET**: Sets the frequency offset to 0 Hz.
- **GRAT**: Turns on the graticule.
- **LG**: Selects 10 dB per division log scale.
- **LIMIDISP**: Sets LIMIDISP to AUTO.
**LIMIHALF**
Clears any limit line trace specified by LIMIHALF.

**LIMITEST**
Turns off limit line testing.

**MDS**
Selects data size of one word, which is two S-bit bytes.

**MKFCR**
Marker counter resolution is set to AUTO, but a calculated value other than 0 may be returned if the marker counter resolution is queried.

**MKNOISE**
Turns off noise markers.

**MKOFF**
Turns off all markers.

**MKPAUSE**
Turns off marker pause mode.

**MKPX**
Minimum excursion for peak identification is set to 6 dB.

**MKREAD**
Sets marker readouts to frequency.

**MKTH**
Sets the marker threshold to -90 dBm.

**MKTRACK**
Turns off signal track.

**ML**
Sets mixer level to -10 dBm.

**NDB**
Sets the number of dB for the NDBPNT measurement to -3 dB.

**NORMLIZE**
Turns off normalization.

**NRPOS**
Sets the normalize reference level position to 10.

**RB**
Couples the resolution bandwidth to the frequency span.

**RFCALIB**
Turns off the 50 MHz oscillator.

**RL**
Sets reference level to 0 dBm.

**ROFFSET**
Sets reference offset to 0.

**RQS 41**
Allows SRQ 101 for operator notification, SRQ 110 for illegal commands, or SRQ 140 for broken hardware.

**SPEAKER**
Turns on the speaker.

**SRCPSWP**
Sets the source power sweep to off (Option 1DN or 1DQ only).

**SRCPWR**
Sets the source power level to -10 dBm (Option 1DN or 1DQ only).

**SS**
Couples the center frequency step size to the span.

**STB**
Clear the status byte.
**TH**  
One division above bottom graticule line, threshold line off.

**TITLE**  
Clears the title from the spectrum analyzer screen.

**TM**  
Selects free run trigger mode.

**TDF**  
Selects parameter units output format.

**TRC**  
Sets the trace values to the top graticule line.

**VAVG**  
Turns off video averaging and sets the video averaging limit to 100.

**VB**  
Couples the video bandwidth to the resolution bandwidth.

**VBR**  
Sets VBR to 0.300.

Instrument preset automatically occurs when you turn on the spectrum analyzer. IP is a good starting point for many measurement processes. When IP is executed remotely, the spectrum analyzer does not necessarily execute a complete sweep, however. You should execute a take sweep (TS) to ensure that the trace data is valid after an IP.
LG Logarithmic Scale

Specifies the vertical graticule divisions as logarithmic units, without changing the reference level.

Syntax

![Diagram of LG Logarithmic Scale syntax]

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default units are dB.</td>
<td>0.1 to 20 dB.</td>
</tr>
</tbody>
</table>

Equivalent Key: ScaleLogLin (when Log is underlined)

Preset State: 10 dB

Related Commands: LN

Example

```
10 OUTPUT 718; "LG 1DB;"
```

Description

The vertical scale can be specified (in tenths) from 0.1 to 0.9 dB, or in integers from 1 to 20 dB per graticule division.
Query Response
LIMIDEL Delete limit line Table

Deletes all upper and lower segments in the current limit line table.

Syntax

\[
\begin{align*}
\text{LIMIDEL} \\
\text{XLIMIDEL}
\end{align*}
\]

Equivalent Key: **Delete Limits**

Related Commands: LIMIFT, LIMIREL, LIMISEG, LIMISEGT, SEGDEL

Example

\[
\text{20 OUTPUT 718;"LIMIDEL;"}
\]

Description

Use LIMIDEL before entering a new limit line.

**NOTE:**

Use SAVE if you want to save the current limit line table. LIMIDEL does not affect stored limit line data.

LIMIDEL sets LIMIREL OFF (specifies that the limit line is fixed) and LIMIFT FREQ (specifies that the limit line is based on frequency). See “LIMILINE” for more information about limit line construction.
LIMIDISP Limit Line Display

Controls when the limit line (or limit lines) are displayed.

Syntax

LIMIDISP

Parameters:
- **ON**
  - Turns on the limit line display.
- **OFF**
  - Turns off the limit line display.
- **AUTO**
  - Allows LIMITEST to control the display of the limit lines. If LIMITEST is on, the limit lines will be displayed. If LIMITEST is off, the limit lines will not be displayed.
- **UPPER**
  - Displays the upper limit line only.
- **LOWER**

Equivalent Key: Limit Display Y N Auto

Preset Value: AUTO

Related Commands: LIMILINE, LIMITEST

Example

10 OUTPUT 718;"LIMIDISP ON;"

Displays any portion of the limit lines that are currently within the spectrum analyzer screen boundaries.

Description

If a limit line is currently in spectrum analyzer memory, you can use LIMIDISP to control the display of the limit lines. The parameters of LIMIDISP do the following:

- **ON**
  - Turns on the limit line display.
- **OFF**
  - Turns off the limit line display.
- **AUTO**
  - Allows LIMITEST to control the display of the limit lines. If LIMITEST is on, the limit lines will be displayed. If LIMITEST is off, the limit lines will not be displayed.
- **UPPER**
  - Displays the upper limit line only.
LOWER  Displays the lower limit line only.

Query Response

ON  OFF  AUTO  UPPER  LOWER  output termination

OLIMIDISP
LIMIFAIL Limits Failed

Returns a “0” if the last measurement sweep of trace A is equal to or within the limit line bounds.

Syntax

```
LIMIFAIL
```

Related Commands: LIMILINE, LIMISEG, LIMISEGT, LIMITEST

Example

```
10 OUTPUT 718;"IP;SNGLS;CF300MHZ;SP100MHZ;"
Initializes spectrum analyzer and changes the frequency and span settings.

20 OUTPUT 718;"LIMIDEL;"
Deletes any limit line tables, sets the table type to fixed.

30 OUTPUT 718;"LIMIHALF UPPER;"
Specifies the upper limit line table.

40 OUTPUT 718;"LIMISEG 250MHZ,-60DB,FLAT;"
Creates an entry for the upper limit line table. Because the LIMISEG command is used, the limit line will be based on the frequency.

50 OUTPUT 718;"LIMISEG 290MHZ,-60DB,SLOPE;"

60 OUTPUT 718;"LIMISEG 295MHZ,-15DB,FLAT;"

70 OUTPUT 718;"LIMISEG 305MHZ,-15DB,SLOPE;"

80 OUTPUT 718;"LIMISEG 310MHZ,-60DB,FLAT;"

90 OUTPUT 718;"LIMISEG 910MHZ,-60DB,FLAT;"
```
100 OUTPUT 718;"LIMITEST ON;TS;"
Turns on limit line testing.

110 OUTPUT 718;"LIMIFAIL?;"
Returns the status of the limit line testing.

120 ENTER 718;A

130 DISP A
Displays the result.

140 END

Description
LIMIFAIL returns one of the following values:

- 0 indicates that the measurement sweep was within the limit line bounds.
- 1 indicates that the measurement sweep failed the lower limit.
- 2 indicates that the measurement sweep failed the upper limit.
- 3 indicates that the measurement sweep failed both the lower and upper limits.
- 4 indicates that no test was performed. A “4” is returned if LIMITEST is set to OFF.

Query Response
LIMIFT Select Frequency or Time Limit Line

Selects how the limit line segments are defined: according to frequency, or according to the sweep time setting of the spectrum analyzer.

Syntax

\[
\text{LIMIFT } \text{TIME} ;
\]

Equivalent Key: \textbf{X Axis Units Freq Time}

Related Commands: \text{LIMIDEL, LIMILINE, LIMIHALF, LIMIREL, LIMISEG, LIMISEGT, SEGDEL}

Example

\[
10 \text{ OUTPUT 718;} \text{"LIMIFT TIME;"}
\]

If the current limit line table is a frequency limit line table, it is purged. LIMIFT TIME places the limit line segments on the spectrum analyzer display with respect to the sweep time of the spectrum analyzer.

Description

If you execute “LIMIFT TIME;” or LIMISEGT the limit line segments are placed on the spectrum analyzer display with respect to the sweep time setting of the spectrum analyzer. If you execute “LIMIFT FREQ;” or LIMISEG the limit line segments are placed according to the frequency that is specified for each segment. If a limit line has already been defined, changing the LIMIFT setting clears the existing limit line.

Query Response
LIMIHALF Select Upper or Lower Limit Line

Determines whether the limit line entries are treated as upper amplitude values, lower amplitude values.

Syntax

```
LIMIHALF
```

Equivalent Key: Select Line Upper Lower

Related Commands: LIMILINE, LIMISEG, LIMISEGT, SEGDEL

Example

This example uses LIMIHALF to enter segments into the upper limit line table, and then to enter a segment into the lower limit line table (upper and lower limit lines are treated as separate tables). Line 60 demonstrates entering a segment into a combined upper and lower limit line table.

```
10 OUTPUT 718;"LIMIDEL;"
Deletes the current limit line table, sets the table type to fixed.
20 OUTPUT 718;"LIMIHALF UPPER;"
Specifies the upper limit line table.
30 OUTPUT 718;"LIMIPT FREQ;"
Selects a limit line based on frequency.
40 OUTPUT 718;"LIMISEG 300MHZ,-30DB,SLOPE;"
Enters a segment into the upper limit line table. Because the LIMISEG command is used, the limit line table will be based on frequency.
50 OUTPUT 718;"LIMIHALF LOWER;"
```
Specifies the lower limit line table.

60 OUTPUT 718; "LIMISEG 300MHZ, -70DB, SLOPE;"
Enters a segment into the lower limit line table.

70 OUTPUT 718; "LIMIDISP ON;"
Displays the limit lines.

80 END

Description

Use LIMIHALF in conjunction with LIMISEG, LIMISEGT. Specify LIMIHALF UPPER or LIMIHALF LOWER before using LIMISEG or LIMISEGT.

The LIMIHALF command determines whether the limit line table entries are to be treated separately (upper or lower) or together (upper and lower) when deleting a segment with SEGDEL (see “SEGDEL”).

Query Response

![Diagram of LIMIHALF query response]

QLIMIHAL
LIMILINE Limit Lines

Outputs the current limit line table definitions.

Syntax

```
LIMILINE
```

Related Commands: LIMIFT, LIMIREL, LIMIHALF, LIMIREL, LIMISEG, LIMISEGT, LIMITEST

Example

```
10   DIM States$(2000)
Dimensions an array to store the limit line information.
20   OUTPUT 718;"IP;CF300MHZ;SP100MHZ;"
Initializes spectrum analyzer.
30   OUTPUT 718;"LIMIDEL;"
Deletes any limit line tables, sets the table type to fixed.
40   OUTPUT 718;"LIMIHALF UPPER;"
Specifies the upper limit line table.
45   OUTPUT 718;"LIMIFT FREQ;"
Selects a limit line based on frequency.
50   OUTPUT 718;"LIMISEG 250MHz,-60DB,FLAT;"
Enters a value for the upper limit line table. Because the LIMISEG command is used, the limit line segment is for a limit line based on frequency.
60   OUTPUT 718;"LIMISEG 290MHz,-60DB,SLOPE;"
70   OUTPUT 718;"LIMISEG 295MHz,-15DB,FLAT;"
80   OUTPUT 718;"LIMISEG 305MHz,-15DB,SLOPE;"
90   OUTPUT 718;"LIMISEG 310MHz,-60DB,FLAT;"
```
Programming Commands
Programming Command Descriptions

100 OUTPUT 718;"LIMISEG 910MHZ,-60DB,FLAT;"

110 OUTPUT 718;"LIMILINE?;"
Gets the current limit line table definitions.

120 ENTER 718 USING "#,-K";States$
Enters information into array.

130 PRINT States$
Prints the current limit line table definitions.

140 END

Description
LIMILINE is used to query the current limit line. Executing LIMILINE returns an ASCII string containing the commands needed to create the limit line.

Use these commands (in the order given) to build a limit line:

1 Use LIMIDEL to clear the limit line table.
2 Use LIMIFT to select a limit line that is either based on frequency or sweep time.
3 Use LIMIREL to determine whether the values of the limit line are absolute values or positioned relative to the reference level and center frequency settings.
4 Use LIMIHALF, LIMISEG, LIMISEGT to enter the limit line segments. (Use LIMISEG for a limit line based on frequency; use LIMISEGT for a limit line based on sweep time.)
5 Use the LIMIDISP command to select if the limit line is displayed or not.
6 Use the LIMITEST command to turn on limit line testing.
7 Use the LIMIFAIL command to determine if the measurement sweep passed or failed the limit line boundaries.

Enabling limit line testing: When limit testing is enabled, the segments in the current table are interpolated into the limit line traces according to the current span and center frequency or sweep time of the spectrum analyzer. After the sweep, each value in trace A is compared to its corresponding value in the limit line traces. If the current limit line table is empty (for instance after using the command LIMIDEL) and limit testing is enabled, then the limit line traces are blanked and set to out-of-range values. By using the SUB, MKPK HI, and MKF? commands, you can read the point of greatest difference between the trace and limit line. See “LIMITEST” for more information about limit line testing.
**Saving the limit line table:** Once you have built the limit line, you can save the limit line table in spectrum analyzer memory. Use SAVE to store the limit line table in spectrum analyzer memory.

**Query Response**

The query response is a character string consisting of LIMILINE, LIMIREL, LIMIHALF, LIMISEG, or LIMISEGT commands.
LIMIREL Relative Limit Lines

Specifies whether the current limit lines are fixed or relative.

Syntax

Equivalent Key: **Limits Fixed Rel**
Related Commands: LIMIDEL, LIMIFT, LIMILINE

Example

10 OUTPUT 718; "LIMIFT FREQ;"
Selects a limit line based on frequency.

20 OUTPUT 718; "LIMIREL ON;"
Specifies that the limit line will be relative to the reference-level and center-frequency settings.

Description

You should use **LIMIFT** to select whether the limit lines are based on frequency or sweep time before using LIMIREL, because changing between a frequency or sweep time limit line purges the current limit line table and sets LIMIREL to **OFF**.
LIMIREL and the reference level: Regardless of whether the limit line is based on frequency or sweep time, LIMIREL determines if the amplitude parameter in a limit line table represents absolute values or relative values. If LIMIREL is set to OFF, the limit lines amplitude values are specified in absolute amplitude and do not depend on the reference level (RL) setting. If LIMIREL is set to On, the limit line amplitude values are relative to the current reference level (RL) setting.

For limit lines that are based on frequency: The LIMIREL command determines whether the frequency parameter in a limit line table represent absolute or relative values that are referenced to the center-frequency settings.

- Executing “LIMIREL OFF;” specifies that the frequency values in a limit line table are fixed values, and the limit line is positioned accordingly. Fixed limit lines are specified in absolute frequency and do not depend upon the center frequency value.

- Executing “LIMIREL ON;” specifies that the frequency values in a limit line table are relative values and positions the limit line relative to the center-frequency settings. Relative limit lines are specified in relative frequency and are positioned with respect to the current center frequency. When the current center frequency value is changed, the segment frequencies are converted according to the current center frequency value.

For limit lines that are based on the sweep time: Limit lines that are based on sweep time are always relative to the start time, and the horizontal position of the limit line is not affected by the setting of LIMIREL.

Query Response
LIMISEG Enter limit line Segment for Frequency

Adds new segments to the current frequency limit line in either the upper limit line or the lower limit line.

Syntax

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is dBm.</td>
<td>Varies with FOFFSET and ROFFSET.</td>
</tr>
</tbody>
</table>

Equivalent Key: Segment
Related Commands: LIMIDEL, LIMILINE, LIMIHALF, LIMIREL, SEGDEL
Example

10 \texttt{OUTPUT 718;"IP;SNGLS;CF300MHZ;SP100MHZ;RB 3MHZ;"}  
Initializes spectrum analyzer, changes the frequency, span, and bandwidth.

20 \texttt{OUTPUT 718;"LIMDEL;"}  
Deletes the current limit line table, sets the table type to fixed.

30 \texttt{OUTPUT 718;"LIMIHALLF UPPER;"}  
Specifies the upper limit line table.

35 \texttt{OUTPUT 718;"LIMIFT FREQ;"}  
Selects a limit line based on frequency.

40 \texttt{OUTPUT 718;"LIMISEG 250MHZ,--60DB,FLAT;"}  
Adds segment to the upper limit line table.

50 \texttt{OUTPUT 718;"LIMISEG 290MHZ,--60DB,SLOPE;"}  

60 \texttt{OUTPUT 718;"LIMISEG 295MHZ,--15DB,FLAT;"}  

70 \texttt{OUTPUT 718;"LIMISEG 305MHZ,--15DB,SLOPE;"}  

80 \texttt{OUTPUT 718;"LIMISEG 310MHZ,--60DB,FLAT;"}  

90 \texttt{OUTPUT 718;"LIMISEG 910MHZ,--60DB,FLAT;"}  

100 \texttt{OUTPUT 718;"LIMIHALLF LOWER;"}  
Specifies the lower limit line table.

110 \texttt{OUTPUT 718;"LIMISEG 250MHZ,--75DB,FLAT;"}  
Adds segment to the lower limit line table.

120 \texttt{OUTPUT 718;"LIMISEG 910MHZ,--75DB,FLAT;"}  

130 \texttt{OUTPUT 718;"LIMITEST ON;TS;"}  
Enables limit line testing.

140 \texttt{OUTPUT 718;"LIMIFAIL?;"}  
Returns the result of limit line testing.

150 \texttt{ENTER 718;A}
Programming Commands
Programming Command Descriptions

160  DISP A
Displays the result.

170  END

Description

If the current limit line table contains lines based on sweep time (as opposed to a limit line based on the frequency), executing LIMISEG will clear the current sweep time limit line table, and set LIMIREL to OFF.

Each limit line segment is specified with a starting frequency, an amplitude, and a segment type. The segment type defines how the line segment is to extend from its starting point to the next segment. The segment types are FLAT, SLOPE, and POINT.

- FLAT draws a zero-slope line between the coordinate point of the current segment and the coordinate point of the next segment, producing limit line values equal in amplitude for all frequencies between the two points. If the amplitude values of the two segments differ, the limit line will “step” to the value of the second segment.

- SLOPE draws a straight line between the coordinate point of the current segment and the coordinate point of the next segment, producing limit line values for all frequencies between the two points.

- POINT specifies a limit value for the coordinate point, and no other frequency points, so that a POINT segment specifies a limit value for a single frequency. For an upper limit line, a POINT segment is indicated by a line drawn from the coordinate point, to a point that is vertically off the top of screen. For a lower limit line, a POINT segment is indicated by a line drawn from the coordinate point, to a point that is vertically off the bottom of screen. The POINT segment type should be used as the last segment in the limit line table. However, if the last segment in the table is not specified as the POINT segment type, an implicit point is used automatically. If a visible POINT segment at the right edge of the display is not desired, add an explicit last point segment (higher in frequency than the stop frequency) to the limit line table.

Segments are sorted according to starting frequency. A maximum of 20 segments can be defined in each of the upper and lower halves of a limit line. When the segment type is omitted, the last type given (or SLOPE if no previous type has been given) is used. Use LIMISEG if you want to enter amplitude data in the upper or lower limit lines.
LIMISEGT Enter limit line Segment for Sweep Time

Adds new segments to the current sweep time limit line in either the upper limit line or the lower limit line.

Syntax

```
LIMISEGT, time, us/MS, trace element
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. For the sweep time, the default unit is seconds. For the amplitude value, the default unit is dBm.</td>
<td>The range of the sweep time is 0 to 100 s. The range of the amplitude varies with ROFFSET.</td>
</tr>
</tbody>
</table>

Equivalent Key: Segment
Related Commands: LIMIDEL, LIMIFT, LIMIHALF, LIMILINE, LIMIREL, SEGDEL
Example

10 OUTPUT 718; "LIMIDEL;"
Deletes the current limit line table, sets the table type to fixed.

20 OUTPUT 718; "LIMIHALF UPPER; "
Specifies the upper limit line table.

30 OUTPUT 718; "LIMIFT TIME;"
Selects a limit line based on the sweep time.

40 OUTPUT 718; "LIMISEGT 0MS, -60DB, FLAT;"
Adds segment to the upper limit line table.

50 OUTPUT 718; "LIMISEGT 6MS, -60DB, SLOPE;"

60 OUTPUT 718; "LIMISEGT 8MS, -15DB, FLAT;"

70 OUTPUT 718; "LIMISEGT 11MS, -20DB, SLOPE;"

80 OUTPUT 718; "LIMISEGT 14MS, -60DB, FLAT;"

90 OUTPUT 718; "LIMISEGT 20MS, -60DB, POINT;"

100 OUTPUT 718; "LIMIHALF LOWER;"
Specifies the lower limit line table.

110 OUTPUT 718; "LIMISEGT 0MS, -75DB, FLAT;"
Adds segment to the lower limit line table.

120 OUTPUT 718; "LIMISEGT 20MS, -75DB, POINT;"

130 OUTPUT 718; "LIMITEST ON; TS;"
Enables limit line testing.

140 OUTPUT 718; "LIMIFAIL?;"
Returns the result of limit line testing.

150 ENTER 718; A

160 DISP A
Displays the result.
Description

Each limit line segment is specified with a starting sweep time, an amplitude, and a segment type.

NOTE:

If the current limit line table contains limit lines based on frequency (as opposed to a limit line based on the sweep time), executing LIMISEGT will clear the current frequency limit line table, and set LIMIREL to OFF.

Starting sweep time: When you specify the starting sweep time, you are specifying the starting sweep time with respect to the sweep time of the spectrum analyzer. For example, if you specify a starting sweep time of 0, the limit line segment will start at the left side of the spectrum analyzer display.

Segment type: The segment type defines how the line segment is to extend from its starting point to the next segment. The segment types are FLAT, SLOPE, and POINT.

- FLAT draws a zero-slope line between the coordinate point of the current segment and the coordinate point of the next segment, producing limit line values equal in amplitude for all sweep times between the two points. If the amplitude values of the two segments differ, the limit line will “step” to the value of the second segment.

- SLOPE draws a straight line between the coordinate point of the current segment and the coordinate point of the next segment, interpolating amplitude values for all sweep times between the two points.

- POINT specifies a limit value for the coordinate point, and no other sweep time points, so that a POINT segment specifies an amplitude value for a single sweep time. For an upper limit line, a POINT segment is indicated by a line drawn from the coordinate point, to a point that is vertically off the top of the gaticule area. For a lower limit line, a POINT segment is indicated by a line drawn from the coordinate point, to a point that is vertically off the bottom of the gaticule area. The POINT segment type should be used as the last segment in the limit line table. However, if the last segment in the table is not specified as the POINT segment type, an implicit point is used automatically. If a visible POINT segment at the right edge of the display is not desired, add an explicit last point segment to (higher in sweep time than the current sweep time setting of the spectrum analyzer) the limit line table.

Segments are sorted according to starting sweep time. A maximum of 20 segments can be defined in each of the upper and lower halves of a limit line.

Use LIMISEGT if you want to enter amplitude data in the upper or lower limit lines. Use LIMIHALF to specify entry into the upper limit line table or the lower limit line table (see line 30 of example).
LIMTEST Enable Limit Line Testing

Compares trace A with the current limit line data.

Syntax

```
Equivalent Key: Limit Test On Off
Preset State: LIMTEST OFF
Related Commands: LIMIFAIL, LIMISEG, LIMISEGT
```

Example

```
10 OUTPUT 718;"IP;SNGLS;CF300MHZ;SP100MHZ;"
Initializes spectrum analyzer and changes the frequency and span settings.

20 OUTPUT 718;"LIMIDEI;"
Deletes any limit line tables, sets the table type to fixed.

30 OUTPUT 718;"LIMIHLF UPPER;"
Specifies the upper limit line table.

35 OUTPUT 718;"LIMIFT FREQ;"
Selects a limit line based on frequency.

40 OUTPUT 718;"LIMISEG 250MHZ,-60DB,FLAT;"
```
Creates an entry to the upper limit line table.

50  OUTPUT 718; "LIMISEG 290MHZ,-60DB,SLOPE;"

60  OUTPUT 718; "LIMISEG 295MHZ,-15DB,FLAT;"

70  OUTPUT 718; "LIMISEG 305MHZ,-15DB,SLOPE;"

80  OUTPUT 718; "LIMISEG 310MHZ,-60DB,FLAT;"

90  OUTPUT 718; "LIMISEG 910MHZ,-60DB,FLAT;"

100 OUTPUT 718; "LIMITEST ON;TS;"

Turns on limit line testing.

110 OUTPUT 718; "LIMIFAIL?;"

Returns the status of the limit line testing.

120 ENTER 718; A

130 DISP A

Displays the result.

140 END

Description
A test is made of the data in TRA (trace A), and the result can be read, using LIMIFAIL, after each sweep.

Query Response
LN Linear Scale

Specifies the vertical graticule divisions as linear units, without changing the reference level.

Syntax

```
LN
```

Equivalent Key: Scale LogLin (when Lin is underlined)
Related Commands: LG, RL

Example

```
10 OUTPUT 718;'LN;'

Selects linear mode.

