Programming Note

Agilent Technologies
Introductory Programming Guide
For the 8757D/E Scalar Network Analyzer
with the HP Vectra Personal Computer
Using Microsoft® QuickBasic 4.5

Agilent Technologies

Manufacturing Part Number: 08757-90117

Printed in USA

Print Date: July 1992

08757-90117

Hewlett-Packard to Agilent Technologies Transition

This manual may contain references to HP or Hewlett-Packard. Please note that Hewlett-Packard's former test and measurement, semiconductor products and chemical analysis businesses are now part of Agilent Technologies. To reduce potential confusion, the only change to product numbers and names has been in the company name prefix: where a product number/name was HP XXXX the current name/number is now Agilent XXXX. For example, model number HP8648 is now model number Agilent 8648.

Documentation Warranty

DFARS/Restricted Rights Notice

Printing Copies of Documentation from the Web

To print copies of documentation from the Web, download the PDF file from the Agilent web site:

- Go to http://www.agilent.com.
- Enter the document's part number (located on the title page) in the **Quick Search** box.
- Click GO.
- Click on the hyperlink for the document.
- Click the printer icon located in the tool bar.

Contacting Agilent

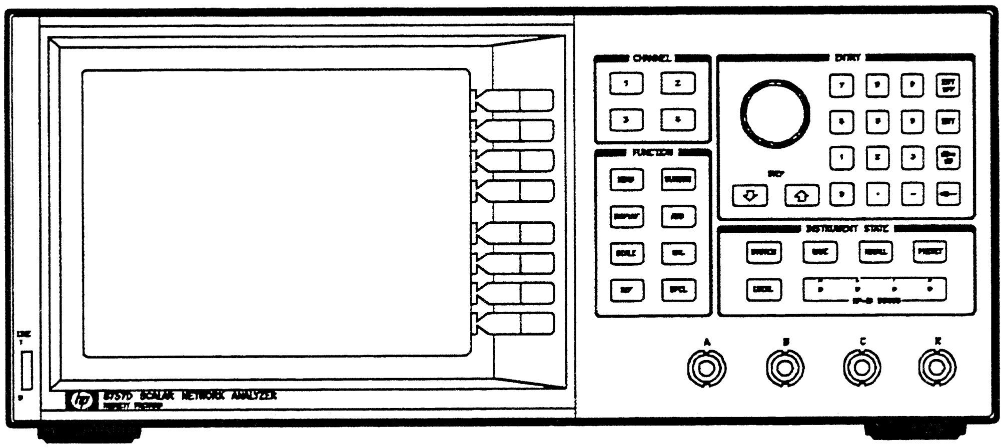
This information supersedes all prior HP contact information.			
Online assistance: www.agilent.com/find/assist			
Americas			
Brazil *(tel)* (+55) 11 3351 7012 *(fax)* (+55) 11 3351 7024	**Canada** *(tel)* 888 447 7378 *(fax) 905 282 6495*	**Mexico** *(tel)* 1 800 254 2440 *(fax)* 1 800 254 4222	**United States** *(tel)* 800 829 4444 *(alt)* (+1) 303 662 3998 *(fax)* 800 829 4433
Asia Pacific and Japan			
Australia (*tel*) 1 800 225 574 (*fax*) 1 800 681 776 (*fax*) 1 800 225 539	**China** (*tel*) 800 810 0508 (*alt*) 800 810 0510 (*fax*) 800 810 0507 (*fax*) 800 810 0362	**Hong Kong** (*tel*) 800 933 229 (*fax*) 800 900 701	**India** (*tel*) 1600 112 626 (*fax*) 1600 112 727 (*fax*) 1600 113 040
Japan (Bench) (*tel*) 0120 32 0119 (*alt*) (+81) 426 56 7799 (*fax*) 0120 01 2144	**Japan (On-Site)** (*tel*) 0120 802 363 (*alt*) (+81) 426 56 7498 (*fax*) (+81) 426 60 8953	**Singapore** (*tel*) 1 800 275 0880 (*fax*) (+65) 6755 1235 (*fax*) (+65) 6755 1214	**South Korea** (*tel*) 080 778 0011 (*fax*) 080 778 0013
Taiwan (*tel*) 0800 047 669 (*fax*) 0800 047 667 (*fax*) 886 3492 0779	**Thailand** (*tel*) 1 800 2758 5822 (*alt*) (+66) 2267 5913 (*fax*) 1 800 656 336	**Malaysia** (*tel*) 1800 880 399 (*fax*) 1800 801 054	
Europe			
Austria (*tel*) 0820 87 44 11* (*fax*) 0820 87 44 22	**Belgium** (*tel*) (+32) (0)2 404 9340 (*alt*) (+32) (0)2 404 9000 (*fax*) (+32) (0)2 404 9395	**Denmark** (*tel*) (+45) 7013 1515 (*alt*) (+45) 7013 7313 (*fax*) (+45) 7013 1555	**Finland** (*tel*) (+358) 10 855 2100 (*fax*) (+358) (0) 10 855 2923
France (*tel*) 0825 010 700* (*alt*) (+33) (0)1 6453 5623 (*fax*) 0825 010 701*	**Germany** (*tel*) 01805 24 6333* (*alt*) 01805 24 6330* (*fax*) 01805 24 6336*	**Ireland** (*tel*) (+353) (0)1 890 924 204 (*alt*) (+353) (0)1 890 924 206 (*fax*)(+353) (0)1 890 924 024	**Israel** (*tel*) (+972) 3 9288 500 (*fax*) (+972) 3 9288 501
Italy (*tel*) (+39) (0)2 9260 8484 (*fax*) (+39) (0)2 9544 1175	**Luxemburg** (*tel*) (+32) (0)2 404 9340 (*alt*) (+32) (0)2 404 9000 (*fax*) (+32) (0)2 404 9395	**Netherlands** (*tel*) (+31) (0)20 547 2111 (*alt*) (+31) (0)20 547 2000 (*fax*) (+31) (0)20 547 2190	**Russia** (*tel*) (+7) 095 797 3963 (*alt*) (+7) 095 797 3900 (*fax*) (+7) 095 797 3901
Spain (*tel*) (+34) 91 631 3300 (*alt*) (+34) 91 631 3000 (*fax*) (+34) 91 631 3301	**Sweden** (*tel*) 0200 88 22 55* (*alt*) (+46) (0)8 5064 8686 (*fax*) 020 120 2266*	**Switzerland (French)** (*tel*) 0800 80 5353 opt. 2* (*alt*) (+33) (0)1 6453 5623 (*fax*) (+41) (0)22 567 5313	**Switzerland (German)** (*tel*) 0800 80 5353 opt. 1* (*alt*) (+49) (0)7031 464 6333 (*fax*) (+41) (0)1 272 7373
Switzerland (Italian) (*tel*) 0800 80 5353 opt. 3* (*alt*) (+39) (0)2 9260 8484 (*fax*) (+41) (0)22 567 5314	**United Kingdom** (*tel*) (+44) (0)7004 666666 (*alt*) (+44) (0)7004 123123 (*fax*) (+44) (0)7004 444555		
(tel) = primary telephone number; *(alt)* = alternate telephone number; *(fax)* = FAX number; * = in country number 8/03/04			

HP–IB Programming Note

HEWLETT PACKARD

Introductory Programming Guide

For the HP 8757D/E scalar network analyzer with
the HP Vectra Personal Computer using Microsoft® QuickBasic 4.5



Introduction

This programming note describes the remote operation of the HP 8757D/E Scalar Network Analyzer with the HP Vectra Personal Computer (or IBM compatible) using the HP 82335A HP–IB Command Library and Microsoft QuickBASIC 4.5. Included in this guide are several short programs that demonstrate the use of the analyzer with HP–IB commands, and a diagram of system connections for remote control.

The HP 8757D/E is a fully programmable analyzer capable of making magnitude–only transmission and reflection measurements over an RF and microwave frequency range of 10 MHz to 100 GHz. When used with an HP–IB computer, the analyzer's front panel may be remotely controlled, along with most softkey functions and some functions accessible only via HP–IB. The analyzer exerts control over a source (HP 8350B, 8340B/41B, or 8360), digital plotter (HP 7440A or 7550A), and printer (HP 2225A ThinkJet, HP 3630A PaintJet, or HP 2227B QuietJet Plus) connected to the 8757 SYSTEM INTERFACE.

This note assumes you are familiar with local (non–remote) operation of the HP 8757D/E. If not, refer to the operating manual. You should also be familiar with the HP Vectra Personal Computer (or compatible), particularly HP–IB operation using the HP 82335A Command Library.

The following are sample programs included in this guide:

- Program 1: Remote, Local, and Local Lockout.
- Program 2: Controlling the Front Panel.
- Program 3: Passthru Mode.
- Program 4: Cursor Operations.
- Program 5: Read a Single Value.
- Program 6: Trace Transfer.
- Program 7: Using the TAKE SWEEP Command.
- Program 8: Programming the Softkeys.
- Program 9: CRT Graphics.
- Program 10: Learning the Instrument State.
- Program 11: Guided Instrument Setup with CRTGraphics.

Reference information

The following texts provide additional information on the HP Interface Bus, analyzer, source, and HP Vectra Personal Computer.

HP 8757D/E literature

- *HP 8757D Operating Manual.*
- *HP 8757C/E Operating Manual.*
- Programming Note: *Quick Reference Guide for the HP 8757D/E Scalar Network Analyzer.*

Microsoft® is a U.S. registered trademark of Microsoft Corp.

Source literature

- Programming Note: *Quick Reference Guide for the HP 8350B Sweep Oscillator.*
- Programming Note: *Quick Reference Guide for the HP 8340B Synthesized Sweeper.*
- *HP 8360 Operating and Programming Reference.*

HP Vectra Personal Computer literature

- *HP 82335A HP–IB Command Library Manual.*
- Microsoft QuickBASIC: BASIC Language Reference Manual.
- *Microsoft QuickBASIC: Learning and Using Microsoft QuickBASIC.*
- *Microsoft QuickBASIC: Programming in BASIC: Selected Topics.*

Equipment required

1 HP 8757D/E Scalar Network Analyzer.

1 HP 8350B Sweeper with plug–in, HP 8340B/41B Synthesized Sweeper, or HP 8360 series Synthesized Sweeper.

1 HP Vectra Personal Computer (or compatible) with Microsoft QuickBASIC 4.5, HP 82335A HP–IB Interface Card, MS–DOS 3.3 or higher, and at least 320K bytes of memory.