20 OUTPUT 718;'LN;RL 30MV;'
```

Description

The LN command scales the amplitude (vertical graticule divisions) proportionally to the input voltage, without changing the reference level. The bottom graticule line represents a signal level of zero volts.

Voltage entries are rounded to the nearest 0.1 dB. Thus, 30.16 mV becomes -17.4 dBm for a 50 Ω spectrum analyzer system.
LOAD Load

Specifies the file to be loaded into the analyzer.

**Syntax**

```
LOAD file \ character \ delimiter
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character</td>
<td>Any valid character</td>
<td>Any valid file name</td>
</tr>
<tr>
<td>Delimiter</td>
<td>Matching characters marking the beginning and end of the file name</td>
<td>!, @, ^, $, %, ;</td>
</tr>
</tbody>
</table>
Example

10 OUTPUT 718; "LOAD $MYTRA.TRC%, TRA;"

Loads MYTRA from the memory into trace A.

Description

The LOAD command is used to load files into the analyzer from the memory. There are several different types of file data. The type of data, and its corresponding data destination, are shown in the following table.

To use the LOAD command, you must specify the file name of the file to be loaded from the memory, including the appropriate file extension. If you are loading trace data, you must also specify the trace destination, TRA, TRB or TRC. Trace A will be assumed if you do not specify the destination.

Some of the characters that are available to use as delimiters are also allowed in the file name. Do not use any of the same characters for the delimiters as are used in the file name. The analyzer will interpret the second occurrence of the character as the end delimiter, rather than a part of the file name.

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Type of Data Transferred</th>
<th>File Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRA</td>
<td>Trace</td>
<td>.TRC</td>
</tr>
<tr>
<td>TRB</td>
<td>Trace</td>
<td>.TRC</td>
</tr>
<tr>
<td>TRC</td>
<td>Trace</td>
<td>.TRC</td>
</tr>
<tr>
<td>STATE</td>
<td>Instrument state</td>
<td>.STA</td>
</tr>
<tr>
<td>LIMILINE</td>
<td>Limit lines</td>
<td>.LIM</td>
</tr>
<tr>
<td>AMPCOR</td>
<td>Amplitude correction factors</td>
<td>.AMP</td>
</tr>
</tbody>
</table>

When recalling trace data, you need to specify either TRA, TRB, or TRC as the destination. You can omit the destination information when recalling states, limit line tables, or amplitude correction factors as long as the file extension is correctly specified.
LSPAN Last Span

Changes the spectrum analyzer’s span to the previous span setting.

Syntax

```
LSPAN
```

Equivalent Key: **Last Span**

Related Commands: SP

Example

```
10 OUTPUT 718;"LSPAN;"
```
MDS Measurement Data Size

Specifies measurement data size as byte or word when the trace data format mode is B, A, or I. See the TDF command for more information.

Syntax

![Syntax Diagram]

Related Commands: MKA, TDF, TRA
Preset State: W

Example

These commands transfer trace A in binary, 2 bytes per word.

10 INTEGER TRACE-A (1:401)
Declares variable, Trace-A.

20 OUTPUT 718;“SNGLS;TS;”
Activates single-sweep, updates trace A.

30 OUTPUT 718;“TDF B;MDS W;TRA?;”
Reads trace A in “word” format.

40 ENTER 718 USING ”#,401(W)”;TRACE_A()
Formats trace A output using data size of one word.

50 PRINT TRACE-A(*)
Prints trace A.

Description

The MDS command formats binary data in one of the following formats:

- B selects a data size of one 8-bit byte. When transferring trace data, MDS B transfers trace data the faster than MDS W
because only 401 bytes are transferred. Because MDS B combines two bytes into one byte, some resolution is lost.

W selects a data size of one word, which is two 8-bit bytes. When transferring trace data, MDS W transfers 802 bytes of trace data with no loss of resolution.

**How data is represented with MDS W:** When data is sent with MDS W, the trace data is converted into two bytes as follows:

1. The trace element’s amplitude (in measurement units) is divided by 256. The binary representation of the result is placed in the most significant byte (MSB).
2. The binary representation of the remainder is placed in the least significant byte (LSB).

For example, a trace element that is at the reference level has the value of 8000 (in measurement units). The result of 8000 divided by 256 is 30, with a remainder of 120. For this data, the contents of the MSB would contain the binary representation for 30.

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
\end{array}
\]

For this data, the contents of the LSB would contain the binary representation for 120.

\[
\begin{array}{cccccccc}
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
\end{array}
\]

**How data is represented with MDS B:** When data is sent with MDS B, the trace data is converted into one byte as follows:

- The trace element’s amplitude (in measurement units) is divided by 32. The binary representation of the result is placed into one byte.

For example, a trace element that is at the reference level has the value of 8000 (in measurement units). The result of 8000 divided by 32 is 250. For this data, the contents of the byte would contain the binary representation for 250.

\[
\begin{array}{cccccccc}
1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
\end{array}
\]

See “TDF” for information about using MDS for trace data transfers.
Query Response:

![Diagram showing query response]

QMDS
MDU Measurement Data Units

Returns values for the indicated display location in measurement units based on the current analyzer settings.

Syntax

Related Commands: TDF

Example

10 OUTPUT 718; "IP;TDF M;"
Initializes the spectrum analyzer and formats the trace data in measurement units.

20 OUTPUT 718; "RL -10DM;"
Changes the reference level to -10 dBm.

30 OUTPUT 718; "MDU RL_LOG?;"
Queries the value of the top graticule, the reference level.

40 ENTER 718; Reflevel
Moves the spectrum analyzer response to the computer.

30 OUTPUT 718; "MDU LOG-RES?;"
Programming Commands

Programming Command Descriptions

Queries the value of one measurement unit.

40 ENTER 718;Res
Moves the spectrum analyzer response to the computer.

30 OUTPUT 718;"MDU M80?;"
Queries the value at 80 dB down from the top graticule.

40 ENTER 718;Minus80
Moves the spectrum analyzer response to the computer.

50 PRINT Reflevel,Res,Minus80
Displays the results on the computer screen.

60 END

The example returns the following to the controller for a logarithmic analyzer display with 10 dB/division: 8000, 0.01, 0. The first number indicates the value of the top graticule. The second number received indicates that 0.01 dB is the resolution of a measurement unit, so a value of 7999 is 0.01 dB below the reference level or -10.01 dB. The last value is the measurement units of a position 80 dB below the top of screen.

Description

The MDU command returns information about the measurement units based on the current analyzer settings. Measurement units are an internal format for trace data related to display location. The range of valid numbers changes depending on whether the analyzer is in linear or logarithmic display mode. The value, in measurement units, of various locations on the display can be queried to determine the values for the current instrument settings. The values that can be queried are indicated in the following table.

<table>
<thead>
<tr>
<th>Secondary Keyword</th>
<th>Variable Returned</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL_LIN</td>
<td>The top-of-screen value (in linear mode)</td>
<td>8000</td>
</tr>
<tr>
<td>RI_LOG</td>
<td>The top-of-screen value (in log mode)</td>
<td>8000</td>
</tr>
<tr>
<td>LOG_RES</td>
<td>The resolution of one measurement unit (in log mode)</td>
<td>0.01</td>
</tr>
<tr>
<td>LIN_RES</td>
<td>The resolution of one measurement unit (in linear mode)</td>
<td>0.000125 = 1/8000</td>
</tr>
</tbody>
</table>

244
<table>
<thead>
<tr>
<th>Secondary Keyword</th>
<th>Variable Returned</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M80</td>
<td>The value 80 dB down from the reference level (in log mode)</td>
<td>0</td>
</tr>
<tr>
<td>M100</td>
<td>The value 100 dB down from the reference level (in log mode)</td>
<td>-2000</td>
</tr>
<tr>
<td>invalid keyword</td>
<td>A value returned if an invalid keyword is sent</td>
<td>-32768</td>
</tr>
</tbody>
</table>

**Query Response**

```
| number | output termination |
```

*Q31*
MEAN Trace Mean

Returns the mean value of the given trace in measurement units.

Syntax

![Diagram of MEAN command flow]

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace Range</td>
<td>A segment of trace A, trace B, trace C</td>
<td></td>
</tr>
</tbody>
</table>

Prerequisite Commands: TS when using trace data
Related Commands: MEANTH, RMS, STDEV, VARIANCE

Example

10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"SNGLS;"
Activates the single-sweep mode.

30 OUTPUT 718;"CF 300MHZ;SP 1MHZ;"
Sets measurement range.

40 OUTPUT 718;"TS;"
Sweeps trace A.

50 OUTPUT 718;"MEAN TRA?;"
Returns the mean value of trace A to the computer.

\[ \text{60 ENTER 718;} \text{Number} \]
Assigns value to computer variable, Number.

\[ \text{70 DISP "MEAN OF TRACE A IS ";Number} \]
Displays result on the computer screen.

\[ \text{80 END} \]

**Query Response**
MEANTH Trace Mean Above Threshold

Returns the mean value of the given trace above the threshold, in measurement units.

Syntax

```
MEANTH
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace Range</td>
<td>A segment of trace A, trace B, trace C</td>
<td></td>
</tr>
</tbody>
</table>

Prerequisite Commands: TS when using trace data.
Related Commands: MEAN, RMS, STDEV, TH, VARIANCE.

Example

```plaintext
10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"SNGLS;"
Activates the single-sweep mode.

30 OUTPUT 718;"CF 300MHZ;SP 1GHZ;"
Sets measurement range.

40 OUTPUT 718;"TH -40;"
Sets threshold level to -40 dB.
```
50 OUTPUT 718;"TS;"
Sweeps trace A.

60 OUTPUT 718;"MEANTH TRA?;"
Returns the mean value of trace A above the threshold to the computer.

70 ENTER 718;Number
Assigns value to computer variable, Number.

80 DISP "MEAN OF TRACE A ABOVE THE THRESHOLD IS ";Number
Displays result on the computer screen.

90 END

Description
MEANTH returns the mean value of the trace above the threshold; MEAN returns the mean value of the trace, regardless of the threshold level. MEANTH returns a "0" if there is not a signal above the threshold.

Use TH (threshold) to set the threshold level.

Query Response
MF Marker Frequency Output

Returns the frequency (or time) of the on-screen active marker.

Syntax

MF

Related Key: Marker Normal.
Related Commands: MKA, MKCF, MKD, MKF, MKN, MKPK, MKREAD.

Example

10 OUTPUT 718;“IP;SNGLS;”
Initializes the spectrum analyzer, activates single-sweep mode.

20 OUTPUT 718;“FA 280MHZ;FB 320MHZ;TS;”
Sets up the measurement range.

30 OUTPUT 718;“MKN;MKPK HI;”
Places marker on peak of a 300 MHz input signal.

40 OUTPUT 718;“MF;”
Takes frequency of marker.

50 ENTER 718;A
Returns frequency to the computer.

60 PRINT A
Prints the frequency on the computer screen.

Description

The MF command returns the frequency of the active marker to the controller if the marker is on screen. In delta marker mode, nonzero span, “MF;” returns the frequency difference between the two markers. In zero span, “MF;” returns the marker time, or the delta marker time.

The data that is returned by “MF;” depends on many command conditions including TDF, MKREAD, and MDS.
If the trace data format P is used with MF, the result is one real value in time units or frequency units, depending on the marker readout format. (See “MKREAD.”)

**Example 1**

10 OUTPUT 718; "TDF P; MKREAD FRQ; MF;"
This returns a frequency value (in Hz) if not in zero-span.

20 OUTPUT 718; "TDF P; MKREAD FRQ; MF;"
This returns a time value (in seconds) if in zero-span.

30 OUTPUT 718; "TDF P; MKREAD PER; MF;"
This returns the time value (in seconds) of 1/(marker frequency).

40 OUTPUT 718; "TDF P; MKREAD SWT; MF;"
This returns the marker time value (in seconds).

50 OUTPUT 718; "TDF P; MKREAD I ST; MF;"
This returns the frequency value (in Hz) for 1/(marker time).

If the trace data format is used with trace data format A, the result depends on the setting of the MDS command.

**Example 2**

10 OUTPUT 718; "TDF A; MDS B; MF;"
Returns one byte representing the marker position. The byte can assume values 0 to 100. Where 0 is the left edge screen position and 100 is the right edge screen position.

20 OUTPUT 718; "TDF A; MDS W; MF;"
Returns two bytes in a binary word format that has a value from 1 to 40 1.

If the trace data format is used with trace data format M, the result is the marker horizontal position value, from 1 to 401, in ASCII.

**Example 3**

10 OUTPUT 718; "TDF M; MF;"
Returns marker horizontal position value in ASCII.
MINH Minimum Hold

Updates trace C with the minimum level detected.

Syntax

```
MINH  CP  TRC  ;
```

Equivalent Key: Min Hold C.
Related Commands: BLANK, CLRW, MXMH, VAVG, VIEW.

Example

```
10 OUTPUT 718;"CLRW TRC;CONTS;"
```
Clears trace C and begin taking data.

```
20 OUTPUT 718;"MINH TRC;"
```
Updates trace C with the minimum level detected.

Description

MINH updates trace C with a new value from a detector only if the new value is smaller than the previous trace data value.
MKA Marker Amplitude

Returns the marker amplitude, independent of marker type.

**Syntax**

```
MKA ?
```

**Item** | **Description/Default** | **Range**
---|---|---
Number | Any real or integer number. Unit is current amplitude type. | Amplitude range of spectrum analyzer

Related Key: **Marker Normal**

**Example**

```
10 OUTPUT 718;"IP;"
```

Initializes the spectrum analyzer.

```
30 OUTPUT 718;"MKA -50;"
```

Places the marker at -50 dBm.

**Description**

If both the delta marker and active marker are on the screen, “MKA?;” returns the amplitude difference between the two markers. If the trace data format P (TDF P), is used with MKA, the result is one real value in the current amplitude units (AUNITS can be used to change the current amplitude units).

**Example**

```
10 OUTPUT 718;"TDF P;AUNITS DBM;MKA?;"
```

This returns the amplitude value of the marker (amplitude unit is dBm).
If the trace data format is used with trace data format A, the result depends on the setting of the MDS command.

**Example**

10 OUTPUT 718; "TDF A;MDS B;MKA?;"

Returns one byte representing the marker vertical position (-32,768 to 32,767) divided by 32 and then ANDed with 255.

20 OUTPUT 718; "TDF A;MDS W;MKA?;"

Returns two bytes in a binary word format that has a value from -32,768 to 32,767. The value represents the binary trace amplitude value.

Using the trace data format I is equivalent to the TDF A format. If the trace data format is used with trace data format M, the result is returned in ASCII measurement units (-32,768 to 32,767).

**Example**

10 OUTPUT 718; "TDF M;MKA?;"

Returns one vertical position value in measurement units.

**Query Response**

![Diagram](image)
MKACT Activate Marker

Specifies the active marker.

Syntax

```
MKACT
```

**Item** | **Description/Default** | **Range**
---|---|---
Number | Any valid integer. Default value is 1. | 1 to 4

Equivalent Key: **Select Marker 1 2 3 4**
Related Commands: MKA, MKF

**Example**

```
10 OUTPUT 718;"MKACT 4;"
```

Marker 4 becomes the active marker.

**Description**

There can be four different markers, but only one marker can be active at any time.

When this command is used, the following results occur:

- The marker number supplied by the command is made the active marker.
- If the marker number is not already on, the marker is turned on with preset type (position), and the marker is placed at center screen. The trace chosen is the first displayed trace found: trace A, trace B, trace C.
NOTE:

If the MKACT command is used to return to a marker that was already active, but was a different marker type, it replaces that marker with the new marker function. The amplitude and frequency for the previous marker are not saved.

Query Response

“MKACT?;” returns the marker number.
MKBW Marker Bandwidth

Returns the bandwidth at the specified power level relative to an on-screen marker (if present) or the signal peak (if no on-screen marker is present).

Syntax

```
MKBW
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any valid negative integer</td>
<td>0 to the amplitude of the noise floor</td>
</tr>
</tbody>
</table>

Equivalent Key: is similar to N dB Points On Off.

Example

```
10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"CF 300MHZ;SP 100MHZ;SNGLS;"
Changes the center frequency and span, then activates the single-sweep mode.

30 OUTPUT 718;"TS;MKPK HI;"
Updates the sweep, places marker on signal peak.

40 OUTPUT 718;"MKBW -3;"
Uses the MKBW function to find the signal bandwidth at -3 dB below the marker.
```
Description

The MKBW command also displays (in the message area) the bandwidth at the power level in $\text{dB}$ below the current marker position or the current signal peak.

MKBW finds the bandwidth at the specified power level for one measurement sweep. If you want the spectrum analyzer to find the bandwidth at the specified power level during every measurement sweep, use the NDBPNT command instead of MKBW.
MKCF Marker to Center Frequency

Sets the center frequency equal to the marker frequency and moves the marker to the center of the screen.

**Syntax**

```
MKCF
```

Equivalent Key: Mkr -> CF.
Related Commands: CF, MKF.

**Example**

This example provides a quick way to center the desired frequency on the spectrum analyzer screen.

```
10 OUTPUT 718;"IP;SP 1MHZ;SNGLS;"
   Initializes spectrum analyzer, activates single-sweep mode.
20 INPUT "ENTER IN DESIRED STATION FREQUENCY, IN MHZ",Freq
30 OUTPUT 718;"CF ";Freq;"MHZ;"
   Changes spectrum analyzer center frequency.
40 OUTPUT 718;"TS;MKPK HI;MKCF;TS;"
   Updates the trace, places marker at the signal peak and centers it on screen.
60 END
```

**Description**

This command is performed only if an active marker is present on screen.
MKD Marker Delta

Activates the delta marker.

**Syntax**

```
MKD number HZ ;
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is Hz, default value is value of the active marker.</td>
<td>Start frequency to stop frequency of spectrum analyzer</td>
</tr>
</tbody>
</table>

Equivalent Key: **Marker A**
Step Increment: by 1/10 of the frequency span
Related Commands: AUTO, MKCF, MKF, MKN, MKSP, MKSS, MKPK

**Example**

```
10 OUTPUT 718; "IP;"
```
Initializes spectrum analyzer.

```
20 OUTPUT 718; "MKMIN;"
```
Places a marker at the minimum amplitude of trace.
30 OUTPUT 718;"MKD;"
Activates marker delta.

40 OUTPUT 718;"MKPK HI;"
Places marker at highest amplitude of trace.

50 OUTPUT 718;"MKSP;"
Changes span to the values of the left and right markers.

END

Description

The **MKD** command computes the frequency and amplitude difference of the active marker and a special marker, called the delta or differential marker. These values are displayed in the display readout.

The differential value of the frequency is equal to the active marker frequency minus the delta marker frequency. The differential value of the amplitude is equal to the active marker amplitude minus the delta marker amplitude.

If an active marker is not on the screen, MKD positions an active marker at center screen. If a delta marker is not on the screen, MKD places one at the specified frequency, or at the current active marker. If the active marker is in amplitude mode, the delta marker is placed at the same amplitude as the active marker (or a specified value).

To read the amplitude, use MKA?. To read the frequency, use MKF? The results are displayed on the spectrum analyzer screen.

In linear mode, MKD computes the ratio of the amplitudes of the active and delta markers rather than the difference. This results in similar treatment for logarithmic and linear data because the delta (or, the difference) of two logarithmically generated numbers is the logarithm of the linear ratio of the two numbers. You should not change reference level when making a marker delta measurement, however.

If the marker delta function is on, the noise marker can be activated (MKN) and moved to measure the noise floor. The marker readout will then display the **signal-to-noise ratio**.
MKF Marker Frequency

Returns the frequency value of the active marker.

Syntax

```
MKF [frequency value]
```

Item Description/Default Range

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is Hz.</td>
<td>Start frequency to stop frequency of spectrum analyzer</td>
</tr>
</tbody>
</table>

Related Key: **Marker Normal**
Related Commands: AUTO, MKA, MKD, MKCF, MKPK

Example

```
10 OUTPUT 718; "MKF 600MHZ;"
```

Places an active marker at 600 MHz.

Description

In **nonzero** frequency spans, MKF returns the active marker frequency as a real number when MKF is queried. In zero span, “MKF;” returns the time value.

The data that is returned by MKF depends on many command conditions, including TDF, MKREAD, and MDS.
MKF results with TDF set to P: If the trace data format P is used with MKF, the result is one real value in time units or frequency units, depending on MKREAD.

Example

10 OUTPUT 718;"TDF P;MKREAD FRQ;MKF?;"
This returns a frequency value (in Hz) if not in zero-span.

20 OUTPUT 718;"TDF P;MKREAD FRQ;MKF?;"
This returns a time value (in seconds) if in zero-span.

30 OUTPUT 718;"TDF P;MKREAD PER;MKF?;"
This returns the time value (in seconds) of \( \frac{1}{\text{marker frequency}} \).

40 OUTPUT 718;"TDF P;MKREAD SWT;MKF?;"
This returns the marker time value (in seconds).

50 OUTPUT 718;"TDF P;MKREAD IST;MKF?;"
This returns the frequency value (in Hz) for \( \frac{1}{\text{marker time}} \).

MKF results with TDF set to A or I: If the trace data format is used with trace data format A, the result depends on the setting of the MDS command.

Example

10 OUTPUT 718;"TDF A;MDS B;MKF?;"
Returns one byte representing the marker position.

20 OUTPUT 718;"TDF A;MDS W;MKF?;"
Returns two bytes in a binary word format that has a value from 1 to 401. Using the trace data format I is equivalent to the TDF A format.

MKF results with TDF set to M: If the trace data format is used with trace data format M, the result is the marker horizontal position value, from 1 to 401, in ASCII.

Example

10 OUTPUT 718;"TDF M;MKF?;"
Returns marker horizontal position value in ASCII.
Query Response

number  output  termination
MKFA Marker to Start Frequency

Sets the analyzer start frequency to the marker frequency and moves the marker to the left edge of the screen.

Syntax

```
MKFA
```

Equivalent Key: Mkr -> Start.
Related Commands: FA, MKF.

Example

This example provides a quick way to change the start frequency on the spectrum analyzer screen.

```
10 OUTPUT 718;"IP;SP 1MHZ;SNGLS;"
   Initializes spectrum analyzer, activates single-sweep mode.
20 INPUT "ENTER IN DESIRED STATION FREQUENCY, IN MHZ",Freq
30 OUTPUT 718;"CF ";Freq;"MHZ;"
   Changes spectrum analyzer center frequency.
40 OUTPUT 718;"TS;MKPK HI;MKFA;TS;"
   Updates the trace, places marker at the signal peak and changes the displayed start frequency to the signal frequency.
```

Description

This command is performed only if an active marker is present on screen.
MKFB Marker to Stop Frequency

Sets the analyzer stop frequency to the marker frequency and moves the marker to the right edge of the screen.

Syntax

```
MKFB
XMKFB
```

Equivalent Key: **Mkr -> Stop.**
Related Commands: FB, MKF.

Example

This example provides a quick way to change the stop frequency on the spectrum analyzer screen.

```
10 OUTPUT 718;"IP;SP 1MHZ;SNGLS;"
Initializes spectrum analyzer, activates single-sweep mode.

20 INPUT "ENTER IN DESIRED STATION FREQUENCY, IN MHZ",Freq

30 OUTPUT 718;"CF ";Freq;"MHZ;"
Changes spectrum analyzer center frequency.

40 OUTPUT 718;"TS;MKPK HI;MKFB;TS;"
Updates the trace, places marker at the signal peak and changes the displayed stop frequency to the signal frequency.

60 END
```

Description

This command is performed only if an active marker is present on screen.
MKFC Marker Counter

Turns on or off the marker frequency counter.

Syntax

Equivalent Key:Freq Count or Marker Count On Off.
Related Commands: MKFCR, MKN.

Example

10 OUTPUT 718;"MKFC ON;"

Turns on the marker counter.
MKFCR Marker Counter Resolution

Sets the resolution of the marker frequency counter.

**Syntax**

```
MKFCR
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is Hz.</td>
<td>0 Hz to 100 kHz</td>
</tr>
</tbody>
</table>

Equivalent Key: **Resolution Auto Man.**

Preset State: Marker counter resolution is set to AUTO. The calculated value for the marker counter resolution is returned if the MKFCR is queried.

Related Commands: AUTO, MKFC.

**Example**

```bash
10 OUTPUT 718;"MKFCR 10KHZ;"
```

Sets the marker counter resolution to 10 kHz.
20 OUTPUT 718;"MKFCR?;"
Queries the marker counter resolution.

30 ENTER 718;A
Gets the query response.

40 DISP A
Displays the marker counter resolution.

Description
Executing either “MKFCR 0;” or “MKFCR AUTO;” auto-couples the marker counter resolution to the frequency span.

Query Response

![Diagram](attachment:image.jpg)
MKMIN Marker Minimum

Moves the active marker to the minimum value detected.

Syntax

```
MKMIN
```

Related Commands: MKPK, SMOOTH, MKTH, VAVG.