1 HP 85027A/B/C/D/E Directional Bridge.

1 HP 11664A/E Detector, HP 85025A/B/D/E Detector, or HP 85037A/B Precision Detector with connector type to match bridge and test device.

1 Shielded open circuit with connector to mate with bridge.

1 Short circuit with connector to mate with bridge.

3 HP 11170C BNC cables, 122 cm (48 in.). (4 are needed with HP 8340B/41B).

2 HP 10833A/B/C/D HP–IB cables.

1 Test device.

Set–up

Connect the instruments as shown in Figure 1. The following procedure sets the HP–IB addresses of the instruments to operate properly with the programs in this guide. If the HP 82335A HP–IB interface card is not installed in the HP Vectra PC, follow the instructions in the *HP 82335A HP–IB Command Library Manual* for installation. Set the interface select code to 7.

1. Turn on the HP 8350B Sweeper. Press **[SHIFT]** **[LCL]**. The FREQUENCY/TIME display shows the current HP–IB address of the source. If it is not 19, press **[1]** **[9]** **[GHz]** to set the address to 19. The HP 8340B or 8341B Synthesized Sweeper operates the same, although the address is displayed in the right–hand display area. For the HP 8360, access the HP–IB menu under the **[SYSTEM MENU]** key. Verify that the address is 19 and programming language is "Analyzer."
2. Power on the HP 8757D/E Scalar Network Analyzer. The current HP–IB address is shown in the active entry area of the CRT. If it is not 16, press **[LOCAL]** *[8757]* **[1]** **[6]** **[ENT]** to set the address to 16.
3. Load Microsoft QuickBASIC by typing "`QB/L QBHPIB`" at the MS–DOS prompt.

Check out procedure

Press **[PRESET]** on the analyzer. If the 8757 SYSTEM INTERFACE is properly connected, and the address of the source correctly set, both the analyzer and the source will perform an instrument preset. If either instrument detects a failure during instrument preset, that instrument displays the error encountered. The operating manual of the source gives instructions to help you interpret the error message. If the analyzer displays an error message, see "In Case of Difficulty."

Programming examples

In the following sections, example programs introduce the HP–IB capabilities of the analyzer. Each example program consists of these sections:

1. A description of the functions exercised.
2. The program listing.
3. An explanation of each program line.
4. Detailed instructions for operating the program.

When you finish all of the example programs, you will have a good idea of the power of the HP 8757D/E when used in an automatic system. Note that line numbers aren't required in Microsoft QuickBASIC but are included in the examples for clarity. Each line number represents a complete statement. No hard line returns are used in the statements although they may appear that way (to improve your ability to read the programs).

In normal programs an error checking line should follow every call to a subprogram:

```
IF PCIB.ERR <>NOERR THEN ERROR
  PCIB.BASERR
```

This statement may be eliminated if this helps program clarity. During error trapping, if an error occurs, the number corresponding to that error is assigned to the variable PCIB.ERR. PCIB.ERR is compared to NO ERR (=0) and then branches to a HP–IB Command Library subprogram for error handling. A message appears on the computer screen stating the error number and type of error.

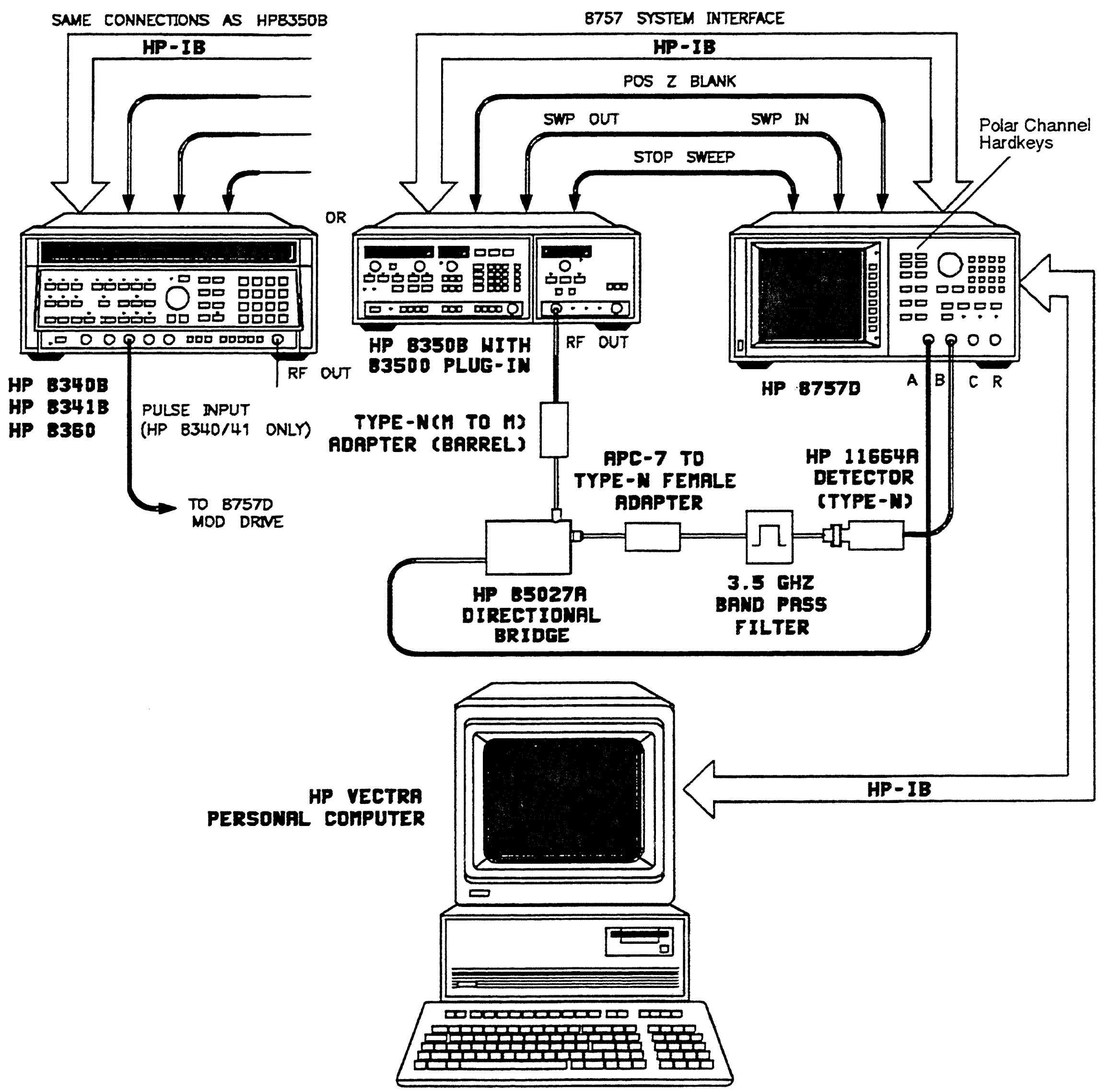


Figure 1. System Connections

Program 1: remote, local, and local lockout

The analyzer may be used with the front panel (local operation) or programmed via HP–IB (remote operation). The programmer of the instrument system has control over the operation of all instruments in the system.

When the computer first addresses an instrument, the instrument is placed in a special remote operating mode, called remote mode. When in remote, the instrument does not respond to its front panel, except for the [LOCAL] key. [LOCAL], when pressed, cancels the remote mode and allows the instrument to be used with its front panel.

The computer can also return the instrument to local operation. To do so, the computer sends a special command that forces the instrument to go to local mode.

Occasionally, the programmer of an automatic system needs to prevent the instrument operator from returning the instrument to local operation (via **[LOCAL]**). When the local lockout function of the computer is used, the instruments are prevented from exiting remote mode, even when [LOCAL] is pressed.

Frequently, the programmer needs to place the instruments connected to the computer into a known state. When preset, the analyzer defaults to the conditions shown below. The instrument preset function operates the same as the front panel **[PRESET]** key on the analyzer and the source. When presetting the analyzer and source, send the PRESET command only to the analyzer. The analyzer will preset the source attached to the 8757 SYSTEM INTERFACE.

HP 8757D/E instrument preset conditions

Channels 1 and 2 on. The channel menu appears in the softkey label area of the CRT.

- Measure power A on channel 1.
- Measure power B on channel 2.
- Measure power C[2] (or B[1]) on channel 3.
- Measure power R on channel 4[1].
- Display measurement data in log magnitude format.
- Scale =20 dB/div.
- Reference level 0 dB for all channels.
- Reference level step size = 20 dB.
- Averaging off.
- Averaging factor =8.
- Cursor off.
- All labels on.
- Channel 1 as the active channel.
- Modulation drive on.
- Number of points =401.
- Detector mode set for AC detection.
- Smoothing set for 5.0% of span (off).
- Cursor format =log magnitude.
- Search value = −3 dB[1].
- Adaptive normalization off[1].
- Temperature compensation on.
- Repeat autozero off.
- Detector offset reset to 0 dB.[1]
- Detector frequency offset[3] off, start and stop = 50 MHz.

Source

- Instrument preset.
- Sweep time set to 200 ms.
- HP 8350B square wave modulation on.
- HP 8340/41 SHIFT PULSE on; RF Output on.
- HP 8360 scalar modulation on; RF output on; ramp sweep mode; analyzer mode.

Plotter

- Abort plot if in progress.
- P1 and P2 scaling points unchanged.
- Selection of plotter pens unchanged.

Printer

- Abort plot if in progress.

Disk drive[1]

- Aborts any data transfers in progress.
- Unit number unchanged.
- Volume number unchanged.
- ASCII or binary mode unchanged.

The following analyzer conditions are not changed during a PRESET (IP) command execution:

- Reference position.
- Trace memory.
- Save/Recall registers.
- HP−IB addresses.
- Request mask.
- Limit lines[1].
- Title.
- Detector offset (HP 8757E only).
- User−defined plot.
- 8757 System Interface control on/off.
- Repeat autozero timer.
- Display intensity.
- Display colors[1].