Example

```
10 OUTPUT 718;"IP;SNGLS;"
Initializes spectrum analyzer, activates single-sweep mode.

20 INPUT "ENTER IN THE START FREQUENCY, IN MHZ",Start_freq

30 INPUT "ENTER IN THE STOP FREQUENCY, IN MHZ",Stop_freq

40 OUTPUT 718;"FA ";Start_freq;"MHZ"
Sets the start frequency.

50 OUTPUT 718;"FB ";Stop_freq;"MHZ"
Sets the stop frequency.

60 OUTPUT 718;"TS;MKPK HI;MKD;MKMIN;MKF?;"
Updates trace, finds trace peak, turns on marker delta function, finds the
minimum value of trace, and return the frequency delta.

70 ENTER 718;Delta_freq
Gets the result from spectrum analyzer.

80 PRINT "DIFFERENCE IN FREQUENCY IS ",Delta_freq,"HZ"

90 END
```
MKN Marker Normal

Activates and moves the marker to the specified frequency.

**Syntax**

![Diagram of MKN marker normal]

**Item** | **Description/Default** | **Range**
---|---|---
Number | Any real or integer number. Default unit is Hz. Default value is the center frequency of the spectrum analyzer. | Start frequency to stop frequency of spectrum analyzer

Equivalent Key: **Marker Normal**.
Step Increment: by 1/10 of the frequency span.
Related Commands: AUTO, DEMOD, MKA, MKD, MKF, MKPK.

**Example**

10 INPUT "ENTER IN THE START FREQUENCY, IN MHZ", Start_freq
20 INPUT "ENTER IN THE STOP FREQUENCY, IN MHZ", Stop_freq
30 OUTPUT 718;"IP;FA ";Start_freq;"MHZ"
Initializes spectrum analyzer and changes the start frequency.

40 OUTPUT 718; "FB "; Stop_freq; "MHZ"
Changes the stop frequency.

50 OUTPUT 718; "MKN EK;"
Enables the front-panel knob.

60 PRINT "PLACE MARKER ON THE DESIRED SIGNAL"

70 PRINT "PRESS HOLD THEN PRESS CONTINUE"

80 PAUSE

90 OUTPUT 718; "MKN?;"
Gets the frequency of the marker.

100 ENTER 718; Mkr
Puts the frequency value into the computer variable, Mkr.

110 PRINT "MARKER FREQUENCY IS ", Mkr, "Hz"
Prints the result.

120 END

Description
In nonzero span, "MKN?;" returns the frequency value. In zero span, "MKN?;" returns the time value.

Query Response
MKNOISE Marker Noise

Displays the average noise level at the marker.

Syntax

```
MKNOISE
```

Equivalent Key: Marker Noise On Off.
Related Commands: MKA, MKF, MKMIN, MKN.

Example

This programming example measures carrier to noise ratio, using the noise marker.

```
10  ASSIGN @Sa TO 718
Sets the spectrum analyzer HP-IB address to 718.

20  OUTPUT @Sa;"IP;TS;RFCALIB ON;"
Initializes the analyzer and turns on the internal 50 MHz source.

30  OUTPUT @Sa;"CF 50 MZ;SP 10 MZ;SNGLS;TS;"
Sets up the measurement conditions.

40  OUTPUT @Sa;"MKPK HI;TS;"
Puts the marker on the highest peak.
50 OUTPUT @Sa;"MKD UP;UP;MKNOISE ON;TS;MKA?;"
Moves the marker off the peak and measure noise.

60 ENTER @Sa;C_to_n
Puts the marker value into a variable.

70 OUTPUT @Sa;"MKNOISE OFF;"
Turns off the noise marker.

80 PRINT "Carrier to Noise Ratio in 1 Hz BW is ";C_to_n;"dB"

90 END

Description

The marker value is normalized to a 1 Hz bandwidth. Use “MKA?;” to read the noise marker value. If the marker delta function is on and the noise marker is activated and moved to measure the noise floor, the marker readout will display the signal-to-noise ratio.

The noise marker averages 32 trace data values about the location of the marker on the frequency or time scale. The 32 values averaged, except for the first 15 or last 14 values in the trace, commence with the 16th point to the left of the marker, and end with the 15th point to the right of the marker. Note that the data values averaged are not exactly symmetrical with respect to marker position. At the trace end points, the spectrum analyzer uses the nearest 32 data values. So while the marker may be moved to trace position 1 to 15, the actual amplitude value returned will be the same value for any marker position from 1 to 15. A similar situation applies for markers at the end of the trace.

A nominal correction for equivalent noise bandwidths is made by the firmware based on a nominal 3 dB resolution bandwidth. The firmware assumes the noise bandwidth is 1.12 times the resolution bandwidth. This means the shape of the resolution bandwidth filters cause the noise power to be overstated by 1.12 times. The detection mode also affects the measurement. If in log mode, the log detector understates the noise response. To compensate, 2.5 dB is added to the measurement. If the detector is in linear mode, the firmware uses 1.05 dB as a correction value.

In log detector mode, the final reported value will then be, with the result reported in dBm in a 1 Hz bandwidth:

\[(\text{Averaged value over 32 values}) - 10 \times \log(1.12 \times \text{Resolution bandwidth}) + 2.5 \text{ dB}\]
In linear detector mode (dBm) units, the final reported value will then be, with the result reported in dBm in a 1-Hz bandwidth:

\[
(Averaged\ value\ over\ 32\ values) = 10 \times (\log_{10}(1.12 \times \text{Resolution\ bandwidth})) + 1.05\ \text{dB}
\]

In linear detector mode with the normal display of voltage units, the noise marker voltage value will be related to the present marker voltage by this relation.

\[
(V_{\text{-noise-marker}}) = (V_{\text{-average}}) \times 1.12 \times \text{Resolution\ bandwidth} \times 0.7824
\]

\[
V_{\text{-noise-marker}} = \frac{V_{\text{-average}}}{(1.12 \times \text{Resolution\ bandwidth} \times 0.7824)^{0.5}}
\]

\[
V_{\text{-noise-marker}} = V_{\text{-average}} \times \frac{1.06633}{(\text{Resolution\ bandwidth})^{0.5}}
\]

**Query Response**

![Query Response Diagram](image_url)

002
MKOFF Marker Off

Turns off either the active marker or all the markers.

**Syntax**

```
MKOFF SP ALL ;
```

Equivalent Key: **Marker All Off.**
Related Commands: MKA, MKACT, MKACTV, MKCF, MKD, MKF, MKN, MKPK.

**Example**

```
10 OUTPUT 718; "MKOFF ALL;"
```

Turns off all the on-screen markers.

**Description**

If the ALL parameter is omitted, only the active marker is turned off.
MKP Marker Position

Places the active marker at the given x-coordinate.

Syntax

\[
\text{MKP} \ x \ \text{coordinate} \ ;
\]

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any valid integer</td>
<td>-401 to 401</td>
</tr>
<tr>
<td>Trace element</td>
<td>An element of trace A, trace B, trace C</td>
<td></td>
</tr>
</tbody>
</table>

Related Commands: MKA, MKCF, MKD, MKMIN.

Example

10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"MKP 100;"
Moves the active marker to an element 100 of trace A.

Description

If no marker is active, the marker is turned on with preset type (position) and marker is placed at the given screen position. The marker is placed on the first displayed trace that is found (in order): trace A, trace B, or trace C.
If the marker delta mode is active, the value of the marker position is relative to the fixed marker, and therefore MKP can return a negative position.

Note that MKP and MKCF commands perform different functions. MKCF sets the center frequency equal to the marker frequency and moves the marker to the center of the screen. MKP places the marker to the position of the element specified.

**Query Response**

![Diagram of Query Response]
MKPAUSE Marker Pause

Pauses the sweep at the active marker for the duration of the delay period.

Syntax

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number</td>
<td>2 ms to 100 s</td>
</tr>
</tbody>
</table>

Restrictions: Not available with negative peak detection.
Equivalent Key: is similar to **Dwell Time On Off**.
Step Increment: 1, 2, 5, 10 sequence
Related Commands: DEMOD, DEMODT, MKA, **MKF**, MKFC, MKN, MKOFF, ST.

**Example**

10 OUTPUT 718; “MKPAUSE 10SC;”
Changes the marker pause time to 10 seconds.

**Description**

To turn MKPAUSE off send “MKPAUSE 0;”.
Query Response

number → output termination

Q01
MKPK Marker Peak

Positions the active marker on a signal peak.

Syntax

```
MKPK
```

Related Keys: Search, Next Peak, Next Pk Right, Next Pk Left, and Peak Search.
Related Commands: MKCF, MKF, MKOFF, MKPX, MKTH.

Example

```
10 OUTPUT 718;"IP;"      
Initializes the spectrum analyzer.
20 OUTPUT 718;"SNGLS;TS;MKPK HI;" 
Places active marker on highest peak.
30 OUTPUT 718;"MKA?;"    
Returns amplitude value of marker to the computer.
40 ENTER 718;A           
Puts the spectrum analyzer response in the computer variable, A.
50 DISP A                
Displays amplitude value.
60 END                   
```

Description

Executing MKPK HI, or simply MKPK, positions the active marker at the highest signal detected. If an active marker is on the screen, the MKPK parameters move the marker as follows:
HI (highest) moves the active marker to the highest peak.

NH (next highest) moves the active marker to the next signal peak of lower amplitude.

NR (next right) moves the active marker to the next signal peak of higher frequency.

NL (next left) moves the active marker to the next signal peak of lower frequency.

**NOTE:** This function is for use with the frequency markers only.
MKPP Marker Peak-to-Peak

Positions markers on the highest and lowest points on the trace and displays the frequency and amplitude differences.

Syntax

```
MKPP
```

Related Keys: Search, Peak Search.
Related Commands: MKPK, MKF, MKOFF, MKPX, MKTH.

Example

```
10 OUTPUT 718;"IP;"
Initializes the spectrum analyzer.

20 OUTPUT 718;"SNGLS;TS;MKPP;"
Places markers on highest peak and lowest point.

30 OUTPUT 718;"MKA?;"
Returns amplitude value of delta marker to the computer.

40 ENTER 718;A
Puts the spectrum analyzer response in the computer variable, A.

50 DISP A
Displays amplitude value.

60 END
```
MKPX Marker Peak Excursion

Specifies the minimum signal excursion for the spectrum analyzer’s internal peak identification routine.

Syntax

```
MKPX
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is dB.</td>
<td>Oto 100 dB</td>
</tr>
</tbody>
</table>

Equivalent Key: **Peak Excursn.**

Preset State: 6 dB.

Step Increment: by 1 dB.

Related Commands: MKPK, PEAKS.

Example

10 OUTPUT 718; "IP;CF 300MHZ;SP 1GHZ;"
Initializes spectrum analyzer, changes start and stop frequencies.

20 INPUT "ENTER IN PEAK EXCURSION, IN DB ", Excursion

30 OUTPUT 718; "MKPX "; Excursion; "DB;"
Changes peak excursion level.

40 OUTPUT 718; "TS;MKPK HI;MKPK NH;"
Searches for highest peaks of trace.
50 OUTPUT 718;"MKF?;"
Finds frequency difference between peaks.

60 ENTER 718;Freq
Puts the spectrum analyzer response in the computer variable, Freq.

70 IF Freq <> 0 THEN
Outputs results if marker amplitude was not 0.

80 PRINT "PEAK FOUND"

90 ELSE
Prints “NO PEAKS FOUND” if Freq = 0.

100 PRINT "NO PEAKS FOUND"
110 END IF
120 END

Query Response
MKREAD Marker Readout

Selects the type of active trace information displayed by the spectrum analyzer marker readout.

**Syntax**

```
(MKREAD)
```

Equivalent Key: Mkr Readout provides the marker readouts in the frequency, sweep time, inverse sweep time, and period. The fast Fourier transform readout is not available with the softkey, however.

Related Commands: MKF, MKTYPE.

**Example**

```
10 OUTPUT 718; "TDF P;MKREAD FRQ;MF;"
```

This returns a frequency value (in Hz) if not in zero-span.

**Description**

The MKREAD command can select the following types of active trace information:

- **FRQ**: frequency
- **SWT**: sweep time
- **IST**: inverse sweep time
- **PER**
PER  period

The results of the data depend on the MKREAD parameter, the frequency span, and if the marker delta function is used.

<table>
<thead>
<tr>
<th>MKREAD Type</th>
<th>Non-Zero Span</th>
<th>Non-Zero Span Delta</th>
<th>Zero Span</th>
<th>Zero Span Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRQ</td>
<td>Reads frequency</td>
<td>Reads delta frequency</td>
<td>Reads time</td>
<td>Reads delta time</td>
</tr>
<tr>
<td>SWT</td>
<td>Reads time since the start of sweep</td>
<td>Reads delta time between end points</td>
<td>Waveform measurements of detected modulation</td>
<td>Waveform measurements of detected modulation</td>
</tr>
<tr>
<td>IST</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Computes frequency corresponding to delta of markers. Performs ( \frac{l}{(T_1 - T_2)} )</td>
</tr>
<tr>
<td>PER</td>
<td>Period of frequency</td>
<td>(Pulse measurement) delta time</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Query Response
MKRL Marker to Reference Level

Sets the reference level to the amplitude value of the active marker.

Syntax

```
MKRL
```

Equivalent Key: `Mkr -> RL`
Related Commands: `MKOFF, RL`

Example

10 OUTPUT 718;"IP;SNGLS;CF 300MHZ;SP 20MHZ;"
Initializes spectrum analyzer, changes center frequency and span.

20 OUTPUT 718;"TS;MKPK HI;MKRL;TS;"
Places a marker on trace peak, sets the reference level to the amplitude of the active marker, updates the sweep.

30 OUTPUT 718;"RL?;"
Gets the reference level.

40 ENTER 718 USING "K";Ref_level
Puts the spectrum analyzer response in the computer variable, `Ref_level`.

50 OUTPUT 718;"AUNITS?;"
Gets the current amplitude units.

60 ENTER 718;Aunits$

50 PRINT "REFERENCE LEVEL IS",Ref_level,Aunits$

60 END
MKSP Marker to Span

Sets the start and stop frequencies to the values of the delta markers.

Syntax

```
MKSP
```

Equivalent Key: M kr A -> Span.

Related Commands: MKD, SP.

Example

```plaintext
10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"MKMIN;"
Places a marker at the minimum amplitude of trace.

30 OUTPUT 718;"MKD;"
Activates marker delta.

40 OUTPUT 718;"MKPK HI;"
Places marker at highest amplitude of trace.

50 OUTPUT 718;"MKSP;"
Changes span to the values of the left and right markers.

60 END
```

Description

The left marker specifies start frequency, and the right marker specifies stop frequency. If MKD is off, no operation is performed.
MKSS Marker to Step Size

Sets the center-frequency step-size to the marker frequency.

Syntax

```
MKSS
```

Equivalent Key: Mkr -> CF Step.

Related Commands: CF, MKA, MKCF, MKD, MKF, SS.

Example

This example uses a 300 MHz input signal with harmonics.

```
20 OUTPUT 718;“IP;SNGLS;CF 300MHZ;SP 20MHZ;TS;”
```

Initializes spectrum analyzer, activates single-sweep mode, changes center frequency and span, updates trace.

```
30 OUTPUT 718;“MKPK HI;MKSS;MKD;CF UP;TS;MKPK HI;”
```

Places the marker on the highest point of the trace, changes the step size to the marker frequency, activates marker delta, increase center frequency, update trace, places the marker at highest point of the trace.

```
40 OUTPUT 718;“MKA?;”
```

Gets the amplitude of the marker.

```
50 ENTER 718;Delta_amp
```

Puts the spectrum analyzer response in the computer variable, Delta-Amp.

```
60 OUTPUT 718;“MKF?;”
```

Gets the frequency of the marker.

```
70 ENTER 718;Delta_freq
```

Puts the spectrum analyzer response in the computer variable, Delta-freq.

```
80 PRINT “DIFFERENCE IN AMPLITUDE IS “,Delta_amp,“dB”
90 PRINT “DIFFERENCE IN FREQUENCY IS “,Delta_freq,“Hz”
```
100 END

**Description**

Sets the center-frequency step-size equal to the marker frequency. If in the delta mode, the step size is set to the delta frequency (absolute value).
MKTBL Marker Table

Turns on or off the marker table.

Syntax

Equivalent Softkey: Mk Table On Off
Preset State: 0 (marker table is off)
Related Commands: DL, PEAKS

Example

```
OUTPUT 718; MOV MKTBL, 1;
```

Turns on the marker table.

Description

When the marker table is turned on, the spectrum analyzer screen displays two windows. The upper window displays the traces and the graticule, and the lower window displays the marker table. The marker table displays the following information about the on-screen markers: the trace (trace A, B, or C) on which the marker is located, the type of marker (frequency, time, inverse sweep time, or period), the frequency or time of the marker, and the amplitude of the marker. While the marker table is turned on, the marker table data is updated at the end of every sweep, or whenever a marker is moved. (MKTBL command uses the ONMKRU command to update the marker table information).

The marker table is displayed on the spectrum analyzer display only. To obtain the information that is displayed in the marker table remotely, you must use the following programming commands.

- Use MKACT to select a marker. Use the MKACTV command makes the selected marker the active function.
- Use MKA? to determine the amplitude of a marker.
- Use MKF? to determine the frequency or time of a marker.
- Use MKREAD? to determine the type of marker.
- Use MKTRACE? to determine which trace the marker is located on.

**Restrictions**
Turning on the marker table turns off following functions: windows display mode (WINON), N dB point measurement (NDBPNT), the FFT menu measurements (FFTAUTO, FFTCONTS, FFTSNGLS), gate utility functions (GDRVUTIL), TOI measurement (TOI), peak table (PKTBL), percent AM (PCTAM), peak zoom (PKZOOM), and power menu measurements (ACP, ACPE, CHP, and OBW). Marker noise (MKNOISE) and marker counter (MKFC) are not available with the marker table.

You can execute the MKTBL command two different ways. You can either execute the MKTBL command directly (for example, "MKTBL 1;") or use the MOV command to move the 1 or 0 into the MKTBL command (for example, "MOV MKTBL, 1; "). If you use the MOV command, no text is displayed in the active function area during command execution.

You should turn off the marker table (set MKTBL to 0) when you are done with the marker table.

**Query Response**

![Diagram of Query Response]
MKTH Peak Threshold

Sets a lower boundary to the trace and “clips” signals that appear below the threshold level.

**Syntax**

```
MKTH [ ? | AUTO | DN | UP ] number [ DB ];
```

**Item** | **Description/Default** | **Range**
--- | --- | ---
Number | Any real or integer number. Default unit is dBm. | Range dependent on RL setting

Equivalent Key: **Pk Threshold**
Preset State: -90 dBm
Step Increment: One division
Related Commands: MKPX, MKPK, PEAKS

**Example**

```
10 OUTPUT 718;"MKTH UP;"
```

Increases the threshold level.

**Description**

The MKTH command sets a lower boundary to the active trace. The threshold line “clips” signals that appear below the line when this function is on. The boundary is defined in amplitude units that correspond to its vertical position when com-
pared to the reference level. In other words, if the reference level is -10 dBm, and
the threshold is set to -75 dBm, the threshold will remain 65 dB below the refer-
ence level as the reference level is changed.

The value of the threshold appears in the active function block and on the lower
left side of the display. The threshold level does not influence the trace memory or
marker position. The peaks found by the markers must be at least the peak excur-
sion value above the threshold level. Use the MKPX command to change the
value of peak excursion. The threshold value affects peak searching functions,
except for MKPK.

Query Response
MKTRACE Marker Trace

Moves the active marker to a corresponding position in trace A, trace B, or trace C.

**Syntax**

```
MKTRACE TRA
```

Equivalent Key: Marker Trace Auto A B C.

**Example**

10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"MKMIN;"
Finds the lowest amplitude of trace.

30 OUTPUT 718;"MKTRACE TRB;"
Moves marker to corresponding position on trace B.

40 OUTPUT 718;"BLANK TRA;CLRW TRB;"
Blanks trace A and displays trace B.

50 END

**Query Response**
MKTRACK Marker Track

Moves the signal on which the active marker is located, to the center of the spectrum analyzer display and keeps the signal peak at center screen.

Syntax

```
MKTRACK
```

Equivalent Key: **Signal Track On Off**.
Related Commands: MKA, MKCF, MKF.

Example

```
10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"CF 300MHZ;TS;"
Changes the center frequency.

30 OUTPUT 718;"MKTRACK ON;"
Activates the marker track.

40 OUTPUT 718;"SP 10MHZ;TS;"
Changes the span.

50 OUTPUT 718;"MKTRACK OFF;"
```
Turns off the marker track.

**Description**
To keep a drifting signal at center screen, place the active marker on the desired signal before turning on MKTRACK.

**Query Response**

```
ON termination
OFF
```
ML Mixer Level

Specifies the maximum signal level that is applied to the input mixer for a signal that is equal to or below the reference level.

Syntax

```
ML/amplitude value
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is dBm.</td>
<td>-10 to -60 dBm</td>
</tr>
</tbody>
</table>

Equivalent Key: Max Mixer Lvl.
Preset State: -10 dBm.
Step Increment: 10 dB minimum.
Related Commands: AT, ROFFSET.

Example

```
10 OUTPUT 718; "ML -40DM;"
```

As the reference level is changed, the coupled input attenuator is changed automatically. This limits the maximum signal at the mixer input to -40 dBm for signals less than or equal to the reference level.

Description

The ML command specifies the maximum signal level that is applied to the input mixer for a signal that is equal to or below the reference level.
The effective mixer level is equal to the reference level minus the input attenuator setting. When ML is activated, the effective mixer level can be set from -10 dBm to -60 dBm in 1 dB steps.

**Query Response**

- number
- output termination

001
MOV Move

Covers the source values into the destination.

Syntax

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace range</td>
<td>A segment of trace A, trace B, trace C, or a user-defined trace.</td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>Any real or integer number.</td>
<td>Real number range.</td>
</tr>
</tbody>
</table>
Example

10   INTEGER Tra_binary(1:401)
Declare an array for the trace data.

20   ASSIGN @Sa TO 718;FORMAT OFF

30   OUTPUT @Sa;"IP;TS;RFACALIB ON;"
Turn on the internal 50 MHz signal.

40   OUTPUT @Sa;"CF 50 MZ;SP 10 MZ;SNGLS;TS;"
Set up the analyzer and take a sweep.

50   OUTPUT @Sa;"MDS W;TDF B;TRA?;"
Output the trace A data.

60   ENTER @Sa;Tra_binary(*)

70   OUTPUT @Sa;"TDF A;"
Change the trace data to A-block format.

80   OUTPUT @Sa;"MOV TRA,0;"
Set the trace A data to 0's.

90   DISP "Press CONTINUE when ready"

100  PAUSE

110  DISP

120  OUTPUT @Sa USING ",,K,W","TRA#A",802
Prepare the spectrum analyzer for the data.

130  OUTPUT @Sa;Tra_binary(*)
Transfer the data to the analyzer.

140  OUTPUT @Sa;"VIEW TRA;"

150  LOCAL @Sa

160  END

302
Description

When the source is longer than the destination, the source is truncated to fit. When the source is shorter than the destination, the last element is repeated to fill the destination.
MXMH Maximum Hold

Updates each trace element with the maximum level detected.

Syntax

```
MXMH TRA
```

Equivalent Keys: Max Hold A and Max Hold B.

Related Commands: BLANK, CLRW, MINH, VAVG, VIEW.

Example

```
OUTPUT 718; "MXMH TRA;"
```

Description

MXMH updates the specified trace (either trace A or trace B) with a new value from a detector only if the new value is larger than the previous trace data value.
NDB Number of dB

Specifies the distance (in dB) from the signal peak for the N dB points measurement (NDBPNT).

Syntax

```
NDB from Peak --,--
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any negative real or negative integer number. Default unit is dB.</td>
<td>-1 to -80 dB</td>
</tr>
</tbody>
</table>

Equivalent Key: **N dB Points On Off**

Preset State: -3 dB.

Related Commands: MKBW, MKPX, NDB, NDBPNT, NDBPNTR, MKTH.

Example

10 OUTPUT 718; "NDBPNT 1;"
Turns on the N dB points measurement.

20 OUTPUT 718; "NDB -6DB;"
Sets the N dB points measurement to measure 6 dB below the signal’s peak.

30 OUTPUT 718; "NDBPNTR?;"
Queries NDBPNTR. NDBPNTR contains the measurement results of the N dB points measurement.

40 ENTER 718; Six
Stores the result in the variable Six.
Description
When the N dB points function is turned on, the spectrum analyzer finds the bandwidth, at the number of dB down specified by the NDB command, of the highest on-screen signal. The highest on-screen signal must be greater than the peak excursion above the current threshold, but the N dB points may fall below the threshold.

Query Response
NDBPNT N dB Points

Turns on or off the N dB points function.

Syntax

```
NDBPNT j- off +@-+ ( 0 /- ) On 1 NDBPNT
```
Equivalent Key: N dB Points On Off.
Related Commands: MKBW, MKPX, NDB, NDBPNTR, MKTH.

Example

This example finds the shape factor of a signal.

```
10 OUTPUT 718;"NDBPNT 1;"
Turns on the N dB points measurement.

20 OUTPUT 718;"NDB -6DB;"
Sets the N dB points measurement to measure 6 dB below the signal’s peak.

30 OUTPUT 718;"NDBPNTR?;"
Queries NDBPNTR. NDBPNTR contains the measurement results of the N dB points measurement.

40 ENTER 718;Six
Stores the result in the variable Six.

50 OUTPUT 718;"NDB -60DB;"
Sets the N dB points measurement to measure 60 dB below the signal’s
```
Programming Commands
Programming Command Descriptions

peak.