Program 1 listing

```
10   REM $INCLUDE: 'QBSETUP'
20   CLS
30   ISC& = 7
40   Sna& = 716
50   CALL IOTIMEOUT(ISC&, 10!):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
60   CALL IOABORT(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
70   CALL IOCLEAR(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
80   CALL IOREMOTE(Sna&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
90   GOSUB PAUSE
100  CALL IOREMOTE(Sna&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
110  CALL IOLLOCKOUT(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
120  GOSUB PAUSE
130  CALL IOLOCAL(Sna&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
140  GOSUB PAUSE
150  A$ = "IP" CALL IOOUTPUTS(Sna&, A$, LEN(A$)):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
160  END
170  PAUSE: DO UNTIL INKEY$ = CHR$(13): LOOP
180  RETURN
```

Program 1 explanation

Line 10	Call the QuickBASIC initialization file "QBSETUP", which is the setup program for the MS−DOS HP−IB Command Library. This command must appear before the body of the program whenever calls to the HP−IB Command Library are to be made.
Line 20	Clear the computer CRT.

1. HP 8757D only.
2. HP 8757D Option 001 only.
3. HP 8757D with HP 85037 sseries precision detector only.

Line 30 Assign the interface select code to a variable. This select code is set on the HP 82335A HP–IB interface card.

Line 40 Assign the address of the HP 8757D/E to a variable.

Line 50 Define a system timeout of 10 seconds. Timeout allows recovery from I/O operations that aren't completed in less than 10 seconds. Perform error trapping.

Line 60 Abort any HP–IB transfers. Perform error trapping.

Line 70 Clear the analyzer's HP–IB interface. Perform error trapping.

Line 80 Set the analyzer and source to remote mode. Perform error trapping.

Line 90 Press **[ENTER]** to continue.

Line 100 Set the analyzer and source to remote mode. Perform error trapping.

Line 110 Lock out the **[LOCAL]** key of the analyzer and source. Perform error trapping.

Line 120 Press **[ENTER]** to continue.

Line 130 Set the analyzer and source to local mode. Perform error trapping.

Line 140 Press **[ENTER]** to continue. Perform error trapping.

Line 150 Preset the analyzer and source. Perform error trapping.

Line 160 End program execution.

Line 170 Define a subroutine that waits for the **[ENTER]** key to be pressed.

Line 180 Return from the subroutine.

Running program 1

1. Press **[ALT] [F] [N]** on the computer. This clears the QuickBASIC screen.
2. Type in the program.
3. Press **[ALT] [D] [T]** to trace the program as it runs. Press **[ALT] [R] [S]** on the computer to run the program.
4. When the program pauses, the analyzer is in remote mode. You can verify this by observing the lights in the INSTRUMENT STATE area of the analyzer. The R (remote) and L (listen) lights should be on. Try pressing any key on the analyzer (except **[LOCAL]**). Nothing happens. The source is also in remote mode. Now press **[LOCAL]** and verify that the keys on the analyzer are active. Also, notice the R light went out when you pressed **[LOCAL]**. The source went into local mode along with the analyzer.
5. Press **[ENTER]** on the computer. The analyzer is again in remote mode. This time, however, the **[LOCAL]** key is locked out. Try pressing **[LOCAL]** and the other keys. None of the keys on the analyzer or the source cause any action.
6. Press **[ENTER]** on the computer. All instruments on the HP–IB interface are returned to local mode, including the analyzer and source. Verify that the R light on the analyzer is off and the REM light on the source is off.
7. Press **[ENTER]** on the computer. The analyzer and source are both preset. Note that the computer sent the instrument PRESET command only to the analyzer. The analyzer, in turn, preset the source.

Remember, to preset both the analyzer and the source, you only need to send the instrument PRESET command to the analyzer. Do not send instrument PRESET to the source by way of passthru mode (discussed in program 3).

Program 2: controlling the front panel

All front panel keys and most of the softkeys of the analyzer may be programmed remotely via HP–IB. For example, you can program the scale per division, reference level, and reference position for each channel.

Program 2 listing

```
10   REM $INCLUDE: 'QBSETUP'
20   CLS
30   ISC& = 7
40   Sna& = 716
50   CALL IOTIMEOUT(ISC&, 10!):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
60   CALL IOABORT(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
70   CALL IOCLEAR(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
80   A$ = "IP" GOSUB IOOUTS
90   GOSUB PAUSE
100  A$ = "C1C0C2" GOSUB IOOUTS
110  GOSUB PAUSE
120  A$ = "SD10" GOSUB IOOUTS
130  GOSUB PAUSE
140  A$ = "RL-10" GOSUB IOOUTS
150  GOSUB PAUSE
160  A$ = "RP4" GOSUB IOOUTS
170  GOSUB PAUSE
180  A$ = "IA" GOSUB IOOUTS
190  GOSUB PAUSE
```

```
200  A$ = "C0C1 SD5; RP4; RL-5" GOSUB IOOUTS
210  END
220  PAUSE: DO UNTIL INKEY$ = CHR$(13)
230  LOCATE 25, 1: PRINT "Press ENTER to continue"
     LOOP: CLS
240  RETURN
250  IOOUTS: CALL IOOUTPUTS(Sna&, A$, LEN(A$))
260  IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
270  RETURN
```

Program 2 explanation

Line 10 Call the QuickBASIC initialization file "QBSETUP".

Line 20 Clear the computer CRT.

Line 30 Assign the interface select code to a variable.

Line 40 Assign the HP–IB address of the analyzer to a variable.

Line 50 Define a system timeout of 10 seconds. Perform error trapping.

Line 60 Abort any HP–IB transfers. Perform error trapping.

Line 70 Clear the HP–IB interface of the analyzer. Perform error trapping.

Line 80 Preset the analyzer and source.

Line 90 Press **[ENTER]** to continue.

Line 100 Select channel 1 and turn it off. Turn on channel 2.

Line 110 Press **[ENTER]** to continue.

Line 120 Set the scale per division to 10 dB. No terminator (;) is needed because this is the only command in the statement.

Line 130 Press **[ENTER]** to continue.

Line 140 Set the reference level to —10 dBm. Again, note the absence of a terminator (;).

Line 150 Press **[ENTER]** to continue.

Line 160 Set the reference position line to the center of the screen (graticule 4). No terminator is needed because this is the only command on the line.

Line 170 Press **[ENTER]** to continue.

Line 180 Program channel 2 to measure reflection (input A) instead of transmission (input B).

Line 190 Press **[ENTER]** to continue.

Line 200 There are many commands on one line, with terminators. Turn channel 2 off and channel 1 on (C0C1). Set the scale per division (SD) to 5 dB, reference position line (RP) to the center of the screen, and reference level (RL) to −5 dBm.

Line 210 End execution.

Line 220 Define a subroutine that waits for the **[ENTER]** key to be pressed before returning to program execution.

Line 230 Locate and print a prompt on the CRT. Clear the screen if the loop is terminated.

Line 240 Return from the subroutine.

Line 250 Define a subroutine that outputs commands to the analyzer.

Line 260 Perform error trapping.

Line 270 Return from the subroutine.

Running program 2

1. Press **[ALT] [R] [N]** on the computer. This clears the previous program.
2. Type in this program and press **[ALT] [R] [S]** on the computer.
3. The computer presets the analyzer and source and pauses. Note the settings of channel 1 and 2. Press **[ENTER]**.
4. Channel 1 is turned off. Channel 2 is now the active channel, (notice the highlighted box around the channel 2 mode labels on the analyzer CRT). Press **[ENTER]**.
5. Channel 2 scale per division is now set to 10 dB. It defaulted to 20 dB/div at preset. Press **[ENTER]**.
6. The reference level is set to −10 dBm (it was 0.0 dBm). Press **[ENTER]**.
7. The reference position line is set to the center of the CRT (graticule 4). The top of the CRT is graticule 8 and the bottom is graticule 0. Press **[ENTER]**.
8. Change the measurement to reflection (input A), instead of transmission (input B). At preset, channel 2 defaults to input B. Press **[ENTER]**.
9. In one statement: turn off channel 2, turn on channel 1, set the scale per division to 5 dB, set the reference position line to the center of the CRT, and set the reference level to −5 dBm.

NOTE: The semicolon (;) terminators are needed after any analyzer command that can have a variable length. Extra terminators never hurt, so use them liberally.

Program 3: passthru mode

In normal operation, the system source, digital plotter, printer, and disk drive (HP 8757D only) are connected to the 8757 SYSTEM INTERFACE. This connection allows the analyzer to control and extract information from the other parts of the measurement system. To allow you to control other instruments with the computer, the analyzer has a built–in PASSTHRU command that takes a command from the computer and passes it on to one of the instruments connected to the 8757 SYSTEM INTERFACE.

To initiate passthru mode, tell the analyzer which instrument you wish to command by setting the passthru address. Talk (or listen) to that device, address the analyzer's special passthru HP–IB address (which is different from the analyzer's HP–IB address). While in the passthru mode, the analyzer stops updating its CRT and does not respond to its front panel (because it's in remote mode). To remove the analyzer from passthru mode, address it via HP–IB. While in passthru mode, do not press **[LOCAL]** on the analyzer.

The analyzer's passthru address is calculated from its HP–IB address. If the address of the analyzer is even (such as 16 decimal) then the passthru address is the next larger number (17 decimal). If the address of the analyzer is odd (such as 15 decimal), then the passthru address is the next smaller number (14 decimal). Never set the address of the analyzer so that its address conflicts with one of the instruments connected to the 8757 SYSTEM INTERFACE. For instance, if the source is set to 19 decimal, do not set the analyzer address to 19.

Data can be sent to or received from any instrument on the 8757 SYSTEM INTERFACE via passthru mode. The IOLOCAL, IOREMOTE, and IOTRIGGER HP–IB messages do not pass through the analyzer.

Program 3 listing

```
10   REM $INCLUDE: 'QBSETUP'
20   CLS
30   ISC& = 7
40   Sna& = 716
50   Passthru& = 717
60   CALL IOTIMEOUT(ISC&, 10!):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
70   CALL IOABORT(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
80   CALL IOCLEAR(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
90   A$ = "IP" GOSUB IOOUTS
100  A$ = "PT19" GOSUB IOOUTS
110  A$ = "OPFA" GOSUB IOOUTSP
120  CALL IOENTER(Passthru&, Min.freq):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
130  Min.freq = Min.freq / 1E+09
140  A$ = "OPFB" GOSUB IOOUTSP
150  CALL IOENTER(Passthru&, Max.freq):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
160  Max.freq = Max.freq / 1E+09
170  A$ = "" GOSUB IOOUTS
180  PRINT "Frequency limits:"; Min.freq; "to";
     Max.freq; "GHz"
190  INPUT "Start frequency (GHz)?", Start.freq
200  INPUT "Stop frequency (GHz)?", Stop.freq
210  A$ = "FA" +STR$(Start.freq) + "GZ FB"
     +STR$(Stop.freq) + "GZ" GOSUB IOOUTSP
220  A$ = "" GOSUB IOOUTS
230  END
240  IOOUTS: CALL IOOUTPUTS(Sna&,A$,LEN(A$))
250  IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
260  RETURN
270  IOOUTSP: CALL IOOUTPUTS(Passthru&, A$, LEN(A$))
280  IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
290  RETURN
```

Program 3 explanation

Line 10	Call the QuickBASIC initialization file "QBSETUP".
Line 20	Clear the computer CRT.
Line 30	Assign the interface select code to a variable.
Line 40	Assign the address of the analyzer to a variable. (This is the analyzer's control address).
Line 50	Assign the analyzer's passthru address to a variable. By communicating to this HP–IB address, the computer will control a device connected to the 8757 SYSTEM INTERFACE.
Line 60	Define a system timeout of 10 seconds. Perform error trapping.
Line 70	Abort any HP–IB transfers. Perform error trapping.
Line 80	Clear the HP–IB interface of the analyzer. Perform error trapping.
Line 90	Preset the analyzer and source.
Line 100	Tell the analyzer which device is controlled through the analyzer's passthru address. In this case, the source (device 19).
Line 110	Send a command to the source. Command it to output its current start frequency.
Line 120	Read the start frequency from the source. Perform error trapping.
Line 130	Scale the start frequency to display it in GHz.
Line 140	Command the source to output its current stop frequency.
Line 150	Read the stop frequency from the source. Perform error trapping.
Line 160	Scale the stop frequency to display it in GHz.
Line 170	Exit passthru mode by addressing the analyzer.
Line 180	Print the start and stop frequencies.
Line 190	Get start frequency from user.
Line 200	Get stop frequency from user.

Line 210 Set start and stop frequencies of source to those given by the user.

Line 220 Exit passthru mode by addressing the analyzer.

Line 230 End program execution.

Line 240 Define a subroutine that outputs commands to the analyzer.

Line 250 Perform error trapping.

Line 260 Return from the subroutine.

Line 270 Define a subroutine that outputs commands to the source through the passthru address of the analyzer.

Line 280 Perform error trapping.

Line 290 Return from the subroutine.

Running program 3

1. Clear the computer CRT and type in the program.

2. Press **[ALT]** **[R]** **[S]** on the computer to run the program.

3. The computer presets the analyzer and the source, reads the start and stop frequency of the source, and displays it on the CRT of the computer. At preset, the source defaults to the full frequency range of the plug–in. The values read represent the frequency limits of this plug–in. When the computer stops, it displays the prompt:

```
Start frequency (GHz)?
```

Enter a start frequency in the frequency range of the plug–in and press **[ENTER]**.

4. The computer displays the prompt:

```
Stop frequency (GHz)?
```

Enter a stop frequency in the frequency range of the plug–in (but higher than the start frequency) and press **[ENTER]**.

5. The computer sets the start and stop frequency of the source to those you entered. The analyzer immediately begins sweeping the frequency range you defined.

NOTE: You must address the analyzer after using passthru mode to return it to normal swept operation. Any command can be sent via passthru mode to any instrument on the 8757 SYSTEM INTERFACE, and any data can be read. Service requests and parallel polls do not pass through the analyzer.

Program 4: cursor operations

To enhance the speed and accuracy of measurements, the analyzer contains a built–in cursor that displays the frequency and magnitude of a trace at any given point. To make measurements even more efficient, the cursor may be set to the maximum or minimum point on the trace simply by pressing a softkey. These cursor functions are available via HP–IB commands.

With a computer, the cursor may be turned on and off, its position (0 to n–1, where n is the number of points per trace) set, its value and position read, and set to the maximum or minimum point on the trace. The cursor functions all apply to the active channel (the channel accessed most recently). You have complete control over cursor operations via HP–IB.

Cursor programming is especially useful for measuring parameters like flatness and maximum power, where you are interested in the highest and lowest point on the trace. For measuring parameters such as 3 dB points and other specific points (not a maximum or minimum), it is more efficient to use either the cursor search functions (available on the HP 8757D only) or to read the entire trace and search for the points you need.

Program 4 listing

```
10   REM $INCLUDE: 'QBSETUP'
20   CLS
30   Start.freq = 2
40   Stop.freq = 5
50   ISC& = 7
60   Sna& = 716
70   Passthru& = 717
80   CALL IOTIMEOUT(ISC&, 10!):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
90   CALL IOABORT(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
100  CALL IOCLEAR(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
110  A$ = "IP" GOSUB IOOUTS
120  A$ = "PT19" GOSUB IOOUTS
130  A$ = "FA" +STR$(Start.freq) + "GZ FB"
     +STR$(Stop.freq) + "GZ" GOSUB IOOUTSP
135  A$=""GOSUB IOOUTS
140  A$ = "C2 CXOC" GOSUB IOOUTS
150  Max% = 2
160  Actual% = 0
170  DIM Cursor.vals(Max%)
180  CALL IOENTERA(Sna&, SEG Cursor.vals(0), Max%,
     Actual%):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
190  PRINT "Cursor reads"; Cursor.vals(0); "dB at
     position"; Cursor.vals(1)
200  INPUT "Desired cursor position (0..400)?",
     New.posn
210  A$ = "SC" +STR$(INT(New.posn +.5)): GOSUB
     IOOUTS
220  A$ = "OC" GOSUB IOOUTS
230  CALL IOENTERA(Sna&, SEG Cursor.vals(0), Max%,
     Actual%):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
240  PRINT "Value at position"; Cursor.vals(1); " is
     "; Cursor.vals(0); "dB."
```

```
250  INPUT "Cursor frequency (GHz)?", Cur.freq
260  New.posn = 400 * ((Cur.freq -Start.freq) /
     (Stop.freq -Start.freq))
270  A$ = "SC" +STR$(INT(New.posn +.5)): GOSUB
     IOOUTS
280  A$ = "OC" GOSUB IOOUTS
290  CALL IOENTERA(Sna&, SEG Cursor.vals(0), Max%,
     Actual%):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
300  Cur.freq = Start.freq +(Stop.freq -Start.freq)
     * (Cursor.vals(1) / 400)
310  PRINT "Cursor reads"; Cursor.vals(0); "dB at";
     Cur.freq; "GHz."
320  END
330  IOOUTS: CALL IOOUTPUTS(Sna&, A$ LEN(A$))
340  IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
350  RETURN
360  IOOUTSP: CALL IOOUTPUTS(Passthru&, A$, LEN(A$))
370  IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
380  RETURN
```

Program 4 explanation

Line 10 Call the QuickBASIC initialization file "QBSETUP".

Line 20 Clear the computer CRT.

Line 30 Define the start frequency of the desired sweep in GHz.

Line 40 Define the stop frequency of the desired sweep in GHz.

Line 50 Assign the interface select code to a variable.

Line 60 Assign the address of the analyzer to a variable.

Line 70 Assign the passthru address of the analyzer to a variable.

Line 80 Define a timeout of 10 seconds. Perform error trapping.

Line 90 Abort any HP–IB transfers. Perform error trapping.

Line 100 Clear the HP–IB interface of the analyzer. Perform error trapping.

Line 110 Preset the analyzer and source. This sets the number of points per trace to 401.

Line 120 Tell the analyzer which instrument is controlled through the passthru address (19 is the source).

Line 130 Command the source to set a start frequency of 2 GHz and a stop frequency of 5 GHz.

Line 135 Turn passthru mode off. Allow analyzer to update.

Line 140 Set the cursor to the maximum point on channel 2 and command the analyzer to output the cursor's value and position.

Line 150 Define the maximum number of elements to be read into an array.

Line 160 Define the actual number of elements read.

Line 170 Dimension an array to contain the cursor value and position.

Line 180 Read the value and position of the cursor. Perform error trapping.

Line 190 Print the value and position of the cursor on the computer CRT.

Line 200 Get new cursor position from the user. Input should be between 0 and 400.

Line 210 Set the cursor to the new cursor position chosen by the user.

Line 220 Command the analyzer to output the cursor's value and position.

Line 230 Read the value and position of the cursor at its new position. Perform error trapping.

Line 240 Print the cursor's value and position on the computer CRT.

Line 250 Get new cursor frequency from the user. It must be within the frequency range of the sweep selected.

Line 260 Calculate the position of the cursor from its frequency and the start and stop frequencies of the current measurement.

Line 270 Set the cursor to the desired position.

Line 280 Command the analyzer to output the cursor's value and position.

Line 290 Read the cursor's value and position. Perform error trapping.

Line 300 Calculate the cursor's actual frequency from its position and the start and stop frequencies of the current measurement. You can easily program other start and stop frequencies by following the example in program 3.

Line 310 On the computer CRT, print the value and actual frequency of the cursor.

Line 320 End program execution.

Line 330 Define a subroutine that outputs commands to the analyzer.

Line 340 Perform error trapping.

Line 350 Return from the subroutine.

Line 360 Define a subroutine that outputs commands to the source through the passthru address of the analyzer.

Line 370 Perform error trapping.

Line 380 Return from the subroutine.

Running program 4

1. Clear the computer CRT and type in the program.
2. Press **[ALT] [R] [S]** on the computer.
3. The computer turns on both channels and sets channel 1 to reflection (input A) and channel 2 to transmission (input B). The cursor is positioned to the maximum point on the channel 2 trace and its value and position are read and displayed. At preset, the number of points per trace is 401.
4. The computer displays the prompt:

   `Desired cursor position (0..400)?`

   Type in a number between 0 and 400 and press **[ENTER]**. A position of 0 represents the left side of the analyzer's CRT (lowest frequency) and 400 represents the right side of the CRT (highest frequency). The position is set, and the cursor's value and position is read and printed on the CRT of the computer.
5. The computer stops and displays the prompt:

   `Cursor frequency (GHz)?`

   Enter a frequency within the current start and stop frequencies of the measurement (0.01 to 20 GHz). The nearest cursor position is calculated and set. The value and position of the cursor are read, and the actual cursor frequency is calculated from the cursor's position.

   NOTE: The original desired frequency and the actual cursor frequency are usually different. Because there are only 401 possible cursor positions, some frequencies cannot be set exactly.

To use more points per trace when using the HP 8757D, modify line 110 to be "IP SP801" for 801 points. Then modify the "400" in lines 200, 260, and 300, to "800".