60 OUTPUT 718;"NDBPNTR?;"
Queries NDBPNTR.

70 ENTER 718;Sixty
Stores the result in the variable Sixty.

80 IF Six <> -100 AND Sixty <> -100 THEN
If both the measurement at -6 dB and -60 dB were valid, print the shape factor of the signal.

90 PRINT "Shape factor is ",Sixty/Six
Prints the shape factor of the signal.

100 ELSE
If the bandwidth at -6 dB or -60 dB could not be found, an error statement is printed.

110 PRINT "Error, bandwidth could not be determined"

120 END IF

130 OUTPUT 718;"NDBPNT 0;"
Turns off the N dB points measurement.

Description
Setting NDBPNT to 1 turns on the N dB points measurement. Setting NDBPNT to 0 turns off the N dB points measurement. When the NdB points function is turned on, the spectrum analyzer finds the bandwidth, at the number of dB down specified by the NDB command, of the highest on-screen signal. The N dB points measurement is repeated at the end of every sweep, to update the measurement data, until you turn it off. To determine the bandwidth measured by NDBPNT, you must query NDBPNTR.

To be able to measure a signal with NDBPNT, there must be an on-screen signal that is greater than the peak excursion (see “MKPX” for more information about the peak excursion) above the threshold, and there cannot be any other signals with amplitudes within N dB of the peak of the highest signal. If a signal cannot be found or there is more than one signal within the value of NDB of the highest signal, the value of NDBPNTR will be -100.
You can execute the NDBPNT function by using the command, "**NDBPNT 1;**".

**Restrictions** Turning on the NDBPNT function turns off the following functions: TOI measurement (TOI), percent AM (PCTAM), and span zoom (SPZOOM).

You should turn off the N dB points measurement (set NDBPNT to 0) when you are done with it.

**Query Response**

![Diagram showing the query response for NDBPNT](image)
NDBPNTR N dB Points Bandwidth

Returns the bandwidth measured by the N dB points measurement (NDBPT).

Syntax

```
NDBPNTR
```

Equivalent Key: **N dB Points On Off**

Related Commands: MKPX, NDB, NDBPNT, MKTH.

Example

```
10 OUTPUT 718;"NDBPNT 1;"
```

Turns on the N dB points measurement.

```
20 OUTPUT 718;"NDB -6DB;"
```

Sets the N dB points measurement to measure 6 dB below the signal’s peak.

```
30 OUTPUT 718;"NDBPNTR?;"
```

Queries NDBPNTR. NDBPNTR contains the measurement results of the N dB points measurement.

```
40 ENTER 718;Six
```

Stores the result in the variable Six.

Description

NDBPNTR returns a -100 if the NDBPNT function has not been turned on, or if NDBPNT did not find a signal to measure. (For NDBPNT to be able to measure a signal, there must be an on-screen signal that is greater than the peak excursion above the threshold, and there cannot be any other signals with amplitudes that are within N dB of the peak of the highest signal.)

Query Response

```
bandwidth
in Hz

number

output
termination

NDBPNTR
```
NORMLIZE Normalize Trace Data

Syntax

Equivalent Key: Normalize
Preset State: OFF
Restrictions: Will not work with linear display mode
Related Commands: NRPOS, NRL, RL

Example

10    OUTPUT 718;"IP;SNGLS;"
20    OUTPUT 718;"FA 300KHZ;FB 1GHZ;"
30    OUTPUT 718;"SRCPWR ON;"
40    OUTPUT 718;"SWPCPL SR;"
50    OUTPUT 718;"RB 100KHZ;"
60    PRINT "CONNECT THRU. PRESS CONTINUE WHEN READY TO STORE."
70    PAUSE
80    OUTPUT 718;"TS;DONE?;"
90    ENTER 718;Done
100   OUTPUT 718;"NORMLIZE ON;"
110   OUTPUT 718;"TS;DONE?;"
120   ENTER 718;Done
130   LOCAL 718
140   END
Description

When the NORMLIZE command is first turned on trace A is copied to trace B so that trace B can be used as the reference trace. Each sweep $A - B + \text{top of screen}$ (the normalized reference level) is calculated and the results are displayed in trace A. Any changes to the absolute reference level (RL command) are reflected in the normalized reference level (NRL command) and in the reference trace data.

When the normalize function is on, the reference level annotation on screen is changed to display the relative reference level, which is effectively the normalized reference level. However, a remote query with the RL command will still return the absolute reference level.

Query Response

![Diagram](attachment://Diagram.png)
NRL Normalized Reference Level

Sets the normalized reference level.

Syntax

```
NRL
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is dBm.</td>
<td></td>
</tr>
</tbody>
</table>

Equivalent Key: **Norm Ref Lvl**

Preset State: **0 dB**

Related Commands: NORMLIZE, RL, NRPOS

Example

```
150 OUTPUT 718; "NRL 40DB;"
```

Sets the normalized reference level to 40 dBm.

Description

This function is a trace-offset function that enables you to offset the displayed trace without introducing hardware-switching errors into the stimulus-response measurement. The input attenuator and IF step gains are not affected when using NRL.

In absolute power mode (dBm), reference level affects the gain and RF attenuation settings of the instrument, which affects the measurement or dynamic range. In normalized mode (relative power or dB-measurement mode), NRL offsets the trace data on-screen and does not affect the instrument gain or attenuation settings. This allows the displayed normalized trace to be moved without decreasing the measurement accuracy due to changes in gain or RF attenuation. If the measurement range must be changed to bring trace data on-screen, then the range...
level should be adjusted. Adjusting the range-level normalized mode has the same effect on the instrument settings as does reference level in absolute power mode (normalize off).

**Query Response**

```
001
```
NRPOS Normalized Reference Position

Syntax

![Diagram of NRPOS command flow]

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any integer number</td>
<td>Oto 10</td>
</tr>
</tbody>
</table>

Equivalent Key: **Norm Ref Posn.**

Preset State: 10 (top graticule line)
Step Increment: 1

Example

10 OUTPUT 718;"IP;SNGLS;"
20 OUTPUT 718;"FA 300KHZ;FB 1GHZ;"
30 OUTPUT 718;"SRCFW ON;"
40 OUTPUT 718;"SWPCPL SR;"
50 PRINT "CONNECT THRU. PRESS CONTINUE WHEN READY TO STORE."
60 PAUSE
70 OUTPUT 718;"TS;DONE?;"
80 ENTER 718;Done
90 OUTPUT 718;"STORETHRU;"
100 OUTPUT 718;"TS;DONE?;"
110 ENTER 718;Done
120 OUTPUT 718;"NORMLIZE ON;"
130 OUTPUT 718;"TS;DONE?;"
140 ENTER 718;Done
150 OUTPUT 718;"NRPOS 5;TS;"
160 PRINT "RECONNECT DUT. PRESS CONTINUE WHEN READY."
170 PAUSE
180 OUTPUT 718;"NRL -10DB;"
190 OUTPUT 718;"TS;DONE?;"
200 ENTER 718;Done
Programming Commands

Programming Command Descriptions

210 LOCAL 718
220 END

Description

The NRPOS command adjusts the normalized reference-position that corresponds to the position on the graticule where the difference between the measured and calibrated traces resides. The dB value of the normalized reference-position is equal to the normalized reference level. The normalized reference-position can be adjusted between 0 and 10, corresponding to the bottom and top graticule lines, respectively.

Query Response

![Diagram showing query response with numbers and output termination](image)
OL Output Learn String

Transmits information to the controller that describes the state of the spectrum analyzer when the OL command is executed.

Syntax

Example

10 DIM Header_str$[10]
Allocates memory space for the Learn-String.

20 OUTPUT 718;"OL;"
Asks for first state.

30 ENTER 718 USING ";10A"; Header_str$
Computer receives the length information.

40 OUTPUT CRT;"Header string= "; Header_str$[1,4]

50 OUTPUT CRT;"MSB length= "; NUM(Header_str$[5])

60 OUTPUT CRT;"LSB length= "; NUM(Header_str$[6])

70 OUTPUT CRT;"MSB FW rev= "; NUM(Header_str$[7])

80 OUTPUT CRT;"LSB FW rev= "; NUM(Header_str$[8])

90 OUTPUT CRT;"Zero pad byte= "; NUM(Header_str$[9])

100 OUTPUT CRT;"ID byte= "; NUM(Header_str$[10])

110 Read_bytes=NUM(Header_str$[5])*256+NUM(Header_str$[6])

120 Read_bytes=Read_bytes-4

130 OUTPUT CRT;"Bytes to read= "; Read_bytes
140 DIM State_str$[1000]
150 State_str$=Header_str$
160 FOR I=11 To Read_bytes+10
170 ENTER 718 USING ",A;State_str$[I;1]
180 NEXT I
190 PAUSE
Send state information back to the analyzer.
200 OUTPUT 718;State_str$;
Returns the spectrum analyzer to its former state (when OL was first activated in line 20).
210 END

Description
The information received from the spectrum analyzer using the OL command is called the learn string. The learn string can be sent from the controller back to the spectrum analyzer to restore the spectrum analyzer to its original state after executing other commands.

The learn string requires a maximum of 210 bytes of storage space. See the information on saving and loading instrument states, in Chapter 3 for more information. To conserve memory space, the length of the learn string can be queried. Then the destination string can be allocated for the exact size needed.

The learn string contains the following information:

```
<lsb of data length><msb of data length> <msb of firmware revision><lsb of firmware revision> <msb of identification><lsb of identification><binary MSTATE data>
```
PCTAM Percent AM

Turns on or off the percent AM measurement.

Syntax

Equivalent Key: % **AMOnOff**.
Related Commands: MKPX, PCTAMR, MKTH.

Example

10 OUTPUT 718; "PCTAM 1;"
Turns on the percent AM measurement.

20 OUTPUT 718; "PCTAM R?;"
Queries PCTAMR. PCTAMR contains the results of the percent AM measurement.

30 ENTER 718;Percent
Stores the value of PCTAMR in the variable Percent.

40 PRINT "Percent AM is ",Percent
Prints the results.

50 OUTPUT 718; "PCTAM 0;"
Turns off the percent AM measurement.
Description

Setting PCTAM to 0 turns off the percent AM function. Setting PCTAM to 1 turns on the percent AM function. When the percent AM function is turned on, the spectrum analyzer finds the signal with the highest amplitude, and then finds two signals (with lower amplitudes) on either side of the highest signal. The highest on-screen signal is assumed to be the carrier, and the adjacent signals are assumed to be the sidebands. The amplitude levels of all three signals are measured, and the percent AM is calculated using the carrier level and the sideband with the higher amplitude level. Percent AM is calculated as follows:

\[
\text{Percent AM} = 200 \times \frac{\text{Level}_{\text{Carrier}}}{\text{Level}_{\text{Sideband}}}
\]

The percent AM measurement is repeated at the end of every sweep until you turn it off. You must query PCTAMR to determine the percent AM.

PCTAM can perform the percent AM measurement only if there are three on-screen signals that have the characteristics of a carrier with two sidebands. Also, to be considered a signal, the levels of the carrier and sideband signals must be greater than the peak excursion above the threshold. If there are not three signals that fit the characteristics of a carrier with two sidebands, the value of PCTAMR will be -100.

You can execute the PCTAM function using the command, “PCTAM 1;”.

Restrictions

Turning on the PCTAM function turns off the following functions: N dB point measurement (NDBPNT), TOI measurement (TOI), and span zoom (SPZOOM).

You should turn off the percent AM measurement (set PCTAM to 0) when you are finished.

Query Response

You can query the percent AM measurement status using the command “QPCTAM.” The response will be 0 for off and 1 for on.
PCTAMR Percent AM Response

Returns the percent AM measured by the percent AM measurement (PCTAM).

Syntax

```
PCTAMR
```

Equivalent Key: % AM On Off.
Related Commands: MKPX, PCTAM, MKTH.

Example

```
10 OUTPUT 718; "PCTAM 1;"
Turns on the percent AM measurement.

20 OUTPUT 718; "PCTAMR?;"
Queries PCTAMR. PCTAMR contains the results of the percent AM measurement.

30 ENTER 718; Percent
Stores the value of PCTAMR in the variable Percent.

40 PRINT "Percent AM is ", Percent
Prints the results.

50 OUTPUT 718; "PCTAM 0;"
Turns off the percent AM measurement.
```

Description

PCTAMR returns a -100 if the PCTAM function has not been turned on, or if the on-screen signal is not valid or is not present. PCTAM can perform the percent AM measurement only if there are three on-screen signals that have the characteristics of a carrier and two sidebands. Also, to be considered a signal, the levels of the carrier and sideband signals must be greater than the peak excursion above the threshold.
Query Response

% A.M. number output termination

OPCTAMR
**PEAKS Peaks**

Sorts signal peaks by frequency or amplitude, stores the results in the destination trace, and returns the number of peaks found.

**Syntax**

Prerequisite Commands: TS when using trace data
Related Commands: MKPX, MKTH
Example
The example uses the internal 50 MHz alignment signal.

10 OUTPUT 718; "IP;"
Initializes spectrum analyzer.

20 OUTPUT 718; "RFCALIB ON; CF 160MHZ; SP 300MHZ; RB 30KHZ; SNGLS;"
Turns on the internal 50 MHz signal and sets the analyzer state to view the signal.

30 OUTPUT 718; "RL -10DB; MKTH -80DB; MKPX 10 DB; TS;"
Sets up threshold, sets minimum peak excursion.

40 OUTPUT 718; "PEAKS TRB, TRA, FREQ?;"
Returns the number of peaks in trace A above the threshold.

50 ENTER 718; Number

Gets the number of peaks from the spectrum analyzer.

60 DISP Number
Displays the result on the computer screen.

70 FOR I=1 TO Number
For one to the number of peaks, do the following steps.

80 OUTPUT 718; "MKP TRB[";I;"];"
Place marker at the position of the first trace B element.

90 OUTPUT 718; "MKA?;"
Find the amplitude of the marker.

100 ENTER 718; A

110 OUTPUT 718; "MKF?;"
Find the frequency of the marker.

120 ENTER 718; B

130 PRINT A, B
Print the amplitude and the frequency of the marker.
Repeat the FOR NEXT loop for all of the peaks that were found.

END

Description

When sorting by frequency (FREQ), PEAKS first computes the horizontal position of all peaks. These positions are loaded into the destination trace consecutively, with the lowest frequency value occupying the first element. Thus, signal frequencies, from low to high, determine the amplitude of the destination trace from left to right.

When sorting by amplitude (AMP), PEAKS first computes the amplitudes of all peaks in the source trace in measurement units, and sorts these values from high to low. The positions of the peaks are then loaded into the destination trace, with the position of the highest amplitude value occupying the first element.

For example, the following spectrum analyzer display shows several peaks:

![Spectrum Analyzer Display](image)

**Figure 15**  
Frequency and Amplitude of the Peaks

If the FREQ parameter is used with the PEAKS command, the displayed peaks would be described by the values shown in the following table.
If the AMP parameter is used with the PEAKS command, the displayed peaks would be described by the values that are shown in the following table.

<table>
<thead>
<tr>
<th>Trace Element</th>
<th>Amplitude</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRB[1]</td>
<td>-25.85</td>
<td>4.98E+7</td>
</tr>
<tr>
<td>TRB[2]</td>
<td>-53.82</td>
<td>1.0E+8</td>
</tr>
<tr>
<td>TRB[3]</td>
<td>-46.17</td>
<td>1.495E+8</td>
</tr>
<tr>
<td>TRB[4]</td>
<td>-68.65</td>
<td>2.5E+8</td>
</tr>
</tbody>
</table>

Notice that MKA? and MKF? are used to determine the amplitude and frequency of the peak position.

PEAKS sorts only signals that are above the threshold value. To be classified as a signal peak, a signal must be MKPX above the threshold value and it must rise and fall at least the peak excursion (MKPX value). To change the threshold, use the MKTH command before PEAKS is executed.

If necessary, the last sorted value is repeated to fill remaining elements of the destination trace.

PEAKS must be used as a query. Form a query by ending the PEAKS statement with a “?.” When used as a query, PEAKS returns the number of peaks found.

**Query Response**

![Query Response Diagram](image-url)
POWERON Power-On State

Selects the state of the spectrum analyzer when the spectrum analyzer is turned on: the IP state (same state as an instrument preset command) or last state (the state the spectrum analyzer was in when the spectrum analyzer was turned off).

Syntax

```
POWERON IP
```

Equivalent Key: Power On IP Last.

Example

```
10 OUTPUT 718;"POWERON LAST;"
```

Description

POWERON LAST restores the last state of the spectrum analyzer. Limit line testing is not considered to be a spectrum analyzer state and is not resumed after the spectrum analyzer is turned off. The limit line table will be restored even if the spectrum analyzer is turned off, however.

NOTE:
The last state of the spectrum analyzer is not retained in the case of battery power failure of the spectrum analyzer’s internal battery.

Query Response
PREAMPG External Preamplifier Gain

Subtracts a positive or negative preamplifier gain value from the displayed signal.

Syntax

```
PREAMPG
```

Equivalent Key: Ext Amp Gain

Example

```
10 OUTPUT 718; "PREAMPG 10DB;"
```

Description

Unlike using ROFFSET, PREAMPG can change the attenuation depending on the preamplifier gain entered.

A preamplifier gain offset is used for measurements that require an external preamplifier or long cables. The offset is subtracted from the amplitude readout so that the displayed signal level represents the signal level at the input of the preamplifier or long cable. The preamplifier gain offset is displayed at the top of the screen and is removed by entering zero.

NOTE: PREAMPG is not reset to 0 by an instrument preset (IP). Be sure to execute “PREAMPG 0;” when the preamplifier gain is no longer needed.

Query Response
PRINT Print

Initiates a output of the screen data to the remote interface. With appropriate commands, the data can be routed to an external printer.

Syntax

Equivalent Key: Printer.
Related Commands: GETPRNT, PLOT.

Example for the HP-IB Interface

The printer usually resides at address 1. (The program is only valid for HP 9000 Series 200 and 300 computers and HP Vectra personal computer with an HP raster graphics printer, such as the HP ThinkJet.)

This example illustrates how an external controller can initiate the sending of print data to an external printer.

10 OUTPUT 718;"PRINT;"
20 SEND 7;UNT UNL LISTEN 1 TALK 18 DATA
Sends data to printer.

**NOTE:**
To print without disconnecting the computer, you must execute the following BASIC commands: `ABORT 7 LOCAL 7`. Then press `Print`.

**Description**
The data is output in HP raster graphics format. PRINT, PRINT 0, or PRINT BW produces a monochrome printout. PRINT 1 and PRINT PJCOLOR produces a “color format” output for an HP PaintJet printer. Execute “MENU 0;” before printing to blank the softkeys.

To send printer data to a parallel printer, see the GETPRNT command.
PRNPRT Printer Port

Selects which port to output printer data from the analyzer. (Option 1AX or Option A4H)

Syntax

Equivalent Key: Port.
Related Commands: PRINT. GETPRINT. PARSTAT, PLTPRT

Example

10 OUTPUT 718;"PRNPRT 4;"
Routes the print information to the serial port for Option 1AX.

Description

PRNPRT 0 = HP-IB port of Option A4H
PRNPRT 3 = serial port of Option 1AX
PRNPRT 4 = parallel port of Option A4H
PRNPRT 4 = parallel port of Option 1AX

NOTE: Setting the PRNPRT to a port inconsistent with the installed hardware option is ignored, so executing PRNPRT 3 on an HP-IB equipped analyzer is ignored.

Query Response
PRNTADRS Printer Address

Sets the HP-IB address of the printer.

**Syntax**

```
PRNTADRS <number>
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any valid integer number</td>
<td>0 to 30</td>
</tr>
</tbody>
</table>

Equivalent Key: **Printer Addr**
Option Required: Option **A4H**

**Example**

```
20 OUTPUT 718;"PRNTADRS 1;"
```

Sets the HP-IB address of the printer to 1.

**Query Response**

PRNTADRS? returns the current HP-IB address of the printer.
PRNTMARGB Printer Margin Bottom

Allows you to set the printer’s bottom margin when using a user defined printer.

Syntax

```
PRNTMARGB number
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any valid integer number</td>
<td>0 to 3000</td>
</tr>
</tbody>
</table>

Equivalent Key: **Bottom Margin.**

Option Required: Option **1AX** or **A4H**.

Example

```
20 OUTPUT 718;"PRNTMARGB 10;"
```

Sets the user defined printer bottom margin to 10 blank raster lines.

Description

When specifying a user defined printer, the bottom margin can be set using the PRNTMARGB command. It is set in terms of blank raster lines. The equivalent height, in cm, varies depending on the printer resolution.

Query Response

The PRNTMARGB command returns the number of lines in the current bottom margin of the user defined printer.
PRNTMARGT Printer Margin Top

Allows you to set the printer’s top margin when using a user defined printer.

Syntax

```
PRNTMARGT SP number ;
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any valid integer number</td>
<td>0 to 3000</td>
</tr>
</tbody>
</table>

Equivalent Key: **Top Margin**.
Option Required: Option 1A8, A4H.

Example

```
20 OUTPUT 718; "PRNTMARGT 10;"
```

Sets the user defined printer top margin to 10 blank raster lines.

Description

When specifying a user defined printer, the top margin can be set using the PRNTMARGT command. It is set in terms of blank raster lines. The equivalent height, in cm, varies depending on the printer resolution.

Query Response

The PRNTMARGT command returns the number of lines in the current top margin of the user defined printer.
Programming Commands

Programming Command Descriptions

`top` number output termination

QPRNTMRT
PRNTRES Printer Resolution

Allows you to set the printer resolution when using a user defined printer.

Syntax

```
PRNTRES <number> ;
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any valid integer number</td>
<td>75 to 2600</td>
</tr>
</tbody>
</table>

Equivalent Key: Resolution.

Option Required: Option 1 AX, A4H.

Example

```
20 OUTPUT 718; "PRNTRES 75;"
```

Sets the user defined printer resolution to 75.

Description

When specifying a user defined printer, the resolution can be set using the PRNTRES command. Set the value appropriate for your printer. Typical values might be 75, 100,300. Decreasing the value of the printer resolution will increase the size of the printed image.

Query Response

The PRNTRES command returns the resolution of the user defined printer.
PSTATE Protect State

Protects all of the spectrum analyzer’s user state and trace registers from being
changed.

Syntax

```
(PSTATE)

SP

OFF

ON

XPSTATE
```

Equivalent Key: **Internal Lock On Off**
Related Commands: ERASE, DELETE, SAVE, LOAD

**Example**

```
10 OUTPUT 718;"PSTATE ON;"
```

**Query Response**

```
```

002

338
**PWRBW Power Bandwidth**

Computes the bandwidth around the trace center, which includes signals whose total power is a specified percentage of the total trace signal power.

**Syntax**

```
   PWRBW SOURCE TRACE RANGE
   number PERCENTAGE trace element
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace range</td>
<td>A segment of trace A, trace B, trace C</td>
<td></td>
</tr>
<tr>
<td>Trace element</td>
<td>An element of trace A, trace B, trace C</td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>Any real or integer number</td>
<td>0 to 100</td>
</tr>
</tbody>
</table>

Parameter Values: The field used for the percentage must be a value between 0 and 100.

Prerequisite Commands: TS when using trace data.
Related Commands: RB, SP.
**Example**

10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 PAUSE

30 DISP "TURN ON THE 50 MHz OSCILLATOR"
Displays a user prompt.

40 OUTPUT 718;"SNGLS;"
Activates single-sweep mode.

50 OUTPUT 718;"CF 50MHZ;SP 1MHZ;RB 300KHZ;"
Changes center frequency, span, and bandwidth.

60 OUTPUT 718;"MXML TRA;TS;TS;TS;TS;"
Activates the maximum hold of trace A, sweep 4 times.

70 OUTPUT 718;"PWRBW TRA, 99.0;"
Returns the 99% power bandwidth.

80 ENTER 718;P
Gets the result from the spectrum analyzer.

90 DISP "THE POWER BANDWIDTH AT 99 PERCENT IS";P/1.0E+3;"kHz"
Displays the frequency of the power bandwidth specified on the computer screen.

**Description**

If trace A is the source, a delta marker is set at the start and stop frequencies.

If 100% is specified, the power bandwidth equals the frequency range of the screen display. If 50% is specified, trace elements are eliminated from either end of the array, until the combined power of the remaining signal responses equals half of the original power computed. The frequency span of these remaining trace elements is the power bandwidth returned.
Query Response

- number
- output termination

001
Programming Commands
Programming Command Descriptions

PWRUPTIME Power Up Time

Returns the number of milliseconds that have elapsed since the spectrum analyzer was turned on.

Syntax

XPWRUPTIME

Example

10 OUTPUT 718;"PWRUPTIME;"
Executes PWRUPTIME.

20 ENTER 718;A
Places the result of PWRUPTIME into A.

30 A = A/1000
Changes the milliseconds to seconds.

40 PRINT "Minutes elapsed ",A/60
Prints the number of minutes that have elapsed since the spectrum analyzer was turned on.

Description

PWRUPTIME can count the number of milliseconds for up to $2^{32}$ milliseconds ($2^{32}$ milliseconds is equivalent to 49.7 days). If the spectrum analyzer is left on for more than 49.7 days, PWRUPTIME is reset to 0 and restarts the count.

Query Response
RB Resolution Bandwidth

Specifies the resolution bandwidth.

Syntax

```
RB
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number.</td>
<td>1 kHz to 3 MHz</td>
</tr>
<tr>
<td></td>
<td>Default unit is Hz.</td>
<td></td>
</tr>
</tbody>
</table>

Equivalent Key: **Resolution BW Auto Man.**

Preset State: 3 MHz.

Step Increment: In a 1, 3, 10.

Related Commands: AUTO, SP, ST, VB, VBR.

**Example**

```
10 OUTPUT 718;"RB 1KHZ;"
```

Sets the resolution bandwidth to 1 kHz.
**Description**

The coupling between sweep time and resolution bandwidth is terminated by this command. Execute RB AUTO to reestablish coupling. (Also see “AUTO.”)

The front-panel knob, step increment keys, and auto-coupled settings provide a 1, 3, 10 resolution bandwidth sequence only. Frequencies are rounded to the nearest value in the 1, 3, 10 sequence.

**Query Response**

```
  number  output termination
```

001
REV Revision

Returns the date code of the firmware revision date in YYYYMMDD format (for example, 199709 10 indicates 10 September 1997).

Syntax

```
REV
```

Equivalent Key: Show Config displays the firmware revision date
Related Commands: ID, SER, SHOWSYS, TIMEDATE

Example

```
10 OUTPUT 718;"REV;"
```

Gets the firmware revision date of spectrum analyzer.

```
20 ENTER 718;A
```

Puts the spectrum analyzer response in the computer variable, A.

```
30 DISP A
```

Displays the firmware revision date on the computer screen.

Description

The date of the firmware revision also appears when the instrument is first turned on.
Query Response

```
year

digit digit digit digit

month
day

output termination

QREV
```
RFCALIB 50 MHz Signal

Turns on or off the internal 50 MHz alignment signal.

Syntax

Equivalent Command: 50 MHz osc On Off
Preset State: 50 MHz osc Off

Example

10 OUTPUT 718; "RFCALIB OFF;"

Description

With the RFCALIB command on, a 50 MHz alignment signal is routed through internal circuitry to the analyzer’s RF input and the front panel RF INPUT connector is disconnected.
The impedance match with the device under test (DUT) varies as the internal alignment signal is switched in and out. When the internal signal is on, the RF INPUT port is switched off presenting a high impedance to the DUT.
**RL Reference Level**

Specifies the amplitude value of the reference level.

**Syntax**

```
RL <number>;
```

- **Item**: Number
- **Description/Default**: Any real or integer number. Default unit is the current amplitude unit.
- **Range**: Amplitude range of the spectrum analyzer

Equivalent Key: **Ref Level**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is the current amplitude unit.</td>
<td>Amplitude range of the spectrum analyzer</td>
</tr>
</tbody>
</table>

Preset State: 0 dBm
Step Increment: by 10 dBm
Related Commands: AT, MKRL, ML

**Example**

10 OUTPUT 718;"IP;SNGLS;CF 300MHZ;SP 20MHZ;"

Initializes spectrum analyzer, activates single-sweep mode, changes center frequency, span.

20 OUTPUT 718;"TS;MKPK HI;MKRL;TS;"

Takes sweep, places marker on signal peak, sets ref level to marker level.
Programming Commands
Programming Command Descriptions

30 OUTPUT 718;"RL?;"
Queries reference level.

40 ENTER 718;Ref_level
Putting the spectrum analyzer response in the computer variable, Ref_level.

50 PRINT "REFERENCE LEVEL IS",Ref_level,"DM"

60 END

Description
The reference level and input attenuator are coupled to prevent gain compression. Signals with peaks at or below the reference level are not affected by gain compression.

CAUTION: Signal levels above +30dBm will damage the spectrum analyzer.

RL may affect the attenuation value.

Query Response

CAUTION: 001
RMS Root Mean Square Value

Returns the root mean square value of the trace in measurement units.

Syntax

```
RMS
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace range</td>
<td>A segment of trace A, trace B, trace C</td>
<td></td>
</tr>
</tbody>
</table>

Prerequisite Commands: TS when using trace data.
Related Commands: MEAN, STDEV, VARIANCE.

Example

```
10 OUTPUT 718;"IP;SNGLS;TS;"
20 OUTPUT 718;"RMS TRA?;"
30 ENTER 718;Number
40 DISP Number
```

Description

Trace data, user-defined trace data, and trace range data are treated as 16-bit integers.
Query Response

number → output termination
ROFFSET Reference Level Offset

Offsets all amplitude readouts without affecting the trace.

Syntax

```
Syntax
```

```
Item | Description/Default | Range
---|---------------------|----
Number | Any real or integer number. Default unit is dB. | -200 dB to +200 dB
```

Equivalent Key: Ref Lev Offst.
Preset State: 0 dB.
Related Commands: AT, RL.

Example

10 OUTPUT 718;“IP;”
Initializes spectrum analyzer.

20 OUTPUT 718;“RL −20DB;”
Changes the reference level.

30 OUTPUT 718;“ROFFSET −10;”
Changes spectrum analyzer reference offset value.

40 OUTPUT 718;“RL?;”
Gets the reference value from spectrum analyzer.

50 ENTER 718;Ref
Puts the spectrum analyzer response in the computer variable, Ref.

60 DISP "THE NEW REFERENCE LEVEL IS ",Ref
Displays -30 as the new reference level.

70 END

Description
Once activated, the ROFFSET command displays the amplitude offset in the active function block. And, as long as the offset is in effect, the offset is displayed on the left side of the screen.

Entering ROFFSET 0 or presetting the spectrum analyzer eliminates an amplitude offset.

Query Response
RQS Service Request Mask

Sets a bit mask for service requests (see “SRQ”).

**Syntax**

```
RQS S_n p number ;
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any valid integer</td>
<td>0 to 63</td>
</tr>
</tbody>
</table>

Related Commands: SRQ, STB
Preset State: 41

**Example**

```
10 OUTPUT 718; "RQS 12;"
```

Sends a mask bit for hardware broken and end of sweep.

**Description**

Assignment of values for the mask is as follows:

- 32 = Illegal command (bit 5)
- 16 = Command complete (bit 4)
- 8 = Hardware broken (bit 3)
- 4 = End of sweep (bit 2)
- 2 = Units key pressed (bit 1)
- 1 = Operator notification (bit 0)

As shown in the example, a mask with hardware broken and end of sweep is equal to 12 (8 + 4). The mask also disables command complete, illegal command, units key pressed, and operator notification interrupts.
To activate all conditions in the mask, the mask value is equal to 63 (32 + 16 + 8 + 4 + 2 + 1). To set the service request mask for all conditions, execute `OUTPUT 718; "RQS 63; "`.

Each bit in the status byte is defined as follows:

0 Indicates that an operator notification has occurred. SRQ $S$ appears on the analyzer screen. Operator notification occurs if an overload is detected on the analyzer RF input, if excessive reverse power is detected on the RF output (options IDN or IDQ only), or if the tracking generator becomes unleveled (options IDN or IDQ only.)

1 Indicates that the units key was pressed. SRQ $S$ appears on the spectrum analyzer screen. If you activate the units key bit, it will remain active until you activate “EE” and press a units key. (See “EE.”)

2 Indicates end of sweep. SRQ $S$ appears on the spectrum analyzer screen. If you send any RQS value that contains mask value 4, another sweep will be taken.

3 Indicates broken hardware. SRQ $S$ appears on the spectrum analyzer screen.

4 Indicates completion of a command. It is triggered by EOI at the end of a command string or the completion of a print or plot.

5 Indicates an illegal spectrum analyzer command was used. SRQ $S$ appears on the spectrum analyzer screen.

0 (LSB), 6, and 7 are not used.

A service request is generated only if the proper request mask bit has been set, and either the condition itself or the Force Service Request (see “SRQ”) is sent. To set the request mask, choose the desired interrupt conditions and sum their assigned values. Executing the RQS command with this value sets the bit mask. After setting the bit mask, only the chosen conditions can produce an interrupt. Generally, you must set the bit mask using the RQS command. However, the “hardware broken”, “operator notification” and “illegal remote command” conditions are automatically enabled after presetting or sending the IP command. Pressing Preset or sending the IP command, then, produces the same interrupt bit mask as sending “RQS 41;” (decimal 41 is the sum of the assigned values of these three interrupt bits, 32 = Bit 5, 8 = Bit 3, and 1 = Bit 0).

For most conditions, the RQS mask bit stays set until the next IP or RQS command is executed. The only condition in which this does not apply is the Units Key Pressed bit. When this bit (bit 1) is set in the RQS mask, a Units Key Pressed interrupt occurs if EE (enable entry mode) is executed and a front-panel units key such as Hz, kHz, MHz, or GHz is pressed.

When a units key is pressed, the interrupt occurs and the Units Key Pressed bit in the RQS mask is reset. To re-enable the Units Key Pressed interrupt, you must send a new RQS mask.
Use SRQ to Check for an Overpower Condition at the Input

The analyzer checks for an overpower condition at the input. It may respond to an overload by changing the attenuation setting or by switching out the input. This can cause misleading results in a measurement. The status bit should be monitored for this condition to avoid incorrect measurement results. See the following error message descriptions for more details about these conditions:

- **Atten auto set to 15 dB**
- **Overload: Reduce Signal**

**Query Response**

```
  number                               output termination
  001
```
SAVE Save File

Specifies the file to be saved in memory.

Syntax

```
SAVE <delimiter> <filename>
```

**Item** | **Description/Default** | **Range**
--- | --- | ---
Delimiter | Matching characters marking the beginning and end of the filename | \1\2\^\$\%\!
Filename | See the file naming rules below | 

Related Commands: LOAD, DELETE, PSTATE

**File Naming Rules**

File names for storing states, traces, limit lines or amplitude correction data files should follow the conventions as indicated below:
They can be up to eight characters long. In addition, they can have a tile extension up to three characters long. The analyzer assigns the extension.

They are not case sensitive. It does not matter whether you use upper case or lower case letters when you type them.

They can contain only the letters A through Z, the number 0 through 9, and the following special characters. No other characters are valid.

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_</td>
<td>underscore</td>
</tr>
<tr>
<td>^</td>
<td>carat</td>
</tr>
<tr>
<td>$</td>
<td>dollar sign</td>
</tr>
<tr>
<td>~</td>
<td>tilde</td>
</tr>
<tr>
<td>!</td>
<td>exclamation point</td>
</tr>
<tr>
<td>#</td>
<td>number sign</td>
</tr>
<tr>
<td>%</td>
<td>percent sign</td>
</tr>
<tr>
<td>&amp;</td>
<td>ampersand</td>
</tr>
<tr>
<td>-</td>
<td>hyphen</td>
</tr>
<tr>
<td>{}</td>
<td>braces</td>
</tr>
<tr>
<td>@</td>
<td>at sign</td>
</tr>
<tr>
<td>'</td>
<td>single quotation mark</td>
</tr>
<tr>
<td>'</td>
<td>apostrophe</td>
</tr>
<tr>
<td>0</td>
<td>parenthesis</td>
</tr>
</tbody>
</table>

They cannot contain spaces, commas, backslashes, or periods. (except the period that separates the name from the extension.)

They cannot be identical to the name of another file in the same directory.

**Example**

```
10 OUTPUT 718;"SAVE %MYSTATE%,STATE;"
```

Saves analyzer state data in a file named **MYSTATE.STA**.
Description

The SAVE command is used to save files in analyzer memory. There are several different types of file data. The type of data saved, and its corresponding data source, are shown in the following table. The data source must be specified when the command is sent. The analyzer will attach the appropriate file extension to the filename you supplied.

Some of the characters that are available to use as delimiters are also allowed in the file name. Do not use the same character for the delimiters as you intend to use in the file name, since the analyzer will interpret the character as the end delimiter, rather than a part of the file name. Do not use any of the common characters twice in the file name, as the analyzer will interpret them as delimiters.

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Type of Data Transferred</th>
<th>File Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRA</td>
<td>Trace A</td>
<td>.TRC</td>
</tr>
<tr>
<td>TRB</td>
<td>Trace B</td>
<td>.TRC</td>
</tr>
<tr>
<td>TRC</td>
<td>Trace C.</td>
<td>.TRC</td>
</tr>
<tr>
<td>STATE</td>
<td>Instrument state</td>
<td>.STA</td>
</tr>
<tr>
<td>LIMILINE</td>
<td>Limit lines</td>
<td>.LIM</td>
</tr>
<tr>
<td>AMPCOR</td>
<td>Amplitude correction factors</td>
<td>.AMP</td>
</tr>
</tbody>
</table>
SEGDEL Segment Delete

Deletes the specified segment from the limit line tables.

Syntax

```
SEGDEL  number
```

Related Commands: LIMIHALF, LIMISEG, LIMISEGT

Example 1

This example uses LIMIHALF for entering segments into the upper limit line table, then entering a segment into the lower limit line table (upper and lower limit lines are treated as separate tables).

```
10 OUTPUT 718;"LIMIDEL;"
```

Deletes the current limit line table, sets the table type to fixed.

```
20 OUTPUT 718;"LIMIHALF UPPER;"
```

Specifies the upper limit line table.

```
25 OUTPUT 718;"LIMIFT FREQ;"
```

Limit lines to be based on frequency.

```
30 OUTPUT 718;"LIMISEG 300MHZ,-30DB,FLAT;"
```

Enters a segment into the upper limit line table.

```
40 OUTPUT 718;"LIMIHALF LOWER;"
```

Specifies the lower limit line table.

```
50 OUTPUT 718;"LIMISEG 300MHZ,-70DB,FLAT;"
```

Enters a segment into the lower limit line table.

```
60 OUTPUT 718;"SEGDEL 1;"
```

Deletes the segment from the lower limit line table.
Description

The result of SEGDEL depends on the setting of the LIMIHALF command as shown in the following table.

<table>
<thead>
<tr>
<th>LIMIHALF Setting</th>
<th>Result of SEGDEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIMIHALF LOWER</td>
<td>Deletes specified segment from the lower limit line table.</td>
</tr>
<tr>
<td>LIMIHALF ULOW</td>
<td>Deletes specified segment from the upper and lower limit line table.</td>
</tr>
</tbody>
</table>

You may want to query LIMIHALF before using SEGDEL if you are unsure of the LIMIHALF setting.

To determine the number of each segment, you can use the softkeys accessed by Edit Limit to display the limit line table. Limit line entries are sorted according to frequency or time.
SER Serial Number

Returns the serial number of the spectrum analyzer.

**Syntax**

```
SER
```

**Example**

```
10 DIM Serial$[24]
Displays the serial number on the analyzer display.

20 OUTPUT 718; "SER?;"
Gets the serial number from the spectrum analyzer.

30 ENTER 718; Serial$
Puts the spectrum analyzer response in the computer variable.

40 PRINT Serial$
Prints the serial number.
```

**Query Response**
SETDATE Set Date

Allows you to set the date of the real-time clock of the spectrum analyzer.

Syntax

```
SETDATE (YYYYMMDD)
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>A number in the YYYYMMDD format</td>
<td>Valid year, month, and day</td>
</tr>
</tbody>
</table>

Equivalent Key: Set Date.
Related Commands: SETTIME, TIMEDATE, TIMEDSP.

Example

```
10 OUTPUT 718; "SETDATE 890212;"
```
Sets the date to February 12, 1989.

Query Response

```
year - digit digit digit digit
month - digit digit
day - digit digit
output termination
```

QSE TEDATE
**SETTIME: Set Time**

Allows you to set the time of the real-time clock of the spectrum analyzer.

**Syntax**

```
SETTIME <time (HHMMSS)>
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>A number in the HHMMSS (24 hour) format</td>
<td>0 to 23599</td>
</tr>
</tbody>
</table>

Equivalent Key: **Set Time**.
Related Commands: **SETDATE, TIMEDATE, TIMEDSP**.

**Example**

```
10 OUTPUT 718;"SETTIME 135501;"
```

Sets the time to 1:55:01 P.M.

**Query Response**

```
QSETTIME
```
SHOWSYS Show System

Shows the system configuration on the spectrum analyzer display.

Syntax

```
SHOWSYS

XSHOWSYS
```

Equivalent Key: Show System
Related Commands: ID, SER, REV

Example

```
10 OUTPUT 718; “SHOWSYS;”
```

Displays the system configuration on the analyzer display.
SMOOTH Smooth Trace

Smooths the trace according to the number of points specified for the running average.

Syntax

```
SMOOTH

number

Trace range  A segment of trace A, trace B, trace C
Number       Any real or integer number  Real number range
Trace element An element of trace A, trace B, trace C
```

Prerequisite Commands: TS when using trace data.
Related Commands: SNGLS, TS, VAVG.
Example

10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"SNGLS;TS;"
Activates single-sweep mode, takes a sweep.

30 OUTPUT 718;"VIEW TRA;"
Stores results of trace A.

40 OUTPUT 718;"SMOOTH TRA,10;"
Smoothes trace A.

50 OUTPUT 718;"VIEW TRA;"
Displays the result.

60 END

Description

Each point value is replaced with the average of the values (in measurement units) of the given number of points centered on it. Increasing the number of points increases smoothing at the cost of decreasing resolution. If the number of points is an even number, then the number of points is increased by one. If the number of points is larger than the size of SOURCE, then the size of SOURCE is used (unless size of SOURCE is even, in which case the size of SOURCE minus one is used). Smoothing decreases at the endpoints.

The purpose of this function is to perform a spatial video averaging as compared to the temporal version supplied by the video-average (VAVG) command. The functions of SMOOTH and VAVG are not interchangeable however. Unlike VAVG, SMOOTH averages values that occur before and after the data point in time. This can cause some display irregularities at the start and stop frequencies. Use low values for the SMOOTH parameter to avoid signal distortion.

By replacing the value of each point in a trace with the average of the values of a number of points centered about that point, any rapid variations in video noise or signals are smoothed into more gradual variations. It thereby performs a function similar to reducing the video bandwidth without the corresponding changes in sweep time. As such, it does result in a reduction of frequency resolution. Also, signal peaks are reduced with large smoothing values, and this can cause the amplitude to appear to be low.
This command requires user memory for its execution. Memory is not permanently allocated, so the largest amount of memory is available for the functions that are used in a particular application. When the command is complete, memory is returned to the free user memory.
SNGLS Single Sweep

Sets the spectrum analyzer to single-sweep mode.

Syntax

```
SNGLS
```

Equivalent Keys: **Single Sweep** or **Sweep Cont Single** (Single is underlined).
Related Commands: CLRW, CONTS, TM, TS.

Example

```
10 OUTPUT 718; "SNGLS;"
```

Description

Each time TS (take sweep) is sent, one sweep is initiated, as long as the trigger and data entry conditions are met.
SP Span

Changes the total displayed frequency range symmetrically about the center frequency.

**Syntax**

```
SP span
```

**Item** | **Description/Default** | **Range**
--- | --- | ---
Number | Any real or integer number. Default unit is Hz. | Frequency span of the spectrum analyzer

Equivalent Key: **Span**
Step Increment: 1, 2, 5, 10 sequence (up to the stop frequency of the spectrum analyzer)
Related Commands: CF, FA, FB, FOFFSET, FS, RB, ST, VB

**Example**

```
10 OUTPUT 718;"IP;SP 20MHZ;"
```

Initializes spectrum analyzer, changes frequency span.
20 OUTPUT 718;"SP?;"
Gets the span value from the spectrum analyzer.

30 ENTER 718;Span
Puts the spectrum analyzer response in the computer variable, Span.

40 PRINT Span
Displays the span value.

**Description**

The frequency span readout refers to the displayed frequency range. Dividing the readout by 10 yields the frequency span per division.

If resolution and video bandwidths are coupled to the span width, the bandwidths change with the span width to provide a predetermined level of resolution and noise averaging. Likewise, the sweep time changes to maintain a calibrated display, if coupled. All of these functions are normally coupled, unless RB, VB, or ST have been executed.

Because span is affected by frequency, change the frequency before changing span.

Specifying 0 Hz enables zero-span mode, which configures the spectrum analyzer as a fixed-tuned receiver.

**Query Response**

![Query Response Diagram](image-url)
SPEAKER Speaker

Turns on or off the internal speaker.

Syntax

Preset State: SPEAKER ON
Related Commands: DEMOD

Example

10 OUTPUT 718; "SPEAKER OFF;"
SPZOOM Span Zoom

Places a marker on the highest on-screen signal (if an on-screen marker is not present), turns on the signal track function, and activates the span function.

Syntax

```
XSPZOOM
```

Equivalent Key: **Span Zoom**.

Example

```
10 OUTPUT 718;“IP;CF 300MZ;TS;”
20 OUTPUT 718;“SPZOOM;”
```

Description

If a marker is present before SPZOOM is executed, SPZOOM turns on the signal track function and activates the span function.
SQR Square Root

Places the square root of the source into the destination.

**Syntax**

```
SQR, destination.
```

### Item Description/Default Range

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace range</td>
<td>A segment of trace A, trace B, trace C</td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>Any real or integer number</td>
<td>Real number</td>
</tr>
</tbody>
</table>

Prerequisite Commands: TS when using trace data
Related Commands: STDEV
**Example**

10 \texttt{OUTPUT 718;"SQR SP,1E8;"}

Changes the span to 10 kHz.

**Description**

If the source is negative, the square root of the absolute value will be returned.

When the source is longer than the destination, the source is truncated to fit. When the source is shorter than the destination, the last element is repeated to fill the destination.
SRCAT Source Attenuator

Attenuates the source output level.

Syntax

```
SRCAT
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number, specified in multiples of 10 dB</td>
<td>0 to 60 dB</td>
</tr>
</tbody>
</table>

Equivalent Key: **Attenuation Auto Man**
Option Required: Option 1DN or 1DQ
Coupling: Coupled to power level of the source output (SRCPWR) when set to auto (SRCAT AUTO)
Related Commands: SRCPSTP
Preset State: SRCAT AUTO

Example

The following example uses the SRCAT command to attenuate the source output.

```
10 OUTPUT 718; "SRCAT AUTO; "
```

Activates source-attenuation coupling.
20 OUTPUT 718; "SRCPWR -20DB;"
Activates source output.

30 OUTPUT 718; "SRCAT 20DB;"
Sets attenuator to 20 dB. This de-couples the attenuator from the source power-level setting.

**Description**
The SRCAT command attenuates the output level of the source. Use SRCAT to attenuate the power level of the source manually, from 0 to 60 dB in 10 dB steps. “SRCAT AUTO;” automatically adjusts the attenuator to yield the source amplitude level specified by the SRCPWR command.

**Query Response**

![Query Response Diagram]
SRCPOFS Source Power Offset

Offsets the source power level readout.

Syntax

```
SRCPOFS Offset
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number</td>
<td></td>
</tr>
</tbody>
</table>

Option Required: Option `IDN` or `IDQ`.
Equivalent Key: **Amptd Offset**.
Related Commands: SRCPWR, SRCPSWP.
Step Increment: Determined by SRCPSTP.
Preset State: 0 dB.

Example

Use SRCPOFS to offset the power-level readout for the tracking-generator source.

```
20 OUTPUT 718; "SRCPWR –10DB;"
```

Turns on source output.

```
30 OUTPUT 718; "SRCPOFS 13DB;"
```

Offsets power-level readout for source by 13 dB.
Description

The SRCPOFS command offsets the displayed power of the built-in tracking generator. This function may be used to take into account system losses (for example, cable loss) or gains (for example, preamplifier gain) reflecting the actual power delivered to the device under test.

Query Response

[number] [output termination]
SRCPSTP Source Power-Level Step Size

Selects the source-power step size.

Syntax

\[ \text{SRCPSTP} \text{ number} \]

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number</td>
<td></td>
</tr>
</tbody>
</table>

Option Required: Option 1DN or 1DQ.
Equivalent Key: **Amptd Step**.
Step Increment: 0.1 dB.
Related Commands: SRCPWR, SRCPOFS, SRCPSWP.
Preset State: SRCPSTP AUTO (one major vertical scale division).

Example

Select incremental changes of power effected by "SRCPWR UP;", "SRCPWR DN;" commands, or the step keys.

20 OUTPUT 718; "SRCPWR -10DB;"
Turns on the source output.

30 OUTPUT 718; "SRCPSTP .3DB;"
Sets power-level step size to 0.3 dB.
40 output 718; "SRCPWR UP;"

Increases the power level.

**Description**

The SRCPSTP command selects the step size for the following source commands:

- Power offset (SRCPOFS).
- Power sweep (SRCPSWP).
- Power (SRCPWR).

Use SRCPSTP to set the step size to a specific value.

“SRCPSTP AUTO;” sets the step size to one vertical scale division.

**Query Response**

[Diagram showing query response]
**SRCPSWP Source Power Sweep**

Selects the sweep range of the source output.

**Syntax**

```
SRCPSWP SP number DB ;
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number</td>
<td></td>
</tr>
</tbody>
</table>

Option Required: Option 1DN or 1DQ.  
Equivalent Key: **Power Sweep On Off**.  
Step Increment: Determined by SRCPSTP.  
Related Commands: SRCPSWP, SRCPOFS, SRCPSTP.  
Preset State: SRCPSWP OFF.

**Example**

Use SRCPSWP to sweep the power level of the source output.

```
20 OUTPUT 718; "SRCPWR -10DM;"
```

Sets power level of source output to -10 dBm.

```
30 OUTPUT 718; "SP 0;"
```

Sets span to 0 Hz.
Sweeps source output from -10 dBm to 0 dBm.