Program 5: read a single value

Measurements often require that a single value be read at a CW frequency, particularly when extremely good frequency accuracy and resolution are required.

The analyzer is able to read and send a single reading of any measurement channel, via HP−IB, to the computer. The OUTPUT VALUE (OV) command operates on the active channel and causes the analyzer to send one reading of measurement data. Even when the analyzer is in normalized mode (MEAS−MEM), the OV command sends the measured, not the normalized, data.

Program 5 listing

```
10   REM $INCLUDE: 'QBSETUP'
20   CLS
30   ISC& = 7
40   Sna& = 716
50   Passthru& = 717
60   CALL IOTIMEOUT(ISC&, 10!):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
70   CALL IOABORT(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
80   CALL IOCLEAR(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
90   A$ = "IP" GOSUB IOOUTS
100  A$ = "PT19" GOSUB IOOUTS
110  A$ = "SW0" GOSUB IOOUTS
120  Freq = 2
130  Freq.step = .1
140  A$ = "CW" +STR$(Freq) + "GZ SF"
     +STR$(Freq.step) + "GZ" GOSUB IOOUTSP
150  A$ = "C1IA" GOSUB IOOUTS
160  FOR I = 1 TO 21
170  A$ = "OV" GOSUB IOOUTS
180  CALL IOENTER(Sna&, Value):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
190  PRINT I; ":", Value; "dB at"; Freq; "GHz"
200  A$ = "CW UP" GOSUB IOOUTSP
210  Freq = Freq +Freq.step
220  NEXT I
230  A$ = "FA2GZ FB4GZ" GOSUB IOOUTSP
240  A$ = "SW1" GOSUB IOOUTS
250  END
260  IOOUTS: CALL IOOUTPUTS(Sna&, A$, LEN(A$))
270  IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
280  RETURN
290  IOOUTSP: CALL IOOUTPUTS(Passthru&, A$, LEN(A$))
300  IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
310  RETURN
```

Program 5 explanation

Line 10	Call the QuickBASIC initialization file "QBSETUP".
Line 20	Clear the computer CRT.
Line 30	Assign the interface select code to a variable.
Line 40	Assign the address of the analyzer to a variable.
Line 50	Assign the passthru address of the analyzer to a variable.
Line 60	Define a system timeout of 10 seconds. Perform error trapping.
Line 70	Abort any HP−IB transfers. Perform error trapping.
Line 80	Clear the HP−IB interface of the analyzer. Perform error trapping.
Line 90	Preset the analyzer and source.
Line 100	Tell the analyzer which instrument is controlled through the passthru address (19 is the source).
Line 110	Put the analyzer in non−swept mode. This step is necessary to read single values. After receiving this command, the analyzer stops updating its display.
Line 120	Define a start frequency for further measurements (in GHz).
Line 130	Define a frequency increment (in GHz).

Line 140 Put the source into CW mode at the start frequency and set its frequency step size to that of the frequency increment.

Line 150 Command the analyzer to measure reflection (input A) on channel 1. This statement also causes the analyzer to exit passthru mode.

Line 160 Make 21 measurements, at equally–spaced CW frequencies.

Line 170 Command the analyzer to send the current reading of channel 1 (the active channel) to the computer. The reading is taken immediately.

Line 180 Read the value. In this instance, no format has been defined so the default format of ASCII is in effect.

Line 190 Print the measurement number, the reading, and the frequency on the computer CRT.

Line 200 Command the source to increment the CW frequency by the step size set earlier (line 110). This is a very fast way of setting a series of equally–spaced frequencies.

Line 210 Increment the variable that contains the current frequency. This variable is only used for printing the current frequency at each iteration of the loop.

Line 220 End of the loop.

Line 230 Command the source to sweep from 2 to 4 GHz. The source exits CW mode and returns to start/stop mode.

Line 240 Command the analyzer to return to swept mode. The analyzer again updates the trace information on the display. This command also exits passthru mode.

Line 250 End program execution.

Line 260 Define a subroutine that outputs commands to the analyzer.

Line 270 Perform error trapping.

Line 280 Return from the subroutine.

Line 290 Define a subroutine that outputs commands to the source through the passthru address of the analyzer.

Line 300 Perform error trapping.

Line 310 Return from the subroutine.

Running program 5

1. Clear the computer CRT and type in the program.
2. Press **[ALT] [R] [S]** on the computer.
3. The source frequency is set immediately to 2 GHz and the computer begins reading reflection (input A) of the analyzer and printing the measurements. After 21 readings, the program ends.

Program 6: trace transfer

One feature that sets the HP 8757D/E apart is its ability to transfer an entire measurement trace to a computer at very high speed. A complete, high–resolution (0.01 dB) 401–point measurement can be sent to the computer in 35 milliseconds (binary format) or 800 milliseconds (ASCII format). Transfer time will be less for fewer points per trace, and greater for more points per trace.

The analyzer gives you complete flexibility when reading measurement traces via HP–IB. You can read from the active channel and you can read the stored memory trace, the current measurement trace, or the normalized trace (measurement–minus–memory). In addition, the memory trace can be written back to the analyzer, allowing you to save and restore calibration traces via HP–IB.

With trace transfer measurements, some frequency resolution is sacrificed for measurement speed. The number of points per trace can be programmed to control the resolution across the frequency range being swept. If you are measuring a device that changes very rapidly with frequency, it is possible to miss very narrowband responses that occur between measurement points if the resolution is low. For these cases, the measurement should be made at a higher resolution. The trace transfer method of measurement is much faster than CW point–by–point measurements.

Program 6 listing

```
10   REM $INCLUDE: 'QBSETUP'
20   CLS
30   ISC& = 7
40   Sna& = 716
50   DIM Ascii.dat(1 TO 401),Binary.dat%(1 TO 401)
60   Max1% = 401
70   Max2% = 2 * Max1%
80   CALL IOTIMEOUT(ISC&, 10!):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
90   CALL IOABORT(ISC&):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
100  CALL IOCLEAR(ISC&):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
110  A$ = "IP" GOSUB IOOUTS
120  A$ = "C1IA;C2IB" GOSUB IOOUTS
130  Start = TIMER
140  Stopped = TIMER
150  DO UNTIL ((Stopped - Start) >2)
160  Stopped = TIMER
170  LOOP
180  A$ = "FD2;C1OD" GOSUB IOOUTS
190  Actual% = 0
200  CALL IOENTERA(Sna&, SEG Ascii.dat(1), Max1%,
     Actual%):IF PCIB.ERR <> NOERR THEN ERROR
     PCIB.BASERR
210  CALL IOEOL(ISC&, CHR$(13) + CHR$(10), 0): IF
     PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
220  A$ = "C1WM" GOSUB IOOUTS
```

```
230 CALL IOEOL(ISC&, CHR$(13) + CHR$(10), 2): IF
    PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
240 CALL IOOUTPUTA(Sna&, SEG Ascii.dat(1), Max1%):
    IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
250 A$ = "C1MY" GOSUB IOOUTS
260 DO UNTIL INKEY$ = CHR$(13): LOCATE 25, 1
270 PRINT "Press ENTER to continue" LOOP: CLS
280 A$ = "C1C0;C2MY" GOSUB IOOUTS
290 Actual% = 0
300 A$ = "FD3;C2OD" GOSUB IOOUTS
310 CALL IOENTERB(Sna&, SEG Binary.dat%(1), Max2%,
    Actual%, 1):
    IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
320 CALL IOEOL(ISC&, CHR$(13) + CHR$(10), 0):
    IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
330 A$ = "C2WM" GOSUB IOOUTS
340 CALL IOEOL(ISC&, CHR$(13) + CHR$(10), 2): IF
    PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
350 CALL IOOUTPUTB(Sna&, SEG Binary.dat%(1), Max2%,
    1):
    IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
360 DO UNTIL INKEY$ = CHR$(13): LOCATE 25, 1
370 PRINT "Press ENTER to continue" LOOP: CLS
380 FOR I% = 1 TO Max1%
390 Binary.dat%(I%) = (I% MOD 100)
400 NEXT I%
410 A$ = "C2C0;C1MY" GOSUB IOOUTS
420 CALL IOEOL(ISC&, CHR$(13) + CHR$(10), 0):
    IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
430 A$ = "FD3;C1WM" GOSUB IOOUTS
440 CALL IOOUTPUTB(Sna&, SEG Binary.dat%(1), Max2%,
    1): IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
450 CALL IOEOL(ISC&, CHR$(13) + CHR$(10), 2):
    IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
460 A$ = "AS" GOSUB IOOUTS
470 END
480 IOOUTS: CALL IOOUTPUTS(Sna&, A$, LEN(A$))
490 IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
500 RETURN
```

Program 6 explanation

Line 10 Call the QuickBASIC initialization file "QBSETUP".

Line 20 Clear the computer CRT.

Line 30 Assign the interface select code to a variable.

Line 40 Assign the address of the analyzer to a variable.

Line 50 Dimension an array to hold a trace of 401 points in ASCII format. Dimension a second array to hold another 401 points trace in binary format.

Line 60 Create a variable based on the number of points per sweep on the analyzer. By using a variable here it helps to make the program easily adaptable to different numbers of trace points.

Line 70 Create a variable to define the number of bytes used in the binary trace transfer.

Line 80 Define a system timeout of 10 seconds. Perform error trapping.

Line 90 Abort any HP–IB transfers. Perform error trapping.

Line 100 Clear the HP–IB interface of the analyzer. Perform error trapping.

Line 110 Preset the analyzer and the source. This sets the number of points per trace to 401.

Line 120 Set channel 1 to reflection (input A) and channel 2 to transmission (input B).

Line 130 Set a start time using the TIMER function in QuickBASIC.

Line 140 Set an initial stop time to be compared to the start time.

Line 150 Loop until 2 seconds have elapsed from the start time.

Line 160 Update the time.

Line 170 End of the 2 second loop.

Line 180 Set the data format to Extended ASCII and command the analyzer to output the channel 1 measurement data.

Line 190 Initialize the variable specifying the number of elements actually read into the array.

Line 200 Read the measurement trace data from channel 1. Perform error trapping.

Line 210 Disable the end–of–line string (carriage return/linefeed) that is sent after any IOOUTPUT command.

Line 220 Command the analyzer to input data into the trace memory of channel 1.

Line 230 Enable the end–of–line string (carriage return/linefeed) that is sent after any IOOUTPUT command.

Line 240 Write the measured trace data back to the trace memory of channel 1. Reading the measurement trace and storing it back into trace memory is equivalent to executing the MEAS–MEM function (HP–IB command SM). Perform error trapping.

Line 250 Command channel 1 to display the trace memory data.

Line 260 Press **[ENTER]** to continue.

Line 270 Print a message on the computer's CRT notifying the user that the computer is waiting for a key to be pressed before continuing.

Line 280 Turn channel 1 off and channel 2 on. Command the analyzer to display the trace memory from channel 2.

Line 290 Initialize the variable specifying the number of elements actually read into the array.

Line 300 Set the data format to PC binary format. Command the analyzer to output its channel 2 measurement trace data.

Line 310 Read the binary measurement data from channel 2. Perform error trapping.

Line 320 Disable the end–of–line string (carriage return/linefeed) that is sent after any IOOUTPUT command.

Line 330 Command the analyzer to input data into the trace memory of channel 2.

Line 340 Enable the end–of–line string (carriage return/linefeed) that is sent after any IOOUTPUT command.

Line 350 Write the binary data array back to the trace memory of channel 2. Perform error trapping.

Line 360 Press **[ENTER]** to continue.

Line 370 Print a message on the computer's CRT notifying the user that the computer is waiting for a key to be pressed before continuing.Line 380 Set up a loop to create 401 measurement points.

Line 390 Calculate some arbitrary function and fill the binary data array. This function has no particular meaning, but represents some special calibration data (such as an open/short average).

Line 400 End of the loop.

Line 410 Turn channel 2 off and display the channel 1 trace memory.

Line 420 Disable the end–of–line string (carriage return/linefeed) that is sent after any IOOUTPUT command.

Line 430 Command the analyzer to input data into the trace memory of channel 2.

Line 440 Write the binary data array to the trace memory of channel 2. Perform error trapping.

Line 450 Enable the end–of–line string (carriage return/linefeed) that is sent after any IOOUTPUT command.

Line 460 Autoscale the display on channel 1.

Line 470 End program execution.

Line 480 Define a subroutine that outputs commands to the analyzer.

Line 490 Perform error trapping.

Line 500 Return from the subroutine.

Running program 6

1. Clear the computer CRT and type in the program.
2. Press **[ALT] [R] [S]** on the computer.
3. Watching the analyzer CRT, you will see `DATA DUMP TO HP–IB` when it begins sending trace data to the computer, and `DATA DUMP TO TRACE MEMORY` when the computer sends data back.
4. Watching the analyzer CRT, press **[ENTER]** on the computer. The computer again reads and writes a trace of data. The analyzer displays the same messages. This time the transfer occurs much more rapidly. A binary transfer takes about 35 milliseconds to be completed while an ASCII trace transfer requires about 800 milliseconds each way.
5. Press **[ENTER]** on the computer. The computer calculates an arbitrary function and sends it to trace memory of the analyzer, where it is autoscaled and displayed. This function has no significance. It represents a special calibration trace, such as a short/open average. With a computer, the analyzer measurement system can be calibrated over several different frequency ranges and changed from one to another very quickly, with–out recalibration.

If you wish to transfer a higher resolution trace with the HP 8757D, modify line 110 to be "IP SP801" for 801 points. Then modify "401" in lines 50 and 60 to "801."

Program 7: using the TAKE SWEEP command

To make measurements as quickly and efficiently as possible, it is often necessary to synchronize the source with the analyzer. The TAKE SWEEP command gives the analyzer the ability to command the source to make a specified number of complete sweeps (1 to 255). This command is especially useful when using the trace transfer method of reading data from the analyzer.

To use the TAKE SWEEP command, place the analyzer in non–swept mode (SW0). Then give the TAKE SWEEP command with the number of sweeps desired (TSd). At the end of the specified number of sweeps, the analyzer informs the computer of the completion of this operation by setting a bit in its status byte.

The computer can detect this event in two ways:

- Monitor the status byte continuously until the bit is set (polling).
- Let the analyzer generate a service request (SRQ) and interrupt the computer.

Table 1 is a diagram of the status bytes of the analyzer. It shows all of the bits that can be used to either monitor or interrupt the computer. In this program, bit 4 (decimal value 16) is used to signal "operation complete" (all of the sweeps specified by the TAKE SWEEP command have been completed.)

When you follow the take sweep command with an output statement, such as OUTPUT DATA (OD), the data is sent immediately, not after the instructed number of sweeps. The two approaches mentioned overcome this by letting us send the data at the end of the specified number of sweeps, not immediately. A third approach is to use the sweep hold mode (SW2) instead of the non–swept mode (SW0). In this mode the analyzer will prevent any HP–IB operations until the completion of the TAKE SWEEP command.

Program 7 listing

```
10   REM $INCLUDE: 'QBSETUP'
20   CLS
30   DIM ASCIIDAT(0 TO 400)
40   Isc& = 7
50   Sna& = 716
60   Passthru& = 717
70   CALL IOTIMEOUT(Isc&, 10):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
80   CALL IOABORT(Isc&):IF PCIB.ERR <> NOERR THEN
     ERROR PCIB.BASERR
90   CALL IOCLEAR(Isc&):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
100  A$ = "IP" GOSUB IOOUTS
110  A$ = "PT19" GOSUB IOOUTS
120  A$ = "ST250MS" CALL IOOUTPUTS(Passthru&, A$,
     LEN(A$)):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
130  A$ = "C2C0 IB" GOSUB IOOUTS
140  A$ = "SW0;CS;RM16;" GOSUB IOOUTS
150  Stat% = 0
160  A$ = "TS10;" GOSUB IOOUTS
170  DO UNTIL ((Stat% MOD 32) > 15)
180  CALL IOSPOLL(Sna&, Stat%):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
190  LOOP
200  A$ = "C1OD" GOSUB IOOUTS
210  Max% = 401
220  Actual% = 0
230  CALL IOENTERA(Sna&, SEG ASCIIDAT(0), Max%,
     Actual%):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
240  A$ = "SW1" GOSUB IOOUTS
250  DO UNTIL INKEY$ = CHR$(13): LOCATE 25, 1
260  PRINT "Press ENTER to continue" LOOP: CLS
270  PEN OFF
280  A$ = "SW0;CS;RM16;" GOSUB IOOUTS
290  CALL IOPEN(Isc&, 0)
300  ON PEN GOSUB Srq.recv
310  PEN ON
320  A$ = "TS10;" GOSUB IOOUTS
330  Wait.srq: '
340  IF Intr.bit% = 0 THEN GOTO Wait.srq
350  PEN OFF
360  END
370  Srq.recv: '
380  CALL IOSPOLL(Sna&, Intr.bit%):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
390  A$ = "RM0" GOSUB IOOUTS
400  A$ = "C1OD" GOSUB IOOUTS
410  Max% = 401
420  Actual% = 0
430  CALL IOENTERA(Sna&, SEG ASCIIDAT(0), Max%,
     Actual%):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
440  A$ = "SW1" GOSUB IOOUTS
450  RETURN
460  IOOUTS: CALL IOOUTPUTS(Sna&, A$, LEN(A$))
470  IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
480  RETURN
```

Program 7 explanation

Line 10	Call the QuickBASIC initialization file "QBSETUP".
Line 20	Clear the computer CRT.
Line 30	Dimension an array large enough to hold a trace of data (401 points).
Line 40	Assign the interface select code to a variable.
Line 50	Assign the address of the analyzer to a variable.
Line 60	Assign the passthru address of the analyzer to a variable.
Line 70	Define a system timeout of 10 seconds. Perform error trapping.
Line 80	Abort any HP–IB transfers. Perform error trapping.
Line 90	Clear the HP–IB interface of the analyzer. Perform error trapping.
Line 100	Preset the analyzer and source.
Line 110	Tell the analyzer which device is controlled through the passthru address. Address 19 belongs to the source.
Line 120	Set the source to 250 milliseconds per sweep. Perform error trapping.
Line 130	Turn off channel 2 of the analyzer and select transmission (input B) for display on channel 1.
Line 140	Put the analyzer into non–swept mode. Clear the status register of the analyzer. Set the request mask to 16 (bit 4) so that the analyzer will set bit 4 (operation complete) at the completion of the TAKE SWEEP command. Table 1 has a description of all bits in the status bytes.
Line 150	Assign the status variable initially to zero.
Line 160	Command the analyzer to take 10 sweeps.
Line 170	Wait for the 10 sweeps to completed by testing the status byte to see if bit 4 is set. Remain in the loop until bit 4 is set.
Line 180	Read the analyzer status byte. Perform error trapping.
Line 190	End of the loop.
Line 200	Command the analyzer to output the channel 1 trace data.
Line 210	Define the maximum number of elements to be read into an array.

Line 220 Define the actual number of elements read.

Line 230 Read the trace data. Perform error trapping.

Line 240 Return the analyzer to swept mode. The display now updates continuously.

Line 250 Wait for the [ENTER] key to be pressed. Locate where the prompt will be displayed on the CRT.

Line 260 Print a prompt on the CRT. Clear the screen if the loop was terminated.

Line 270 HP−IB service requests are implemented as lightpen events. This statement disables any lightpen event trapping.

Line 280 Put the analyzer into non−swept mode. Clear the status register of the analyzer. Set the request mask to 16 (bit 4) so that the analyzer will set bit 4 (operation complete) at the completion of the TAKE SWEEP command. This is the same as in line 140 except that we will look for interrupts this time.

Line 290 Enable the HP−IB interface to detect HP−IB service requests and process the interrupt as ON PEN events.

Line 300 Line label for routine that is executed when an interrupt is detected.

Line 310 Enable HP−IB service request interrupt event trapping.

Line 320 Command the analyzer to take 10 sweeps.

Line 330 Line label for loop that waits for an interrupt.

Line 340 If a service request was not detected, continue looping.

Line 350 Disable HP−IB service request interrupt event trapping.

Line 360 End program execution.

Line 370 Line label for routine that services the interrupts.

Line 380 Serial poll the analyzer. Reading the status byte of the analyzer clears the SRQ. The CLEAR STATUS (CS) command could also be used. Perform error trapping.

Line 390 Disable interrupt generation from the analyzer.

Line 400 Command the analyzer to output the channel 1 trace data.

Line 410 Define the maximum number of elements to be read into an array.

Line 420 Define the actual number of elements read.

Line 430 Read the trace data. Perform error trapping.

Line 440 Return the analyzer to swept mode. The display now updates continuously.

Line 450 Return from subroutine.