Description

The SRCPSWP command works in conjunction with the SRCPWR (source power) command to sweep the amplitude level of the source output. The SRCPWR setting determines the amplitude level at the beginning of the sweep. The SRCPSWP command determines the change in amplitude level of the sweep.

For example, if SRCPWR and SRCPSWP are set to -15 dBm and 4 dB respectively, the source sweeps from -15 dBm to -11 dBm.

NOTE:

Power is swept from low to high.

The minimum sweep time is limited to 50 ms when performing a source power sweep.

Query Response

```
40 OUTPUT 718;"SRCPSPWP 10DB;"
```

Sweeps source output from -10 dBm to 0 dBm.
SRCPWR Source Power

Selects the source power level.

Syntax

![Diagram of SRCPWR]

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is the current amplitude unit.</td>
<td>Actual range is hardware dependent</td>
</tr>
</tbody>
</table>

Option Required: Option 1DN or 1DQ.
Equivalent Key: **Amplitude On Off**.
Step Increment: Set by SRCPSTP.
Related Commands: SRCAT, SRCPSTP, SRCPSWP.
Preset State: -10 dBm.

Example

Use SRCPWR to turn on the source and adjust its power level.

10 OUTPUT 718; "SRCPWR -20DM;"
Changes power level to -20 dBm.

20 OUTPUT 718; "AUNITS DBMV;"
Changes the current amplitude unit.

30 OUTPUT 718; "SRCPWR 37;"
The source power is now 37 dBmV.

**Description**

The SRCPWR command turns the source off or on and sets the power level of the source. The source is turned on automatically whenever its value is specified with SRCPWR. Also see “SRCPSTP.”

**Query Response**

```
  number | output termination
```

001
SRQ Force Service Request

The SRQ command is used by an external controller to simulate interrupts from the spectrum analyzer.

Syntax

```
SRQ service request
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any valid integer</td>
<td>1 to 127</td>
</tr>
</tbody>
</table>

Related Commands: CLS, RQS, STB

Example

```
10 OUTPUT 718;"RQS 8;SRQ 8;"
```

Sets bit mask for a hardware broken service request, generates a hardware broken interrupt.

NOTE: A program can respond to the interrupt in the same way it would under a true service request condition.

Description

The service request condition is also displayed on the spectrum analyzer screen with the annotation s in the upper right corner.

The conditions that can generate a service request are as follows:

- 32 = Illegal command
- 16 = Command complete
- 8  = Hardware broken
- 4  = End of sweep
- 2  = Units key pressed
- 1  = Operator notification
A service request is generated only if the proper request mask bit has been set (see “RQS”), and either the condition itself or the Force Service Request is sent. To set the request mask, choose the desired interrupt conditions and sum their assigned values. Executing the RQS command with this value sets the bit mask. After setting the bit mask, only the chosen conditions can produce an interrupt.

Each bit in the status byte is defined as shown in the following table. Bit numbers 6 and 7 are not used.

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Decimal Equivalent</th>
<th>Spectrum Analyzer State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>32</td>
<td>Set when an illegal command is present</td>
<td>$ appears on the spectrum analyzer screen,</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>Set when any command is completed</td>
<td>It is triggered by EOI at the end of a command string or the completion of a print or plot.</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>Indicates hardware broken condition</td>
<td>$ appears on the spectrum analyzer screen.</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Indicates end of sweep</td>
<td>$ appears on the spectrum analyzer screen. If you send any RQS value that contains mask value 4, another sweep will be taken.</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Indicates a units key was pressed</td>
<td>$ appears on the spectrum analyzer screen. If you activate the units key bit, it will remain active until you activate “EE” and press a units key. (See “EE.“)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Operator notification</td>
<td>$ appears on the spectrum analyzer screen.</td>
</tr>
</tbody>
</table>

Operator notification occurs if an overload is detected on the analyzer RF input, if excessive reverse power is detected on the RF output (options 1DN or 1DQ only), or if the tracking generator becomes unleveled (options 1DN or 1DQ only.)
Generally, you must set the bit mask using the RQS command. However, the “hardware broken”, “operator notification” and “illegal remote command” conditions are automatically enabled after presetting or sending the IP command. Pressing Preset or sending the IP command, then, produces the same interrupt bit mask as sending “RQS 41;” (decimal 41 is the sum of the assigned values of these three interrupt bits, 32 = bit 5, 8 = bit 3, and 1 = bit 0).

For most conditions, the RQS mask bit stays set until the next instrument preset (IP), or RQS command is executed. The only condition to which this does not apply is the Units Key Pressed bit. When this bit (bit 1) is set in the RQS mask, a Units Key Pressed interrupt occurs if EE (enable entry mode) is executed and a front-panel units key such as Hz, kHz, MHz, or GHz is pressed.

When a units key is pressed, the interrupt occurs and the Units Key Pressed bit in the RQS mask is reset. To re-enable the Units Key Pressed interrupt, you must send a new RQS mask. See “RQS” for detailed information.

As mentioned, you can simulate a service request condition. Choose the desired interrupt conditions from the RQS command table (see “RQS”), and sum their assigned values. Use the RQS command with this value to set the bit mask. By setting the corresponding bits in the SRQ command and sending the SRQ command to the spectrum analyzer, the desired interrupt occurs. This allows the user to verify proper operation of software routines designed to handle infrequent or unlikely interrupts.

Interface Differences
As implemented on the HP-IB interface, an spectrum analyzer service request asserts the SRQ control line on the HP-IB.

On the RS-232 interface, the spectrum analyzer does not have a way of signaling the interrupt condition to a controller. In this case, the controller must operate in a polled mode if it requires interrupt information (see “Polled Mode of Operation” below for a discussion of the polled mode).

Interrupt-Related Commands Common to All Interfaces:
- CLS Clear status byte, without read
- RQS Request mask
- SRQ Force service request
- STB Read then clear status byte

The HP-IB interface supports interface commands to read the status byte

On HP-IB in HP 9000 Series 200 or 300 BASIC, the statement SPOLL (Device-address) can be used to read the status byte.
Polled Mode of Operation

The polled mode of operation is probably most applicable to an RS-232 interface user. Because there is no interrupt signal to the RS-232 controller, the user must periodically ask the spectrum analyzer, via the “STB?” command, for the contents of its status register. For example, the RS-232 controller could periodically check for the hardware-broken condition by executing the “STB?” command and reading the results.

Use SRQ to Check for an Overpower Condition at the Input

The analyzer checks for an overpower condition at the input. It may respond to an overload by changing the attenuation setting or by switching out the input. This can cause misleading results in a measurement. The status bit should be monitored for this condition to avoid incorrect measurement results. See the following error message descriptions for more details about these conditions:

- **Atten auto set to 15 dB**
- **Overload: Reduce Signal**
SS Center Frequency Step Size

Specifies center frequency step size.

**Syntax**

```
SS Center Frequency Step Size
```

**Description/Default Range**

- **Number**: Any real or integer number. Default unit is Hz.
- **Range**: Frequency range of the spectrum analyzer

**Equivalent Key**: **CF Step Auto Man**.

**Preset State**: 100 MHz.

**Step Increment**: 1, 2, 5, 10 sequence.

**Related Commands**: AUTO, CF, FOFSET, SP.

**Example**

```
10 CLEAR 718
Clears the HP-IB.

20 OUTPUT 718; "IP;SNGLS;CF 300MHZ;SP 20MHZ;TS;"
```
Programming Commands

Programming Command Descriptions

Initializes the spectrum analyzer, activates single-sweep mode, changes the center frequency, span, takes sweep.

30 OUTPUT 718;"MKPK HI;MKRL;TS;MKF?;"
Finds the highest peak, changes the reference level to the marker, takes sweep, returns the frequency of the marker.

40 ENTER 718 USING "K";Mk_freq
Puts the spectrum analyzer response in the computer variable, Mk_freq.

50 OUTPUT 718;"MKA?;"
Returns the amplitude of the marker.

60 ENTER 718 USING "K";Mk_amp
Puts the spectrum analyzer response in the computer variable, Mk_amp.

70 OUTPUT 718;"SS ";Mk_freq;"HZ"
Changes the step size to the marker frequency.

80 OUTPUT 718;"CF UP;TS;MKPK HI;MKA?;"
Increases the center frequency, takes sweep, puts the marker on the highest peak and returns the amplitude of the marker.

90 ENTER 718;Mk_amp1
Puts the spectrum analyzer response in the computer variable, Mk_amp1.

100 PRINT "THE FUNDAMENTAL IS ";Mk_amp-Mk_amp1
110 PRINT "dB ABOVE THE SECOND HARMONIC"
Outputs the result.

120 END

Description

The AUTO parameter removes SS as an active function but does not have an effect on its value.

Query Response

<table>
<thead>
<tr>
<th>number</th>
<th>output termination</th>
</tr>
</thead>
</table>
ST Sweep Time

Specifies the time in which the spectrum analyzer sweeps the displayed frequency range.

Syntax

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any real or integer number. Default unit is seconds.</td>
<td>Within the sweep time range of the spectrum analyzer</td>
</tr>
</tbody>
</table>

Equivalent Key: **Sweep Time Auto Man**
Sweep Time Range: 5 ms to 2000s
Step Increment: 2, 3, 5, 7.5, 10, 15 sequence
Related Commands: AUTO, CONTS, RB, SNGLS, SP, SRCPSWP, TS

Example

```
10 OUTPUT 718;"ST 100MS;"
```
Sets the sweep time to 100 milliseconds.
Query Response

number -> output termination
STB Status Byte Query

Returns to the controller the decimal equivalent of the status byte.

Syntax

```
STB ?;
```

Related Commands: RQS, SRQ.

Example

10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"SNGLS;"
Activates single-sweep mode.

30 OUTPUT 718;"CLS;"
Clears the status bits.

40 OUTPUT 718;"TS;"
Takes sweep.

50 OUTPUT 718;"STB?;"
Returns the status bits.

60 ENTER 718;Status_Bit
Puts the spectrum analyzer response in the computer variable, Status-Byte.

70 PRINT Status-byte
Displays the result.

80 END
Description

The STB command is equivalent to a serial poll command. The RQS and associated bits are cleared in the same way that a serial poll command would clear them. The bits in the status byte are explained under the RQS command.

Use SRQ to Check for an Overpower Condition at the Input

The analyzer checks for an overpower condition at the input. It may respond to an overload by changing the attenuation setting or by switching out the input. This can cause misleading results in a measurement. The status bit should be monitored for this condition to avoid incorrect measurement results. See the following error message descriptions for more details about these conditions:

- **Atten auto set to 15 dB**
- **Overload: Reduce Signal**

Query Response

```
+----------------+           +----------------+
| number          |           | output termination |
+----------------+           +----------------+
```

001
STDEV Standard Deviation of Trace Amplitudes

Returns the standard deviation of the trace amplitude in measurement units.

**Syntax**

```
STDEV
```

**Table**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace range</td>
<td>A segment of trace A, trace B, trace C, or a user-defined trace.</td>
<td></td>
</tr>
</tbody>
</table>

Prerequisite Commands: TS when using trace data
Related Commands: MEAN, VARIANCE

**Example**

The following program segment finds the standard deviation of the amplitude of trace A.

10 OUTPUT 718;"IP;"
Initializes spectrum analyzer.

20 OUTPUT 718;"SNGLS;"
Activates single-sweep mode.

30 OUTPUT 718;"CF 300MHZ;SP 2MHZ;RB 100KHZ;"
Changes the center frequency, span, and resolution bandwidth.
40 OUTPUT 718;"TS;"
Takes sweep.

50 OUTPUT 718;"STDEV TRA?;"
Finds the standard deviation of trace A.

60 ENTER 718;Number
Get the response from the spectrum analyzer.

70 PRINT "THE STANDARD DEVIATION OF TRACE A " ;Number/100;"DB"

Description
The formula to calculate the standard deviation is as follows:

\[ \sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \]

- \( n \) represents the number of data points.
- \( x_i \) represents a data point.
- \( \bar{x} \) represents the mean of data.

Query Response

```
number \[ \rightarrow \] output termination
```

001
SUM Sum of Trace Amplitudes

Returns the sum of the amplitudes of the trace elements in measurement units.

**Syntax**

```
SUM
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace range</td>
<td>A segment of trace A, trace B, trace C, or a user-defined trace</td>
<td></td>
</tr>
</tbody>
</table>

Prerequisite Commands: TS when using trace data

Related Commands: MEAN, TS, VARIANCE

**Example**

10 OUTPUT 718;"IP;"

Initializes spectrum analyzer.

20 OUTPUT 718;"SNGLS;TS;"

Activates single-sweep mode, takes sweep.

30 OUTPUT 718;"SUM TRA?;"

Gets the result.

40 ENTER 718;Trace_sum

Puts the spectrum analyzer response in the computer variable, Trace-sum.
50 DISP Trace-sum;'MEASUREMENT UNITS"
Displays the result.

60 END

Query Response

- number
- output termination

001
**SUMSQR Sum of Squared Trace Amplitudes**

Returns the sum of the squares of the amplitude of each trace element in measurement units.

**Syntax**

```
<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace range</td>
<td>A segment of trace A, trace B, trace C, or a user-defined trace</td>
<td></td>
</tr>
</tbody>
</table>
```

Prerequisite Commands: TS when using trace data.

**Example**

10 OUTPUT 718;"IP;"

Initializes spectrum analyzer.

20 OUTPUT 718;"SNGLS;TS;"

Activates single-sweep mode, takes sweep.

30 OUTPUT 718;"SUMSQR TRA?;"

Gets the result.

40 ENTER 718;Trace_sqrsum

Puts the spectrum analyzer response in the computer variable, Trace-sqrsum.
50 DISP Trace-sqrsum;"MEASUREMENT UNITS"
Displays the result.

Query Response
SWPCPL Sweep Couple

Selects either a stimulus response (SR) or spectrum-analyzer (SA) auto-coupled sweep time.

Syntax

Equivalent Key: Swp Coupling SR SA.
Option Required: Option 1DN or 1DQ.
Preset State: SWPCPL SA.
Related Commands: SRCPWR.

Example

10 OUTPUT 718;"IP;SNGLS;"
20 OUTPUT 718;"FA 300KHZ;FB 1GHZ;"
30 OUTPUT 718;"SRCPWR -10DB;"
40 OUTPUT 718;"SRCPWR SR;"
50 LOCAL 718
60 END

Description

In stimulus-response mode, auto-coupled sweep times are usually much faster for swept-response measurements. Stimulus-response auto-coupled sweep times are typically valid in stimulus-response measurements when the system’s frequency span is less than 20 times the bandwidth of the device under test.

Query Response
TDF Trace Data Format

Formats trace information for return to the controller.

Syntax

```
TDF
```

Related Commands: MDS, MKA, TRA

Example

```
10 DIM A(401)
Holds trace data.

20 OUTPUT 718; “IP;”
Initializes analyzer.

30 OUTPUT 718; “BLANK TRA;CLRW TRB;”
Views trace B.

40 OUTPUT 718; “SNGLS;CF 300MHZ;SP 2MHZ;TS;”
Activates single-sweep mode, changes center frequency and span.

50 OUTPUT 718; “TDF P;TRB;”
Formats trace data.

60 FOR N = 1 TO 401
Transfers trace data to array A, one element at a time.

70 ENTER 718 USING “#,K”;A(N)

80 NEXT N
```
90 FOR N = 1 TO 401  
Loop prints out trace B data.

100 PRINT A(N)  
Prints out the results.

110 NEXT N

Description
More information about trace data formats is also included in Chapter 3. The different trace data formats are as follows:

**TDF P**

**Description:** TDF P is the real number format. An example of a trace element returned with the real number format is 10.00 dB. When querying the trace or marker value, the value is returned using the amplitude unit set by AUNITS (for example, watts or dBm).

**Restrictions:** The spectrum analyzer must be in log scale to use TDF P. To send the trace data back to the spectrum analyzer, the data must be converted to measurement units.

**How data is returned:** The following table describes what is transferred when the trace data format is set to P, but the AUNITS are changed. In every case, the trace data transfer is ended by a carriage return, and a line feed with an EOI.

<table>
<thead>
<tr>
<th>AUNITS Setting</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Watts</td>
<td>TDF P; AUNITS W; TRA;</td>
<td>Transfers 401 real values, in watts, with each value separated by a carriage return and a line feed.</td>
</tr>
<tr>
<td>dBm</td>
<td>TDF P; AUNITS DBM; TRA;</td>
<td>Transfers 401 real values, in dBm, with each value separated by a carriage return and a line feed.</td>
</tr>
<tr>
<td>dBmV</td>
<td>TDF P; AUNITS DBMV; TRA;</td>
<td>Transfers 401 real values, in dBmV, with each value separated by a carriage return and a line feed.</td>
</tr>
</tbody>
</table>
Table 10

<table>
<thead>
<tr>
<th>AUNITS Setting</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dBμV</td>
<td>TDF P:AUNITS DBUV;TRA;</td>
<td>Transfers 401 real values, in dBμV, with each value separated by a carriage return and a line feed.</td>
</tr>
<tr>
<td>Volts</td>
<td>TDF P:AUNITS V;TRA;</td>
<td>Transfers 401 real values, in volts, with each value separated by a carriage return and a line feed.</td>
</tr>
</tbody>
</table>

**Example of how data is returned:** For example, if the reference level of the spectrum analyzer is set to -10 dBm, the amplitude scale is set to 10 dB per division. For this example trace A could contain the following data:

- TRA[1] = 8000 (in measurement units), indicating it is at the reference level.
- TRA[2] = 7000 (in measurement units), indicating it is 10 dB below the reference level at -10 dBm.
- TRA[3] through TRA[401] each contain 6000 (in measurement units), indicating that the trace elements 3 through 401 are all 20 dB below the reference level at -20 dB.

Querying trace A with the TDF P format and AUNITS set to DBM returns ASCII character codes for the following:

-10.00,-20.00,-30.00, (-30.00 is repeated 398 times). <CR><LF><EOI>

**TDF A**

**Description:** TDF A is the A-block data format. With the A-block data format, trace data is preceded by “#,” “A,” and a two-byte number (the two byte number indicates the number of trace data bytes). The setting of the MDS command determines whether the trace data is transferred as one or two 8-bit bytes.

**Restrictions:** To use the A-block format for sending data, you must provide the number of data bytes.

**How data is returned:** The following table describes what is transferred when the trace data format is set to A, but the MDS setting is changed.
Table 11

<table>
<thead>
<tr>
<th>MDS Setting</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>TDF A;MDS B;TRA;</td>
<td>Transfers “#A,” the number of bytes of trace data, then the 401 bytes of trace data. Using MDS B “reduces” each trace value into one byte by divid-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ing (DIV) the trace value by 32. The trace data transfer is ended with an EOI.</td>
</tr>
<tr>
<td>Word</td>
<td>TDF A;MDS W;TRA;</td>
<td>Transfers “#A,” the number of bytes of trace data, then 802 bytes of trace data. MDS W uses two bytes per trace element to transfer trace data. The</td>
</tr>
<tr>
<td></td>
<td></td>
<td>first byte contains the trace value divided by (DIV) 256, the second byte contains the remainder (MOD) of that division. The trace data transfer is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ended with an EOI.</td>
</tr>
</tbody>
</table>

Example of how data is returned: For the same example and trace A data that is used in the TDF P description, querying trace A with the TDF A format and MDS set to binary (MDS B) would return the ASCII character codes for the following:

#A(401 div 256)(401 mod 256)(8000 div 32)(7000 div 32)(6000 div 32)(the number for 6000 div 32 is repeated 398 times)<EOI>

Notice that #A is followed by the two bytes that contain the number of trace elements. Because MDS is set to binary, the number of trace elements is 401.

If MDS is set to W, querying trace A with the TDF A format would return the ASCII character codes for the following:

#A(802 div 256)(802 mod 256)(8000 div 256)(8000 mod 256)(7000 div 256)(7000 mod 256)(6000 div 256)(6000 mod 256)(the number for 6000 div 256, then the number for 6000 mod 256 is repeated 398 times)

Notice that #A is followed by the two bytes that contain the number of trace elements. Because MDS is set to W (word), the number of trace elements is 802.

TDF I

Description: TDF I is the I-block data format. With the I-block data format, trace data must be preceded by “#,” and “I.” The setting of the MDS command determines whether the trace data is transferred as one or two 8-bit bytes. Unlike using the A-block format, you do not provide the number of data bytes when sending trace data back to the spectrum analyzer.
Restrictions: This format is not recommended for use with an RS-232 interface.

How data is returned: The following table describes what is transferred when the trace data format is set to I, but the MDS setting is changed.

<table>
<thead>
<tr>
<th>MDS Setting</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>TDF I; MDS B; TRA;</td>
<td>Transfers “#I,” then the 401 bytes of trace data. Using MDS B “reduces” the trace value into 1 byte by dividing (DIV) the trace value by 32. The trace data transfer is ended with an EOI.</td>
</tr>
<tr>
<td>Word</td>
<td>TDF I; MDS W; TRA;</td>
<td>Transfers “#A,” two bytes with length information, then 802 bytes of trace data. MDS W uses two bytes per trace element to transfer trace data. The first byte contains the trace value divided by (DIV) 256, the second byte contains the remainder (MOD) of that division. The trace data transfer is ended with an EOI.</td>
</tr>
</tbody>
</table>

Example of how data is returned: For the same example and trace A data that is used in the TDF P description, querying trace A with the TDF I format and MDS set to binary (MDS B) would return the ASCII character codes for the following:

##I(8000 div 32)(7000 div 32)(6000 div 32)(the number for 6000 div 32 is repeated 398 times)

If MDS is set to W, querying trace A with the TDF I format would return the ASCII character codes for the following:

##I(8000 div 256)(8000 mod 256)(7000 div 32)(7000 mod 256)(6000 div 256)(6000 mod 256)(the number for 6000 div 256, then the number for 6000 mod 256 is repeated 398 times)

TDF B

Description: TDF B enables the binary format. With the binary format, the marker or trace data is transferred as bytes. Of all the trace data formats, TDF B transfers trace data the fastest. The setting of the MDS command determines whether the trace data is transferred as one or two 8-bit bytes.

Restrictions: The TDF B format cannot be used to send data back to the spectrum analyzer (you must use the A-block format to send data back to the spectrum analyzer).
How data is returned: The following table describes what is transferred when the trace data format is set to B, but the MDS setting is changed.

<table>
<thead>
<tr>
<th>MDS Setting</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>TDF B;MDS B; TRA;</td>
<td>Transfers the 401 bytes of trace data. Using MDS B “reduces” the trace value into 1 byte by dividing (DIV) the trace value by 32. The trace data transfer is ended with an EOI.</td>
</tr>
<tr>
<td>Word</td>
<td>TDF B;MDS W; TRA;</td>
<td>Transfers the 802 bytes of trace data. MDS W uses two bytes per trace element to transfer trace data. The first byte contains the trace value divided by (DIV) 256, the second byte contains the remainder (MOD) of that division. The trace data transfer is ended with an EOI.</td>
</tr>
</tbody>
</table>

Example of how data is returned: For the same trace A data that is used in the TDF P description, querying trace A with the TDF B format and MDS set to binary (MDS B) would return the ASCII character codes for the following:

(8000 div 32)(7000 div 32)(6000 div 32)(the number for 6000 div 32 is repeated 398 times)

If MDS is set to W, querying trace A with the TDF B format would return the ASCII character codes for following:

(8000 div 256)(8000 mod 256)(7000 div 32)(7000 mod 256)(6000 div 256)(6000 mod 256)(the number for 6000 div 256, then the number for 6000 mod 256 is repeated 398 times)

TDF M

Description: TDF M is the measurement data format. The measurement data format transfers trace data in measurement units, and the measurement data can range from -32768 to +32767.

Restrictions: TDF M cannot be used to send trace data back to the spectrum analyzer.

How trace data is returned: The following table describes what is transferred when the trace data format is set to M.
### Table 13

#### Trace Data Transfers with TDF M

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDF M;TRA;</td>
<td>Transfers 401 bytes, with each trace value in measurement units. The trace data transfer is ended with a carriage return, a line feed with an EOI.</td>
</tr>
</tbody>
</table>

**Example of how data is returned:** For the same example and trace A data that is used in the TDF P description, querying trace A with the TDF M would return the ASCII character codes for the following:

8000,7000,6000,(6000 repeated 398 times),\(<CR><LF>\)

Refer to “Different Formats for Trace Data Transfers” on page 55 for more information about transferring trace data.

**Query Response**

![Diagram of query response](image)
**TH Threshold**

Clips signal responses below the threshold level.

**Syntax**

```
TH <number> DB
```

**Item** | **Description/Default** | **Range**
--- | --- | ---
*Number* | Any real or integer number. Default unit is dBm. | Range dependent on RL setting

Equivalent Key: **Threshold On Off**.
Preset State: Clip off, positioned one division above bottom graticule line.
Step Increment: One division.
Related Commands: AUTO, DL, MEANTH, MKPK, PEAKS, RL.

**Example**

```
10 OUTPUT 718; "TH UP;"
```

Increases the threshold level.
Description

The threshold level is eight graticule divisions below the top of the screen unless otherwise specified. The threshold level is annotated in reference level units at the lower-left corner of the spectrum analyzer screen. AUTO deactivates clipping. The TH level is used for next peak marker movements (see “MKPK”) and the PEAKS command even if the display clipping is off.

Query Response

```
- number
  output
  termination
```

001
TIMEBASEC Timebase Coarse Adjustment

Allows coarse adjustment of the analyzer timebase.

Syntax

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any integer number</td>
<td>0 to 255</td>
</tr>
</tbody>
</table>

Equivalent Key: **Timebase Coarse**
Step Increment: One division
Related Commands: TIMEBASEF

**NOTE:** The value of the **timebase** coarse adjustment is not reset to the factory value by an instrument preset (IP).

Example

10 OUTPUT 718; "TIMEBASEC 128;"

Sets the **timebase** coarse adjustment to 128.

Description

The TIMEBASEC command makes coarse adjustments to the timebase. The **timebase** is the spectrum analyzer’s internal 10 MHz reference oscillator. The **timebase** coarse value is reset to the factory default at power on.
Query Response
TIMEBASEF Timebase Fine Adjustment

Allows fine adjustment of the analyzer timebase.

Syntax

```
TIMEBASEF number
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any integer number</td>
<td>0 to 255</td>
</tr>
</tbody>
</table>

Equivalent Key: Timebase Fine

Step Increment: One division

Related Commands: TIMEBASEC

**NOTE:**

The value of the timebase fine adjustment is not reset to the factory default by an instrument preset (IP).

**Example**

```
10 OUTPUT 718; "TIMEBASEF 128;"
```

Sets the timebase fine adjustment to 128.

**Description**

The TIMEBASEF command makes fine adjustments to the timebase. The time-base is the spectrum analyzer’s internal 10 MHz reference oscillator. The time-base fine value is reset to the factory default at power on.
Query Response

number → output termination

001
TIMEDATE Time Date

Allows you to set the time and date for the spectrum analyzer real-time clock in the YYYYMMDDHHMMSS format.

Syntax

```
TIMEDATE date &time
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>A number representing the date and time in the YYYYMMDDHHMMSS (24 hour) format</td>
<td>A valid date and time</td>
</tr>
</tbody>
</table>

Equivalent Keys: Set Time, Set Date.
Related Commands: SETDATE, SETTIME, TIMEDSP.

Example

```
10 OUTPUT 718; "TIMEDATE 19971231135501;"
```

Sets the analyzer time and date to 1:55:01 P.M. on 31 December 1997.
Query Response

- Year
- Digit
- Digit
- Month
- Day
- Hour
- Minute
- Second
- Output termination

QTIME0ATE
TIMEDSP Time Display

Enables the display of the time and date on the spectrum analyzer screen.

Syntax

Equivalent Key: Time/Date On Off.
Related Commands: ANNOT, SETDATE, SETTIME, TIMEDATE.

Example

10 OUTPUT 718; "TIMEDSP OFF;"

Query Response
Programming Commands

Programming Command Descriptions

TITLE Title

Activates the screen title mode. The title is displayed above the top graticule and is left justified.

Syntax

```
TITLE title to be displayed
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delimiter</td>
<td>Matching characters marking the beginning and end of the list of analyzer commands</td>
<td>1 @ ^ $ % ;!</td>
</tr>
<tr>
<td>Character</td>
<td>Any valid character</td>
<td>Up to 53 characters</td>
</tr>
</tbody>
</table>

Equivalent Key: Title

Related Commands: IP, SAVE, LOAD

Example

```
10 OUTPUT 718; "TITLE %ADJUST ANTENNA%;"
```

 Displays “ADJUST ANTENNA” on the analyzer screen.

Description

This function writes a message at the top of the spectrum analyzer screen. The full width of the display is available for writing a maximum of 53 characters. However, the marker readout may interfere with the last 26 characters. IP removes the message.

The SAVE command saves the screen title along with the trace in the trace register.
TM Trigger Mode

Selects a trigger mode: free, line, video, or external.

Syntax

```
XTM
```

Equivalent Keys: Trig
Related Commands: DL

Example

```
10 OUTPUT 718;"TM EXT;"
```

Activates the external trigger mode.

Description

The conditions of the four trigger modes are as follows:

- **FREE**: allows the next sweep to start as soon as possible after the last sweep. The functions of TM FREE and **FREE RUN** are identical.
- **VID**: allows the next sweep to start if the trace data rises across a level set by the display line. The functions of TM VID and **VIDEO** are identical.
- **LINE**: allows the next sweep to start when the line voltage passes through zero, going positive. The functions of TM LINE and **LINE** are identical.
EXT allows the next sweep to start when an external voltage level passes through approximately 1.5 V, going positive. The external trigger signal level must be between 0 V and +5 V. Connect the external trigger to the EXT TRIG INPUT. The functions of TM EXT and EXTERNAL are identical.

NOTE: Some spectrum analyzer functions are not performed until the spectrum analyzer is triggered.

Query Response
TOI Third-Order Intermodulation Measurement

Turns on or off the third-order intermodulation (TOI) measurement.

Syntax

Equivalent Key: **TOI On Off**.
Related Commands: **AUNITS, MKPX, TH, TOIR**.

Example

10 **OUTPUT 718;"TOI 1;"**
Turns on the third-order intermodulation measurement.

20 **OUTPUT 718;"TOIR?;"**
Queries TOIR. TOIR contains the results of the third-order intermodulation measurement.

30 **ENTER 718;Toi**
Stores the value of TOIR in the variable Toi.

40 **PRINT "Third-order intermodulation is ",Toi**
Prints the results.

50 **OUTPUT 718;"TOI 0;"**
Turns off the third-order intermodulation measurement.
Description

Setting TOI to 1 turns on the third-order intermodulation measurement. Setting TOI to 0 turns off the third-order intermodulation measurement. When the third-order intermodulation measurement is turned on, the spectrum analyzer first determines that there are four signals on the spectrum analyzer display; the four signals must be the two fundamental signals and two distortion products. All of the signals must be greater than the peak excursion above the threshold. If four valid signals could not be found for the third-order intermodulation measurement, the value of TOIR is -100. If four valid signals could be found, the spectrum analyzer does the following:

1. Finds the four highest on-screen signals. (If the four highest on-screen signals are not the two signals and two distortion products, the TOI measurement cannot be performed.)
2. Determines the spacing between the highest two signals. The highest two signals are tone A and tone B.
3. Verifies that the third and fourth highest signals (distortion A and distortion B) fall above and below tone A and tone B by the frequency difference between tone A and tone B.
4. Measures the levels of the four signals (tone A, tone B, distortion A, and distortion B) and calculates the third-order intermodulation intercept.

The third-order intermodulation intercept is calculated as follows:

\[ TOI = \frac{2 \times Level\_{ToneA} - Level\_{DistortionA} + Level\_{ToneB}}{2} \]

The frequency of the distortion product (Distortion A) is equal to the following:

\[ Frequency\_{DistortionA} = 2 \times Frequency\_{ToneA} - Frequency\_{ToneB} \]

You must query TOIR to determine the value of the higher third-order intermodulation product.

The third-order intermodulation measurement is repeated at the end of every sweep, to update the measurement data, until you turn it off.
**Restrictions Turning** the TOI measurement on turns off the following functions: N dB point measurement (NDBPNT), percent AM (PCTAM), and span zoom (SPZOOM).

You can execute the TOI function by using the command "TOI 1;".

Because TOI is performed at the end of every measurement sweep, you should turn off the third-order intermodulation measurement (set TOI to 0) when you are done with it.

**Query Response**
TOIR Third-Order Intermodulation Response

Returns the intercept point for the highest third-order intermodulation product measured by the third-order intermodulation measurement (TOI).

Syntax

```
TOIR
XTOIR
```

Equivalent Key: TOI OnOff.
Related Commands: AUNITS, MKPX, TH, TOI.

Example

```
10 OUTPUT 718;"TOI 1;"
Turns on the third-order intermodulation measurement.

20 OUTPUT 718;"TOIR?;"
Queries TOIR. TOIR contains the results of the third-order intermodulation measurement.

30 ENTER 718;Toi
Stores the value of TOIR in the variable Toi.

40 PRINT "Third-order intermodulation is ",Toi
Prints the results.

50 OUTPUT 718;"TOI 0;"
Turns off the third-order intermodulation measurement.
```

Description

TOIR returns a -100 if the TOI function has not been turned on, or if four on-screen signals are not valid or are not present. For TOI to perform a third-order intermodulation measurement, there needs to be four signals on the spectrum analyzer display, and all four signals must be greater than the peak excursion above the threshold.
Query Response

TOI amplitude

number

output termination

QTOIR
**TRA/TRB/TRC Trace Data Input and Output**

The TRA/TRB/TRC commands provide a method for returning, storing or changing the 16-bit trace values.

**Syntax**

```
, trace point, TRA number, trace / value, data byte, data byte & EOI
```

Use the same syntax for TRB and TRC as shown for TRA, just substitute TRB or TRC for TRA.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace point number</td>
<td>Integer number that identifies a specific trace point</td>
<td>0 to 400</td>
</tr>
<tr>
<td>Trace value number</td>
<td>Integer number that describes the value of a trace point, in measurement units</td>
<td>Varies depending on display mode and units selected. See the trace data format examples on page 55</td>
</tr>
<tr>
<td>Item</td>
<td>Description/Default</td>
<td>Range</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>MSB length</td>
<td>Most significant byte of a two-byte word that describes the number of bytes transmitted</td>
<td></td>
</tr>
<tr>
<td>LSB length</td>
<td>Least significant byte of a two-byte word that describes the number of bytes transmitted</td>
<td></td>
</tr>
<tr>
<td>Data byte</td>
<td>8-bit byte containing numeric or character data. A typical trace has 401 data bytes.</td>
<td></td>
</tr>
<tr>
<td>Data byte &amp; EOI</td>
<td>8-bit byte containing numeric or character data followed by END character</td>
<td></td>
</tr>
</tbody>
</table>

Related Commands: LOAD, SAVE, DELETE, TDF

**Example**

10 REAL Trace_a(1:401)

Creates a **401-point** trace array.

20 OUTPUT 718;"IP;"

Initializes analyzer.

30 OUTPUT 718;"TDF P;"

Changes the format for real numbers.

40 OUTPUT 718;"SNGLS;"

50 OUTPUT 718;"CF 300MHZ;"

Changes the center frequency.

60 OUTPUT 718;"SP 200MHZ;"

Changes the span.

70 OUTPUT 718;"TS;"

80 OUTPUT 718;"MKPK HI;"

90 OUTPUT 718;"MKCF;"

Moves peak to center of analyzer screen.

100 OUTPUT 718;"TS;"

Updates measurement trace.
Programming Commands
Programming Command Descriptions

110 OUTPUT 718;"TRA?;"
Gets the trace data.

120 ENTER 718;Trace_a(*)
Sends the trace data to the computer.

130 OUTPUT 718;"CONTS;"
Activates continuous sweep mode.

140 END

Description
The TRA command may be used to input integer data to traces. See the information on saving trace data in Chapter 3. Because the lengths of trace A, trace B, and trace C are fixed, there are always 401 bytes transferred during binary input mode.

Trace data that is input in the A-block or the I-block format is treated as measurement units independent of trace data format (TDF) and is preceded by a #A or #I (See the command syntax.) For a 401 point trace, the lsb of the length is 145 (401-256) and the msb of the length is 1 (the integer part of 401/256). Trace data that is input in the A-block, I-block, or B-block format can be sent in byte or word format (MDS). Enter words in measurement units only.

A single trace point can be input by identifying which of the 401 trace points it is, and sending the desired trace value in measurement units.

Query Response
When TRA is queried to output trace data, the output format is specified by the TDF and MDS commands. Because the lengths of trace A, trace B, and trace C are fixed, there are always 802 bytes transferred during binary output mode.

The form of the query response is dependent upon the previously used TDF and MDS commands as follows:
TRSTAT Trace Status

Returns the status of traces A, B, and C: clear write, blank, view, minimum hold, or maximum hold.

Syntax

```
TRSTAT
```

Related Commands: BLANK, CLRW, DET, MINH, TRDSP, VIEW.

Example

This example returns the measurement state of traces A, B, and C.

```plaintext
10 DIM States$[40]
   Declares array for results.
20 OUTPUT 718;"TRSTAT?;"
30 ENTER 718 USING "-K";States$
   Returns the status results to the computer.
40 PRINT States$
   Prints out status of traces.
```
Query Response

Output termination

Programming Commands
Programming Command Descriptions
TS Take Sweep

Starts and completes one full sweep before the next command is executed.

Syntax

```
( TS )
```

Equivalent Key: Single
Related Commands: SNGLS, TM

Example

```
10 OUTPUT 718;"SNGLS;TS;"
```

Activates the single-sweep mode, and takes a sweep.

Description

A take sweep is required for each sweep in the single-sweep mode. TS prevents further input from the interface bus until the sweep is completed to allow synchronization with other instruments.

In the example below, the command sequence does not allow sufficient time for a full sweep of the specified span before VIEW is executed. Therefore, only the span set by the instrument is displayed in trace A.

```
OUTPUT 718;"IP;SNGLS;CF 400MHz;SP 20KHz;VIEW TRA;"
```

A TS command inserted before VIEW makes the spectrum analyzer take one complete sweep before displaying trace A. This allows the spectrum analyzer sufficient time to respond to each command in the sequence.

```
OUTPUT 718;"IP;CF 400MHz;SP 20MHz;TS;VIEW TRA;"
```

TS is recommended before transmission of marker data and before executing marker operations such as peak search. This is because the active marker is repositioned at the end of each sweep. When the spectrum analyzer receives a TS command, it is not ready to receive any more data until one full sweep has been completed. However, when slow sweep speeds are being used, the controller can be programmed to perform computations or address other instruments while the spectrum analyzer completes its sweep.
NOTE: When MKPAUSE or MKSTOP are activated, TS considers the sweep complete when it reaches the active marker.
UP up

Increases the value of the active function by the applicable step size.

Syntax

```
UP
```

Equivalent Key: ‘I

Related Commands: See the list of active functions listed in the description for UP

Example

```
10 OUTPUT 718; "IP;MKN;RB 10KHZ;MKPK NH;UP;"
```

Increases the resolution bandwidth to 30 kHz because MKPK NH does not change the active function.

Description

Before executing UP, be sure that the function to be increased is the active function. For example, the programming example increases the resolution bandwidth, because marker peak (MKPK) is not an active function.

The active functions are AT, CF, CONTRAST, DEMODT, DL, FA, FB, LG, MKA, MKD, MKFCR, MKN, MKPAUSE, MKPX, MKTH, ML, NDB, NRL, NRPOS, PRNTMARGB, PRNTMARGT, PRNTRES, RB, ROFFSET, RL, SETDATE, SETTIME, SP, SRCALC, SRCAT, SRCPOFS, SRCPSWP, SRCPWR, SRCTK, SS, ST, TH, TIMEBASEC, TIMEBASEF, VB, and VBR.
VARIANCE Variance of Trace Amplitudes

Returns the amplitude variance of the specified trace, in measurement units.

**Syntax**

```
VARIANCE source TRA
   TRB
   TRC

? ;
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace range</td>
<td>A segment of trace A, trace B, trace C</td>
<td></td>
</tr>
</tbody>
</table>

Prerequisite Commands: TS when using trace data.
Related Commands: MEAN, RMS, STDEV.

**Example**

10 OUTPUT 718;“IP;”
Initializes analyzer.

20 OUTPUT 718;“SNGLS;TS;”
Activates single-sweep mode.

30 OUTPUT 718;“VARIANCE TRA?;”
Returns variance of trace A to computer.

40 ENTER 718;Number
Stores value in computer variable.
50 DISP Number; "MEASUREMENT UNITS"
Displays the results on computer screen.

Description
Taking the square root of a variance yields the standard deviation value.
The formula to calculate the variance is as follows:

\[
\sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n - 1}
\]

- \( n \) represents the number of data points.
- \( x_i \) represents a data point.
- \( \bar{x} \) represents the mean of data.

Query Response
**VAVG Video Average**

Enables the video-averaging function, which averages trace points to smooth the displayed trace.

**Syntax**

```
VAVG number
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any valid integer. Default is 100.</td>
<td>1 to 16384</td>
</tr>
</tbody>
</table>

Equivalent Key: **Video Average On Off**
Related Commands: AUTO, CLRAVG, IP, SMOOTH

**Example**

```
10 OUTPUT 718;"VAVG 150;"
```

Video averages the trace.

**Description**

Use VAVG to view low-level signals without slowing the sweep time. Video averaging can lower the noise floor by more than a 30 Hz video bandwidth if a large number of sweeps has been specified for averaging. VAVG may also be used to monitor instrument state changes (for example, changing bandwidths, center frequencies) while maintaining a low noise floor.
The active function readout indicates the number of sweeps to be averaged. The default for the number of sweeps is 100 unless otherwise specified. If the number of sweeps is changed, the new number will be remembered when the function is turned off and back on again.

In single sweep mode the selected number of sweeps (N) are taken. Executing “VAVG OFF;” or “VAVG ON;” turns off/on video averaging, but will not trigger a trace when in single sweep mode. A TS command must be sent. After each sweep, the new value of each display point is averaged in with the previously averaged data using the following formula:

\[ A_{avg} = \left(\frac{M-1}{M}\right)A_{prior\ avg} + \left(\frac{1}{M}\right)A_m \]

Where:
- \( A_{avg} \) = new average value
- \( A_{prior\ avg} \) = average from prior sweep
- \( A_m \) = measured value on current sweep
- \( M \) = number of current sweep

In continuous sweep mode the same sequence is followed until \( M=N \). At that point, the sweeps continue rather than stopping. For each new sweep, the current sweep’s measured value divided by \( N \) is added to \( (N-1)/N \) times the prior average, creating a weighted rolling average.

If any measurement parameter such as center frequency, span, amplitude, or \( N \) is changed while video average is on, the video average counter is reset to 0. If the analyzer is in single sweep mode, a new set of sweeps is taken only after a TS or SS command is sent again.

When video averaging is turned on the sample detector mode is automatically switched in. When the averaging is switched off, the detector mode is reset to the previous mode.

**Query Response**

![Diagram](image)
VB Video Bandwidth

Specifies the video bandwidth, which is a post-detection, low-pass filter.

Syntax

Equivalent Key: Video BW Auto Man.
Preset State: 1 MHz.
Step Increment: In a 1, 3, 10 sequence.
Related Commands: AUTO, RB, SP, ST, VBR.

Example

10 OUTPUT 718; "VB 10KHZ;"
Changes the video bandwidth to 10 kHz.
Description

The resolution bandwidth, video bandwidth, and sweep time are normally coupled to the span. Executing VB uncouples video bandwidth from resolution bandwidth (it does nothing to the sweep-time, resolution-bandwidth, and span coupling). Executing AUTO re-couples video bandwidth to the resolution bandwidth.

Frequency values other than the values in the $1, 3, 10$ sequence are rounded to the nearest permissible value.

Query Response