Line 460 Define a subroutine that outputs commands to the analyzer.

Line 470 Perform error trapping.

Line 480 Return from the subroutine.

Running program 7

1. Clear the computer CRT and type in the program.
2. Press **[ALT] [R] [S]** on the computer.
3. The computer first presets the analyzer and source. It then sets the source to 250 milliseconds per sweep, and sets the analyzer to display transmission on channel 1.
4. The computer commands the analyzer to take 10 sweeps and polls the analyzer status byte to determine when they were completed. The computer reads a trace from the analyzer. Just before the trace is sent, you should see the display "freeze" as the TAKE SWEEP command is completed.
5. Press **[ENTER]**, and the computer again tells the analyzer to take 10 sweeps. This time the computer receives an interrupt after the last sweep. The computer sits in a loop (lines 330 and 340) and waits until the analyzer signals completion of the TAKE SWEEP command. In this segment of the program, you should not see the display "freeze" at all. Immediately after it receives the interrupt, the computer puts the analyzer back into swept mode. This method of sensing the end of a TAKE SWEEP command via an interrupt is more time−efficient than the polling method previously used because the computer can be doing something else during the 10 sweeps.

To use the sweep hold mode, modify line 140 to "SW2" (instead of "SW0;CS;RM16;") and delete lines 150, 170, 180, and 190. The program will wait at line 200 until the 10 sweeps are completed. Whenever practical, use the service request interrupt to sense the end of a TAKE SWEEP command. In fact, you can use the time to do plotting or printing of data, instead of sitting in a loop. Service requests are useful for other events, as demonstrated by the next program.

Table 1. HP 8757D/E Status Byte Descriptions

STATUS BYTE (#1)								
BIT #	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Decimal Value	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Function	N/A	Request Service (SRQ)	SRQ on HP–IB Syntax Error	SRQ on Operation Complete (Sweep, Plot or Print)	SRQ on Softkey Only Pressed	SRQ on Change in Extended Status Byte	SRQ on Numeric Entry Completed (HP–IB or Front Panel)	SRQ on Any Front Panel Key Pressed
EXTENDED STATUS BYTE (#2)								
BIT #	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Decimal Value	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Function	N/A	SRQ on Detector Uncal	SRQ on Front Panel Preset or Power–on	SRQ on Limit Test Failed[1]	SRQ on Action Requested not possible	SRQ on Knob Activity	SRQ on Operation Failed[1]	SRQ on Self Test Failure

1. HP 8757D only.

Program 8: programming the softkeys

The HP 8757D/E has eight screen–labeled softkeys that make your measurements faster and easier. Under HP–IB control, you can re–label the softkeys with any annotation and sense when they are pressed.

Use the softkeys to branch to special measurement programs. By making full use of the softkeys, your automatic system may not need a normal computer keyboard at all, making it as easy to use as a manual instrument.

Program 8 listing

```
10   REM $INCLUDE: 'QBSETUP'
20   CLS
30   Isc& = 7
40   Sna& = 716
50   CALL IOTIMEOUT(Isc&, 10!):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
60   CALL IOABORT(Isc&):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
70   CALL IOCLEAR(Isc&):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
80   PEN OFF
90   A$ = "IP" GOSUB IOOUTS
100  A$ = "CS RM8" GOSUB IOOUTS
110  A$ = "WK1 CAL1" GOSUB IOOUTS
120  A$ = "WK2 TEST1" GOSUB IOOUTS
130  A$ = "WK3 CAL2" GOSUB IOOUTS
140  A$ = "WK4 TEST2" GOSUB IOOUTS
150  A$ = "WK8 ABORT" GOSUB IOOUTS
160  PRINT "SOFT KEYS LOADED"
170  CALL IOPEN(Isc&, 0)
180  ON PEN GOSUB Srq.recv
190  PEN ON
200 Wait.srq: '
210  IF Keycode <> 41 THEN GOTO Wait.srq
220  PEN OFF
230  END
240  Srq.recv: '
250  CALL IOSPOLL(Sna&, Intr.bit%):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
260  A$ = "OK" GOSUB IOOUTS
270  CALL IOENTER(Sna&, Keycode)
280  SELECT CASE Keycode: CASE 32
290  CLS : LOCATE 12, 29: PRINT "Calibration #1"
300  CASE 8
310  CLS : LOCATE 12, 29: PRINT "Test #1"
320  CASE 0
330  CLS : LOCATE 12, 29: PRINT "Calibration #2"
340  CASE 16
350  CLS : LOCATE 12, 29: PRINT "Test #2"
360  CASE 41
370  CLS : LOCATE 12, 29: PRINT "Abort"
380  CASE ELSE
390  CLS : LOCATE 12, 29: PRINT "***Undefined***"
400  END SELECT
410  RETURN
420  IOOUTS: CALL IOOUTPUTS(Sna&, A$, LEN(A$))
430  IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
440  RETURN
```

Program 8 explanation

Line 10 Call the QuickBASIC initialization file "QBSETUP".

Line 20 Clear the computer CRT.

Line 30 Assign the interface select code to a variable.

Line 40 Assign the address of the analyzer to a variable.

Line 50 Define a system timeout of 10 seconds. Perform error trapping.

Line 60 Abort any HP–IB transfers. Perform error trapping.

Line 70 Clear the HP–IB interface of the analyzer. Perform error trapping.

Line 80 HP–IB service requests are implemented as lightpen events. This statement disables any lightpen event trapping.

Line 90 Preset the analyzer and source.

Line 100 Set the request mask to 8 (bit 3). See table 1 for the description of the status bytes.

Line 110 Label softkey 1 with "CAL 1". Softkey 1 is the softkey at the top of the CRT.

Line 120 Label softkey 2 with "TEST 1".

Line 130 Label softkey 3 with "CAL 2".

Line 140 Label softkey 4 with "TEST 2".

Line 150 Label softkey 8 with "ABORT".

Line 160 Print a message to the user.

Line 170 Enable the HP–IB interface to detect HP–IB service requests and process the interrupt as ON PEN events.

Line 180 Line label for routine that is executed when an interrupt is detected.

Line 190 Enable HP–IB service request interrupt event trapping.

Line 200 Line label for routine that waits for an interrupt.

Line 210 If the last softkey pressed was not the "Abort" key (softkey 8, key code 41), continue looping.

Line 220 Disable HP–IB service request interrupt event trapping.

Line 230 End program execution.

Line 240 Line label for routine that services the interrupts.

Line 250 Serial poll the analyzer. Reading the status byte of the analyzer clears the SRQ. The CLEAR STATUS (CS) command could also be used. Perform error trapping.

Line 260 Command the analyzer to output the key code of the last key pressed.

Line 270 Read the key code.

Line 280 Multi–way branch on key code value. When lines are labeled with numbers in QuickBASIC, "SELECT CASE ... CASE" for the first case must occur on the same line and be separated by a statement separator. If the key code is 32, then softkey 1 was pressed.

Line 290 Move to row 12, column 29, on the computer CRT and print an appropriate message.

Line 300 If the key code is 8, then softkey 2 was pressed.

Line 310 Move to row 12, column 29, and print an appropriate message.

Line 320 If the key code is 0, then softkey 3 was pressed.

Line 330 Move to row 12, column 29, and print an appropriate message.

Line 340 If the key code is 16, then softkey 4 was pressed.

Line 350 Move to row 12, column 29, and print an appropriate message.

Line 360 If the key code is 41, then softkey 8 was pressed.

Line 370 Move to row 12, column 29, print an appropriate message, and go to the end of the program.

Line 380 If the key code doesn't match any of the preceding codes, another key was pressed. In this case, the key code has to be for softkey 5, 6, or 7 (key codes 14, 38, or 40) since these are the only other keys that can interrupt the computer.

Line 390 Move to row 12, column 29, and print an appropriate message.

Line 400 End of multi–way branch.

Line 410 Return from subroutine.

Line 420 Define a subroutine that outputs commands to the analyzer.

Line 430 Perform error trapping.

Line 440 Return from the subroutine.

Running program 8

1. Clear the computer CRT and type in the program.
2. Press **[ALT] [R] [S]** on the computer.
3. After the computer presets the analyzer and source, it writes the softkey labels on the analyzer CRT. The analyzer writes the first key label and blanks the other softkey labels. Softkeys 5, 6, and 7 remain blank because they are not given new labels.
4. Press any key on the analyzer. Pressing a softkey causes a message to be printed on the computer CRT. Softkeys 5, 6, and 7 generate an interrupt, even though they weren't labeled. No other keys of the analyzer generate an interrupt, because of the SRQ mask specified.

Because the analyzer was left in remote mode, it didn't respond to any keys pressed on its front panel. In some applications it is useful to put the analyzer into local operation, so that it can be controlled from the front panel and still generate interrupts whenever a key is pressed.

Program 9: CRT graphics

For applications requiring diagrams, drawings, or special limit lines, the CRT of the analyzer may be used as a plotter.

This program draws a connection diagram for a hypothetical test system measuring an amplifier. It will blank the analyzer's standard display containing the graticule, annotation, and softkeys so that we have a blank CRT. Figure 2 shows what the CRT should look like when the program is done.

For fast, easy–to–use graphics, the graphics memory of the HP 8757D/E is divided into seven "pages" of 500 words. One vector requires two words. Each of the pages may be selected to receive data, and turned on and off independently. You can keep different drawings in each of the graphics memory pages and simply turn on the drawing you need by turning on the appropriate page. Each page may also be erased independently.

To use the graphics capability of the HP 8757D/E, first define the passthru address to be one less than the analyzer's control address. If the analyzer's address is 16, its graphics address is 15. To the computer, the CRT of the analyzer looks like a plotter connected to the 8757 SYSTEM INTERFACE.