```
  number      output
   ────        ────
             termination
```

001
VBR Video Bandwidth Ratio

The VBR parameter is multiplied by the resolution bandwidth to determine the automatic setting of video bandwidth.

Syntax

<table>
<thead>
<tr>
<th>Item</th>
<th>Description/Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Any valid real number</td>
<td>0 to 3000000</td>
</tr>
</tbody>
</table>

Equivalent Key: **VBW/RBW Ratio**.
Preset State: 0.300.
Step Increment: 1, 3, 10 sequence.
Related Commands: AUTO, RB, SP, VB.

Example

10 OUTPUT 718; "VBR 1;"

Description

Ratio values other than the values in the 1, 3, 10 sequence are rounded to the nearest permissible value.

Query Response
VIEW View Trace

Displays trace A, trace B, or trace C, and stops taking new data into the viewed trace.

Syntax

Equivalent Key: View A, View B, and View C.
Related Commands: BLANK, CLRW, MINH, MXMH.

Example

10 OUTPUT 718; "VIEW TRA;"

Description

In the VIEW mode the trace is not updated. When VIEW is executed, the contents of the trace are stored in display memory.
XCH Exchange

Exchanges the contents of sources 1 and 2.

Syntax

Prerequisite Commands: TS when using trace data
Related Commands: AXB, BXC

Example

10 OUTPUT 718; "XCH TRA,TRB;"
Exchanges the contents of trace A with trace B.
Description

When the source is longer than the destination, the source is truncated to fit. When the source is shorter than the destination, the last element is repeated to fill the destination.
Error Messages
Error Messages
Descriptions

The spectrum analyzer can generate various messages that appear on its screen during operation to indicate a problem.

There are three types of messages: hardware error messages (H), user-created error messages (U), and informational messages (M).

- Hardware error messages indicate the spectrum analyzer hardware is probably broken. Additional troubleshooting should be done to verify the failure.

- User-created error messages appear when the spectrum analyzer is used incorrectly. They are often generated during remote operation (entering programming commands using either a controller or the external keyboard).

- Informational messages provide information indicating the spectrum analyzer’s progress within a specific procedure.

The messages are listed in alphabetical order on the following pages; each message is defined, and its type is indicated by an (H), (U), or (M).

- **ADC-2V fail** Indicates a hardware failure. (H)

- **ADC-Gnd fail** Indicates a hardware failure. (H)

- **ADC-time fail** Indicates a hardware failure. (H)

- **Align IF: gain limited, may recover next alignment** Indicates that an Align Now for the IF Assembly reached a hardware limit. Perform another Align Now IF. If the message persists there may be a hardware failure. Load the default values by pressing **System, Alignments, Load Defaults** and see if the alignment process fixes the problem. (M) or (H)

- **Align IF: symmetry limited, may recover next alignment** Indicates that an Align Now for the IF Assembly reached a hardware limit. Perform another Align Now IF. If the message persists there may be a hardware failure. Load the default values by pressing **System, Alignments, Load Defaults** and see if the alignment process fixes the problem. (M) or (H)

- **Align Now All needed** The instrument requires an Align Now, All. (M)

- **Align RF skipped 50 MHz detected** Indicates that an Auto Align of the RF Assembly was not performed because of an input signal at 50 MHz that is >–25 dBm. The alignment will not work when there is too much input power at 50 MHz. The analyzer will continue to measure properly, but automatic RF alignment will not function. To remove the message, remove the 50 MHz input signal and perform an Align Now RF. (U) or (M)

- **Align RF skipped No align signal** Indicates that an Auto Align of the RF Assembly
was not performed because the instrument’s internal alignment signal was not detected or was invalid. This could be due to a problem with the accuracy or stability of the external 10 MHz reference, if one is present. (U) or (H)

**Alignment Needed** Occurs after a ‘Load Defaults’ of alignment data. A complete alignment must be performed. Press **Align Now, All.** (U) or (M)

**Atten auto set to 15 dB** Indicates that an input signal has been detected which is of sufficient level to force the input attenuator to be auto-coupled at 15 dB. No operator intervention is required. If the signal level is reduced the attenuator will stay at 15 dB.

This overload protection occurs at an input power level of 13 dBm ±7 dB (nominal), when the input attenuation is auto-coupled and set to <15 dB. To return to the original measurement setup, reduce the input signal level and press **Amplitude.** Then press **Attenuation Auto Man** twice, to toggle back to Auto. (U) or (M)

When operating remotely, the status bit should be monitored for this condition to avoid incorrect measurement results.

**Cannot reach N dB points** Indicates that the number of dB specified for the N dB PTS function is greater than the distance of the signal peak from the spectrum analyzer noise floor or peak threshold. (U)

**CMD ERR:-** The specified programming command is not recognized by the spectrum analyzer. (U)

**Configuration error: Front Panel/Display ID not recognized** At power on an error was encountered identifying the Front Panel/Display assembly. (H)

**Configuration error: Option ID not recognized** The hardware for an option was identified in an instrument that does not support it. (H)

**Configuration error: RF Assembly ID not recognized** At power on an error was encountered when identifying the RF Assembly. (H)

**Configuration error: RF-Source Impedance mismatch** A power up check determined that the RF Assembly and optional Tracking Generator Assembly do not have the same impedance. (H)

**Data Memory Full<data type>:-** An attempt was made to save a file of <data type> to memory, however the memory space is full. (U)

**EXT REF** Indicates that the frequency reference is being supplied by an external 10 MHz source. (M)

**EXT Reference in use** An attempt to perform a **Timebase** alignment with an External Reference in use. (U)

**FILE NOT FOUND** Indicates that the specified file could not be loaded into spectrum analyzer memory or purged from memory because the file name cannot be found. (U)

**Freq corr off** Indicates that the Freq Correct selection if Off. (M)
Input is internal Indicates that the 50 MHz osc selection is On. With the 50 MHz oscillator On, the input is routed through an internal signal path. (M)

INVALID AMPCOR: FREQ For the AMPCOR command, the frequency data must be entered in increasing order. See the description for the AMPCOR programming command for more information. (U)

INVALID CHECKSUM: USTATE The user-defined state does not follow the expected format. (U)

INVALID ENTER FORMAT The enter format is not valid. See the appropriate programming command description to determine the correct format. (U)

INVALID FILE NAME <file name> Indicates the specified file name is invalid. A file name is invalid if there is no file name specified or the tile name is too long. See the description for the SAVE programming command for more information. (U)

Invalid flatness data: _ _ _ _ A power up check determined an error in the instrument’s flatness data. The 4-digit code may indicate the type of error. Error codes are described in the spectrum analyzer’s service guide. (H)

INVALID HP-IB ADRS/OPERATION An HP-IB operation was aborted due to an incorrect address or invalid operation. Check that there is only one controller (the spectrum analyzer) connected to the printer. (U)

INVALID HP-IB OPERATION REN TRUE The HP-IB operation is not allowed. (This is usually caused by trying to print when a controller is on the interface bus with the spectrum analyzer.) To print to an HP-IB printer from the spectrum analyzer, you must disconnect any other controllers on the HP-IB. If you are using programming commands to print, you can use an HP BASIC command instead of disconnecting the controller. See the description for the PRINT command for more information. (U)

INVALID ITEM: Indicates an invalid parameter has been used in a programming command. (U)

INVALID OUTPUT FORMAT The output format is not valid. See the appropriate programming command description to determine the correct format. (U)

INVALID RANGE: Start _ Stop Indicates that the first trace element specified for a range of trace elements is larger than the ending trace element. When specifying a trace range the starting element must be less than the ending element. For example, TRA[2,300] is legal but TRA[300,2] is not. (U)

INVALID RS-232/OPERATION An RS-232 operation was aborted due to an invalid operation. (U)

INVALID START INDEX Indicates that the first trace element specified for a range of trace elements is not within the trace range of the specified trace. (U)

INVALID STOP INDEX Indicates that the ending trace element specified for a range of trace elements is not within the trace range of the specified trace. (U)

INVALID STORE DEST: _ _ _ The specified destination field is invalid. (U)
**Invalid Trace:** The specified trace is invalid. (U)

**Invalid Value Parameter:** The specified value parameter is invalid. (U)

**Marker Count Reduce Span** Indicates the resolution bandwidth to span ratio is too small to use the frequency count function. Check the span and resolution bandwidth settings. The acceptable Span/Res BW ratio is 1500. (U)

**Marker Count Widen Res BW** Indicates that the current resolution bandwidth setting is too narrow to use with the marker counter function. (U)

**Meas uncal** The measurement is uncalibrated. Check the sweep time, span, and bandwidth settings, or press **Auto couple**. (U)

**No points defined** Indicates the specified limit line or amplitude correction function cannot be performed because no limit line segments or amplitude correction factors have been defined. (U)

**Overload: Reduce Signal** A signal has been applied to the input connector that caused the overload protection circuitry to engage. The input signal must be reduced. After the signal is reduced, press **Esc**. The change in the instrument’s amplitude state will disengage the overload protection function. Overload protection engages under the following conditions: (U) or (M)

<table>
<thead>
<tr>
<th>Attenuator Setting</th>
<th>Attenuator Coupling</th>
<th>Input Power Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>215 dB</td>
<td>Auto or Man</td>
<td>33 dBm ±3 dB (nominal)</td>
</tr>
<tr>
<td>&lt;15 dB</td>
<td>Man</td>
<td>13 dBm ±7 dB (nominal)</td>
</tr>
</tbody>
</table>

**CAUTION:** Exposing the analyzer to high levels of input power over a prolonged time period can damage the circuitry.

When operating remotely, the status bit should be monitored for this condition to avoid incorrect measurement results.

**Parameter Error:** The specified parameter is not recognized by the spectrum analyzer. See the appropriate programming command description to determine the correct parameters. (U)

**Password required** Indicates that the service function cannot be accessed without the password. (U)

**Power-up selftest failed Boot ROM checksum:** Indicates a test of the bootrom produced a checksum error. (H)

**Power-up selftest failed RAM error:** Indicates a test of RAM produced an error. (H)

**Power-up selftest failed Checksum:** Indicates a test of the instrument ROM produced a checksum error. (H)

**Require 1 signal > Peak Excursion above Peak Threshold** Indicates that the N dBPTS routine cannot locate a signal that is high enough to measure. The signal...
must be greater than the peak excursion above the threshold level to measure. (U)

**Require 3 signals > PEAK EXCURSION above PEAK THRESHOLD** Indicates that the % AM routine cannot locate three signals that are high enough to measure. The signals must be greater than the peak excursion above the threshold level to measure. (U)

**Require 4 signals > PEAK EXCURSION above PEAK THRESHOLD** Indicates that the TOI routine cannot locate four signals that are high enough to measure. The signals must be greater than the peak excursion above the threshold level to measure. (U)

**Required option not installed** Indicates that the function selected requires optional hardware that is not currently available. (U)

**RF LO Unlocked** Indicates that the local oscillator within the instrument’s RF assembly is unlocked. (H)

S in the status area of the display indicates that the service request is active. Service requests are a form of informational message. More information is available in the Programmer’s Guide. (M)

**SAVE LOCK** The spectrum analyzer’s internal memory has been locked. To unlock the memory, press **Internal Lock On Off** so that Off is underlined. For remote operation, use PSTATE OFF. (U)

**Signals do not fit expected % AM pattern** Indicates that the % AM routine cannot perform the percent AM measurement because the on-screen signals do not have the characteristics of a carrier with two sidebands. (U)

**Signals do not fit expected TOI pattern** Indicates that the TOI routine cannot perform the third-order intermodulation measurement because the on-screen signals do not have the characteristics of two signals and two distortion products. (U)

**Source Protected** A signal has been applied to the RF OUT connector which caused the source output protection circuitry to engage. The signal must be reduced. After the signal is reduced, press the **Esc** key to disengage the source output protection circuitry. (U) or (M)

**TABLE FULL** Indicates the upper or lower table of limit lines contains the maximum number of entries allowed. Additional entries to the table are ignored. (U)

**TG LO Unlocked** Indicates that the local oscillator within the instrument’s tracking generator assembly is unlocked. (H)

**TG unleveled** This message can indicate the following: that the source power is set higher or lower than the spectrum analyzer can provide, that the frequency span extends beyond the specified frequency range of the tracking generator, or that the calibration data for the tracking generator is incorrect. See “Stimulus-Response Measurements” in the User’s Guide for more information. (U)

**Too many signal with valid N dB points** Indicates the N dB PTS function has located two or more signals that have amplitudes within the specified dB from the signal peak.
If this happens, you should decrease the span of the spectrum analyzer so that only the signal that you want to measure is displayed. (U)

**Trace A is not available** Indicates that trace A is in the store-blank mode and cannot be used for limit-line testing. Use **CLEARWRITE A** or **VIEW A** to change trace A from the store-blank mode to the clear write mode, and then turn on limit-line testing. (U)
Index

Symbols
% AM commands
  percent AM (PCTAM), 3 19
  percent AM response (PCTAMR), 321

B
A1, 120
A2, 120
A3, 120
A4, 120
A-block format, 63
activate marker, 255
active function, 204
active functions, 186, 436
activate function
decreasing value with DN, 186
address
  changing the HP-IB address, 27
  changing the HP-IB operation, 27
  adjusting the timebase, coarse, 413
  adjusting the timebase, fine, 415
ALIGN, 94, 106, 107, 113, 132
AMPCOR, 94, 106, 110, 113, 135
amplitude correction, 84, 135
amplitude units, 126, 144
analyzer command, 125
analyzer status byte, 73
ANNOT, 94, 106, 114, 138
annotation, 138
APB, 94, 140
AT, 94, 106, 113, 141
ATC, 106
atenuation, 141
AUNIT, 61, 94, 106, 113, 144
AUTO, 94, 106, 114, 146
auto couple, 146
average noise level at the marker
  MKNOISE, 273
AXB, 94, 106, 148
B
B format, 61
B1, 120
B2, 120
B3, 120
B4, 120
backward compatible commands, 120
BASIC
  INPUT command, 45
  REAL command, 46
  baud rate of spectrum analyzer, 150
BAUDRATE, 94, 106, 110, 113, 150
BL, 120
BLANK, 94, 106, 152
blank trace, 152
BML, 94, 106, 153
BREAK
  RS-232 operation, 30
  BRIGHT, 94, 106, 114, 154
BTC, 94, 106, 143, 155
BXC, 94, 106, 156
C
CA, 120
calibration alignment signal, 347
CAT, 94, 158
catalog, 158
center frequency, 41, 42, 161
center frequency step size, 391
CF, 41, 42, 95, 106, 107, 114, 161
  changing the amplitude units
  AUNIT, 144
  changing the HP-IB address, 27
  changing the video bandwidth
  VB, 441
  character & EOI, 125
  characters, 124
  characters and secondary keywords, 126
  clear average, 163
  CLEAR command, 37
  clear status byte, 165
clear to send
  RS-232 handshaking line, 29
clear write, 164
CLRAVG, 95, 163
CLR, 95, 106, 164
CLS, 95, 115, 165
CMDERRQ, 95, 166
COM1 command, 38
  command mnemonic, 124
  command terminators, 124
  commands
  compatible with other spectrum analyzers, 120
  comments in a program, 39
compatible commands, 120
CONFIG, 167
configuration screen, 167
configuring your computer system, 16
connecting a modem
  RS-232 operation, 33
  connecting your spectrum analyzer to a printer, 23
continuous sweep, 170
CONTRAST, 95, 107, 169
controlling the marker counter, 267
controlling trace data with a computer, 51
CONT, 95, 111, 170
copying the source to a destination
  MOV, 301
correction factors on, 171
CORREK, 95, 108, 115, 171
CR, 120
cs, 120
CT, 120
CT/CA, 120
CTS
  RS-232 handshaking line, 29
current units, 126
cv, 120
D
data byte, 125
data byte & EOI, 125
data carrier detect
  RS-232 handshaking line, 29
data entry
disabling, 204
data handling, 46
data lines
  RS-232, 28
data status, 172
data terminal ready
  RS-232 handshaking line, 29
  DATASTAT, 95, 172
date mode, 174
datemode, 95, 107, 113, 174
DCD
  RS-232 handshaking line, 29
default configuration, 175
DEFCONFIG, 95, 107, 113, 175
delete, 114
delete, 95, 107, 176
Index

delete tile, 176
delete limit-line table, 215
delimiter, 125
DEMOD, 95, 106, 107, 178
DEMODT, 95, 107
demodulation, 178
detection mode, 182
determining the trace status, 432
digit, 125
display brightness, 154
display contrast, 169
display line, 184
display messages, 448
displaying the screen title, 420
displaying the time and date, 419
dl., 95, 114, 184
dN, 95, 186
DONE, 96, 187
dotted lines
optional path, 124
down, 186
DSR
RS-232 handshaking line, 29
DTR
RS-232 handshaking line, 29
e
El, 120
E2, 120
E3, 120
E4, 120
einverse video, 206
enable limit line testing, 234
END command, 38
END statement, 65
dend-or-identify, 67
enhancing with variables, 44
enter limit-line segment for frequency,
228
enter limit-line segment for sweep time,
231
EOL, 65, 67
ERASE, 96, 106, 107, 189
ERR, 96, 115
error messages, 448
EX, 120
examples, 49
exchange, 445
exchange trace A and trace B, 148
exchanging traces
XCH, 445
exursion
marker peak, 284
exit config/show system screen, 190
EXITSHOWSYS, 96, 111, 190
external preamplifier gain, 328
external trigger mode, 421
F
FA, 96, 107, 111, 114, 191
FB, 96, 107, 111, 114, 193
F
file functions, 114
file name rules, 39, 358
finding a function, 113
FOFFSET, 96, 107, 114, 195
force service request, 387
free trigger mode, 421
FREF, 96, 115, 197
FREF command, 197
frequency
marker readout, 286
start frequency, 169, 192
frequency command, 41, 42
frequency offset, 195
frequency reference, 197
frequency units, 126
frequency reference, 197
FS, 96, 107, 198
full span, 198
function
alignment, 113
amplitude, 113
bandwidth, 113
configuration, 113
coupling, 114
delete, 114
display, 114
frequency, 114
information, 115
input, 115
limit lines, 115
load, 114
marker, 115
measure, 117
operator entry, 117
output, 115
preset, 117
printing, 117
save, 114
source, 117
span, 118
speaker, 118
sweep, 118
synchronization, 118
trace, 118
trace math, 118
trigger, 119
functional index, 113
G
Get Print, 199
GETPRINT, 96, 199
GRAT, 96, 110, 114, 202
graticule, 202
H
handshaking connection
RS-232 operation, 30
handshaking lines
RS-232, 29
hardware error messages, 448
HD, 96, 107, 114, 204
Hewlett-Packard interface bus, 27
hold data entry, 204
HP 9000 Series 300 technical computers, 18
HP BASIC
CLEAR command, 37
CREATE command, 52
DIM command, 67
END command, 38
ENTER, 67
INPUT command, 44
LOCAL command, 38
REAL command, 44
USING command, 67
HP Vectra personal computer
RS-232 interface, 20
HP-IB, 18
HP-IB interface, 17
HP-IB interface bus, 27
**Index**

<table>
<thead>
<tr>
<th>I-block format</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID, 110, 115, 205</td>
<td></td>
</tr>
<tr>
<td>identify, 205</td>
<td></td>
</tr>
<tr>
<td>impedance</td>
<td></td>
</tr>
<tr>
<td>INZ, 208</td>
<td></td>
</tr>
<tr>
<td>impedance units, 126</td>
<td></td>
</tr>
<tr>
<td>increasing the value</td>
<td></td>
</tr>
<tr>
<td>UP, 436</td>
<td></td>
</tr>
<tr>
<td>informational messages, 448</td>
<td></td>
</tr>
<tr>
<td>INPUT command, 44, 45</td>
<td></td>
</tr>
<tr>
<td>input impedance</td>
<td></td>
</tr>
<tr>
<td>input power</td>
<td></td>
</tr>
<tr>
<td>overload protection, 449, 451</td>
<td></td>
</tr>
<tr>
<td>instrument preset, 38, 210</td>
<td></td>
</tr>
<tr>
<td>interface bus</td>
<td></td>
</tr>
<tr>
<td>RS-232, 28</td>
<td></td>
</tr>
<tr>
<td>interrupt process, 72</td>
<td></td>
</tr>
<tr>
<td>INVERTLCD, 108, 206</td>
<td></td>
</tr>
<tr>
<td>INZ, 96, 108, 208</td>
<td></td>
</tr>
<tr>
<td>IP, 38, 96, 110, 210</td>
<td></td>
</tr>
</tbody>
</table>

**L**

| LO, 121 |     |
| last span, 239 |     |
| LF, 67 |     |
| LG, 97, 110, 113, 213 |     |
| LIMIDEL, 97, 107, 115, 215 |     |
| LIMIDISP, 97, 108, 115, 216 |     |
| LIMIFAIL, 97, 115, 218 |     |
| LIMIFTT, 97, 112, 115, 220 |     |
| LIMIH, 97, 110, 115, 221 |     |
| LIMILINE, 97, 108, 115, 223 |     |
| LIMIREL, 97, 108, 115, 226 |     |
| LIMISEG, 97, 228 |     |
| LIMISEGT, 97, 231 |     |
| limit line |     |
| entry upper/lower, 221 |     |
| limit line display, 216 |     |
| limit lines, 81, 223 |     |
| enable limit line testing, 234 |     |
| enter limit-line segment for frequency, 228 |     |
| enter limit-line segment for sweep time, 231 |     |
| limits failed, 218 |     |
| relative limit lines, 226 |     |
| LIMITEST, 97, 108, 115, 234 |     |
| LIMITLINE, 107 |     |
| limits failed, 218 |     |
| line trigger mode, 421 |     |
| linear scale, 236 |     |
| line-feed, 67 |     |
| LN, 97, 110, 113, 236 |     |
| LOAD, 97, 108, 114, 237 |     |
| loading a file from the memory card, 237 |     |
| LOCAL command, 38 |     |
| locating a function, 113 |     |
| logarithmic scale, 213 |     |
| LSB length, 63 |     |
| lsb length, 125 |     |
| LSPAN, 97, 108, 239 |     |
| M |     |
| M format, 56, 58 |     |
| M1, 121 |     |
| M2, 121 |     |
| M3, 121 |     |
| M4, 121 |     |
| MA, 121 |     |
| marker amplitude, 253 |     |
| marker bandwidth, 257 |     |
| marker counter, 267 |     |
| marker counter resolution, 268 |     |
| marker delta, 260 |     |
| marker frequency, 262 |     |
| marker frequency output, 250 |     |
| marker minimum, 270 |     |
| marker noise, 273 |     |
| marker normal, 271 |     |
| marker off, 276 |     |
| marker pause, 180, 279 |     |
| marker peak, 281 |     |
| marker peak excursion, 284 |     |
| marker peak-to-peak, 283 |     |
| marker position, 277 |     |
| marker readout, 286 |     |
| marker readout in frequency, 286 |     |
| marker step size, 290 |     |
| marker table, 292 |     |
| marker table commands |     |
| marker table (MKTBL), 292 |     |
| marker to center frequency, 259 |     |
| marker to reference level, 288 |     |
| marker to span, 289 |     |
| marker to start frequency, 265 |     |
| marker to stop frequency, 266 |     |
| marker trace, 296 |     |
| marker track, 297 |     |
| maximum hold, 304 |     |
| MC, 121 |     |
| MDS, 61, 98, 115, 240 |     |
| MDS command |     |
| MKF programming example, 79, 263 |     |
| MDU, 98, 115, 243 |     |
| MEAN, 98, 246 |     |
| MEANTH, 98, 248 |     |
| measure signal bandwidth, signal bandwidth, 82 |     |
| measurement data size, 61, 240 |     |
| measurement data units, 243 |     |
| measurement units, 56 |     |
| range, 57 |     |
| measuring harmonic distortion |     |
| HP-IB, 69, 71, 90 |     |
| RS-232, 90, 92 |     |
| measuring noise, 83 |     |
| messages on screen, 448 |     |
| MF, 98, 115, 250 |     |
| MINH, 98, 109, 252 |     |
| minimum hold, 252 |     |
| minimum trace value |     |
| MKMIN, 270 |     |
| mixer level, 299 |     |
| MKA, 51, 98, 253 |     |
| MKACT, 98, 108, 110, 255 |     |
| MKBW, 98, 257 |     |
| MKCF, 85, 98, 109, 259 |     |
| MKD, 98, 108, 260 |     |
| MKF, 51, 98, 262 |     |
| MKFA, 98, 109, 265 |     |
| MKFB, 98, 109, 266 |     |
| MKFC, 98, 107, 108, 267 |     |
| MKFCR, 98, 110, 268 |     |
| MKMIN, 98, 109, 270 |     |
| MKN, 98, 108, 271 |     |
| MKNOISE, 98, 108, 273 |     |
| MKOFF, 98, 108, 276 |     |
| MKP, 98, 277 |     |
| MKPAUSE, 99, 107, 180, 279 |     |
| MKPK, 99, 109, 281 |     |
| MKPK HI, 41, 43 |     |
| MKPP, 99, 109, 283 |     |
Index

MKPX, 99, 109, 284
MKREAD, 99, 108, 111, 286
MKRL, 99, 109, 288
MKSP, 99, 109, 289
MKSS, 99, 109, 290
MKTBL, 292
MKTH, 99, 109, 294
MKTRACE, 99, 108, 296
MKTRACK, 99, 111, 297
monitoring system operation, 72
MOV, 99, 301
move, 301
moving the active marker between traces MKTRACE, 296
MSB length, 63
msb length, 125
MT, 121
MXMH, 99, 108, 304

P
P format, 59
parameter units, 61
party-line bus structure, 27
PCTAM, 100, 106, 319
PCTAMR, 100, 321
peak searching, 41, 43
peak threshold, 294
PEAKS, 100, 323
Percent AM, 3 19
percent AM response, 321
placing a marker MKP, 277
placing a marker on a signal peak MKPP, 281
placing markers on a signal high and low MKPP, 283
power bandwidth, 339
power up time, 342
POWERON, 100, 110, 114, 327
power-on state, 327
PREAMPG, 100, 107, 113, 328
preamplifier gain, 328
preset, 38
preset command, 38
PRINT, 100, 107, 110, 113, 329
print address, 332
print margin top, 335
print resolution, 337
printer margin bottom, 333
Printer port, 331
printing, 23
HP-IB, 25
with a LaserJet Printer (RS-232 only), 33
with a ThinkJet Printer (RS-232 only), 31
laserjet, 23
paintjet, 23
thinkjet, 23

Q
Quick BASIC DIM, 88

R
R1, 121
R2, 121
R3, 121
R4, 121
ratio
setting the video bandwidth ratio, 443

real
RB, 101, 106, 110, 113, 343
RC, 121
RCLS, 102
RCLT, 102
reading trace data, 5 1
REAL command, 44, 46
recommended path, 124
records, 67
reference frequency, 197
reference level, 349
marker to reference level, 288
reference level offset, 353
Index

relative limit lines, 226
repeating syntax element, 124
request to send
RS-232 handshaking line, 29
reserved words, 124
resolution bandwidth, 343
returning or storing trace values, 428
returning the spectrum analyzer to its
former state, 68, 89
returning trace data
changing the trace data format, 404

REV, 102, 110, 115, 345
revision, 345
RFCALIB, 102, 106, 113, 347
RFOFFSET, 113
RL, 102, 110, 113, 349
RMS, 102, 351
ROFFSET, 102, 110, 353
root mean square value, 351
RQS, 102, 115, 355
RS-232
baud rate, 29
connecting a modem, 33
connecting a ThinkJet printer, 31
data lines, 28
handshaking connection for personal
computer, 30
handshaking lines, 29
protocol, 29
setting the baud rate, 34
RS-232 interface, 17, 85
RS-232 interface bus, 28
RTS
RS-232 handshaking line, 29

S
S1, 121
S2, 121
Safety Information, 2
sample programs, 49
SAVE, 102, 109, 110, 114, 358
SAVES, 102
saving analyzer data, 358
saving and loading instrument states, 66
saving and recalling instrument states, 87
saving trace data, 52, 85
screen messages, 448
screen title
specifying the frequency of the marker
MKF, 262
spectrum analyzer state, 67
SPZOOM, 103, 111, 374
SQR, 103, 375
square root, 375
SRCAT, 103, 106, 377
SRCPDFS, 103, 106, 379
SRCPWR, 111
SRCPSPT, 103, 106, 381
SRCPSWP, 103, 110, 383
SRCPWWR, 103, 106, 385
SRQ, 103, 115, 387
SS, 103, 106, 115, 391
ST, 103, 111, 393
standard baud rates, 150
standard deviation of trace amplitudes,
397
start frequency, 191
status bits
  clearing, 165
  status byte query, 395
  status of a trace
    TRSTAT, 432
STB, 104, 115, 395
STDEV, 104, 397
step size
decreasing by the step size with DN,
186
  marker step size, 290
  stop frequency, 193
SUM, 104, 399
sum of squared trace amplitudes, 401
sum of trace amplitudes, 399
SUMSOR, 104, 401
Sweep Couple, 403
sweep time, 393
SWPCL, 104, 111, 403
text elements, 124
text for commands, 124

T
TO, 122
T1, 122
T2, 122
T3, 122
T4, 122
TA, 121

459
Index

take sweep, 434
taking a measurement sweep, 434
TB, 122
TDF, 46, 47, 55, 104, 115, 404
TDF command
  MKF programming example, 79, 263
test program, 16
TH, 104, 111, 411
ThinkJet
Printer mode switch settings, 32
ThinkJet RS-232 switch settings, 32
third-order intermodulation measurement, 423
third-order intermodulation response, 426
threshold, 411
time and date
  displaying the time and date, 419
time date, 417
time display, 419
time units, 126
timebase coarse adjustment, 413
timebase fine adjustment, 415
TIMEBASEC, 104, 111, 413
TIMEBASEF, 104, 111, 415
TIMEDATE, 104, 114, 417
TIMEDESP, 104
TIMEDSP, 111, 114, 419
timeing problems, 434
TITLE, 104, 106, 107, 111, 114, 420
TM, 104, 107, 108, 111, 112, 421
TOI, 104, 423
TOI commands
  third-order intermodulation measurement (TOI), 423
  third-order intermodulation response (TOIR), 426
TOIR, 104, 426
TRA, 51, 104, 111, 115, 428
TRA/TRB/TRC, 428
trace A plus trace B, 140
trace B exchange trace C, 156
trace B minus display line, 153
trace data format, 404
  binary, 67, 88
TDF, 46, 47
trace data formats, 55
trace data input and output, 428
trace data transfers, 55
trace element, 125
trace math, 57
trace mean, 246
trace mean above threshold, 248
trace range, 126
trace status, 432
tracking the signal
  MKTRACK, 297
transfer trace B to trace C, 143, 155
TRB, 104, 111, 115, 428
TRC, 104, 111, 115, 428
TRCA, 85
tigger mode, 421
triggering the spectrum analyzer, 422
TRSTAT, 105, 432
TS, 38, 39, 105, 111, 434
turning off markers
  MKOFF, 276
TV trigger mode, 421
two’s complement, 140
U
  units, 126
  UP, 105, 436
user-defined trace, 126
using amplitude correction, 84
using RS-232 interface, 85
V
  variables, 44
VARIANCE, 105, 437
variance of trace amplitudes, 437
VAVG, 105, 106, 112, 113, 439
VB, 105, 112, 113, 441
VBR, 105, 113, 443
  video average, 439
  video bandwidth, 441
  video bandwidth ratio, 443
  video trigger mode, 421
VIEW, 105, 112, 444
viewing a trace
  VIEW, 444
W
  writing a program, 37

X
  XCH, 105, 445

460