Program 9 listing

```
10   REM $INCLUDE: 'QB4SETUP'
20   CLS
30   ISC& = 7
40   Sna& = 716
50   Passthru& = 717
60   CALL IOTIMEOUT(ISC&, 10!):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
70   CALL IOABORT(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
80   CALL IOCLEAR(ISC&):
     IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
90   A$ = "IP BL5 PT15" GOSUB IOOUTS
100  A$ = "EP; GP1,1; DF" GOSUB IOOUTSP
110  A$ = "SP 9" GOSUB IOOUTSP
120  FOR Col = 0 TO 29
130  A$ = "PU;PA" +STR$(Col * 100) +",0;PD;PA"
     +STR$(Col * 100) +",2000;" GOSUB IOOUTSP
140  NEXT Col
150  FOR Row = 0 TO 20
160 A$ = "PU;PA 0," +STR$(Row * 100) +";PD;PA 2900,"
     +STR$(Row * 100) +";" GOSUB IOOUTSP
170  NEXT Row
180  A$ = "SP 1" GOSUB IOOUTSP
190  A$ = "PU; PA 600,1600; PD" GOSUB IOOUTSP
200  A$ = "SI0.28,0.34; LBCONNECTION DIAGRAM"
     +CHR$(3): GOSUB IOOUTSP
210  A$ = "PU; PA 1200,250; PD" GOSUB IOOUTSP
220  A$ = "SI0.28,0.34; LBDUT" +CHR$(3): GOSUB
     IOOUTSP
230  A$ = "PU; PA 300,800; PD; PA
     1100,800,1100,1100,300,1100,300,800" GOSUB
     IOOUTSP
240  A$ = "PU; PA 800,800; PD; PA 800,1100" GOSUB
     IOOUTSP
250  A$ = "PU; PA 1500,800; PD; PA
     2300,800,2300,1200,1500,1200,1500,800" GOSUB
     IOOUTSP
260  A$ = "PU; PA 1950,800; PD; PA 1950,1200" GOSUB
     IOOUTSP
270  A$ = "PU; PA 875,850; PD; PA 875,500,1200,500"
     GOSUB IOOUTSP
280  A$ = "PU; PA 1400,500; PD; PA
     2050,500,2050,850" GOSUB IOOUTSP
290  A$ = "PU; PA 1200,400; PD; PA
     1400,500,1200,600,1200,400" GOSUB IOOUTSP
300  A$ = "PU; PA 0,0" GOSUB IOOUTSP
310  END
320  IOOUTS: CALL IOOUTPUTS(Sna&, A$, LEN(A$))
330  IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
340  RETURN
350  IOOUTSP: CALL IOOUTPUTS(Passthru&, A$, LEN(A$))
360  IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
370  RETURN
```

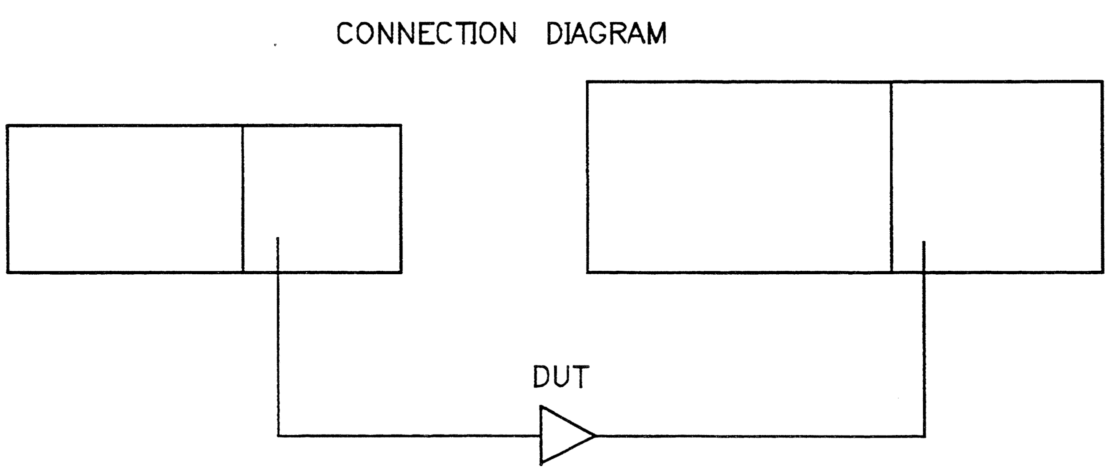


Figure 2. The CRT Graphics Display

Program 9 explanation

Line 10 Call the QuickBASIC initialization file "QBSETUP".

Line 20 Clear the computer CRT.

Line 30 Assign the interface select code to a variable.

Line 40 Assign the address of the analyzer to a variable.

Line 50 Assign the passthru address of the analyzer to a variable.

Line 60 Define a system timeout of 10 seconds. Perform error trapping.

Line 70 Abort any HP–IB transfers. Perform error trapping.

Line 80 Clear the HP–IB interface of the analyzer. Perform error trapping.

Line 90 Preset the analyzer and blank the CRT display. Define the CRT graphics as the target of passthru commands. The graphics address is always one less than the analyzer's HP–IB address.

Line 100 Erase all graphics pages. Turn on graphics page 1 to ensure that the graphics start in it. Set the color selection to default (monochrome) colors.

Line 110 Select to plot with pen 9, the lowest intensity for the analyzer CRT.

Line 120 Repeat a loop 29 times to draw part of the grid.

Line 130 Draw a vertical line down the CRT.

Line 140 End of the loop.

Line 150 Repeat loop 20 times to draw the horizontal part of the grid.

Line 160 Draw a horizontal line across the CRT.

Line 170 End of the loop.

Line 180 Select to plot with pen 1, the brightest intensity for the analyzer CRT.

Line 190 Move the pen to title the display.

Line 200 Specify the width and height of each character, indicate what the title is, terminate the title with an end of text character.

Line 210 Move the pen to label the DUT.

Line 220 Specify the width and height of each character, indicate what the title is, terminate the title with an end of text character.

Line 230 Move the pen and draw the outline of the source.

Line 240 Draw the plug–in of the source.

Line 250 Move the pen and draw the outline of the analyzer.

Line 260 Draw the CRT of the analyzer.

Line 270 Draw the connections from the source to the DUT.

Line 280 Draw the connections from the DUT to the analyzer.

Line 290 Draw the DUT (an amplifier.)

Line 300 Move to the bottom left corner of the CRT.

Line 310 End program execution.

Line 320 Define a subroutine that outputs commands to the analyzer.

Line 330 Perform error trapping.

Line 340 Return from the subroutine.

Line 350 Define a subroutine that addresses the analyzer as a plotter.

Line 360 Perform error trapping.

Line 370 Return from the subroutine.

Running program 9

1. Clear the computer CRT and type in the program.
2. Press **[ALT] [R] [S]** on the computer.
3. After the analyzer and source are preset, the CRT will be blanked. First a grid is plotted on the CRT. While this isn't necessary for our connection diagram, it does give you a good indication of where the X and Y coordinates are on the analyzer's CRT.
4. The labeling is added. The labels "CONNECTION DIAGRAM" and "DUT" are done using the analyzer CRT's internal character set.
5. All of the lines are plotted on the analyzer's CRT. If brighter lines are desired, draw each line twice or select a different pen number.

In this example, only graphics page 1 is used. You can independently control up to 7 separate pages of graphics information. If you write too much information onto one page, it overflows onto the next page.

When a graphics page is selected, the first location of memory that receives information (identified by the "pointer") is reset to the beginning of the page. Thus, as information is written onto the page, the old information is destroyed. If we were plotting a line, this would appear as a new trace overwriting an old one.

Program 10: learning the instrument state

Being able to save a specific instrument state is helpful when it is needed several times in a test or measurement procedure. You can save the instrument state by manually logging the important analyzer and source parameters, such as start/stop frequency, sweep time, number of trace points, scale per division, and display format, then replace them at the appropriate time. A simpler approach is to save the instrument state in one of the nine internal save/recall registers of the analyzer/source combination, then recall it when needed.

You have two additional options with HP–IB: the interrogate function and the learn string. With the output interrogated parameter function (OP), you can selectively interrogate the values of all functions that have numeric values (such as frequency and number of trace points). This function operates the same way in both the analyzer and the source. It is illustrated in program 3 where the source start and stop frequencies are interrogated in lines 110 through 140.

For a more thorough approach, use the learn string functions of the analyzer and source. The learn string describes the present instrument state and is similar to one of the internal save/recall registers. For the analyzer, the learn string also includes all of the global parameters, but does not include limit line information. Once an instrument state is learned, the analyzer and source states can be restored at any later time. The following program demonstrates how to both learn and restore the instrument states of the HP 8757D/E and the HP 8350B Sweeper by using their learn string functions. If you use the HP 8340B, 8341B, or 8360 series Synthesized Sweepers, perform the modification described at the end of "Running program 10."

Program 10 listing

```
10   REM $INCLUDE: 'QBSETUP'
20   CLS
30   Maxsna% = 300
40   Maxswpr% = 90
50   ISC& = 7
60   Sna& = 716
70   Passthru& = 717
80   CALL IOTIMEOUT(ISC&, 10):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
90   CALL IOABORT(ISC&):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
100  CALL IOCLEAR(ISC&):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
110  A$ = "IP" GOSUB IOOUTS
120  A$ = "PT19;" GOSUB IOOUTS
130  CALL IOLOCAL(Sna&):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
140  INPUT "SET UP SYSTEM,PRESS ENTER", B$
150  Match$ = CHR$(10):
     CALL IOMATCH(ISC&, Match$, 0): IF PCIB.ERR <>
     NOERR THEN ERROR PCIB.BASERR
160  A$ = "OL" GOSUB IOOUTS
170  Actual% = 0
180  Lsna$ = SPACE$(Maxsna%)
190  CALL IOENTERS(Sna&, Lsna$, Maxsna%, Actual%):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
200  A$ = "OL" GOSUB IOOUTSP
210  Actual% = 0
220  Lswpr$ = SPACE$(Maxswpr%)
230  CALL IOENTERS(Passthru&, Lswpr$, Maxswpr%,
     Actual%):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
240  CALL IOMATCH(ISC&, Match$, 1):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
250  A$ = "IP" GOSUB IOOUTS
260  INPUT "TO RESTORE SETUP,PRESS ENTER", B$
270  A$ = "IL" + Lsna$: GOSUB IOOUTS
280  A$ = "IL" + Lswpr$: GOSUB IOOUTSP
290  A$ = "" GOSUB IOOUTS
300  CALL IOLOCAL(Sna&):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
310  END
320  IOOUTS: CALL IOOUTPUTS(Sna&, A$, LEN(A$))
330  IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
340  RETURN
350  IOOUTSP: CALL IOOUTPUTS(Passthru&, A$, LEN(A$))
360  IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
370  RETURN
```

Program 10 explanation

Line 10	Call the QuickBASIC initialization file "QBSETUP".
Line 20	Clear the computer CRT.
Line 30	Define the maximum number of characters for the analyzer learn string.
Line 40	Define the maximum number of characters for the learn string of the source.
Line 50	Assign the interface select code to a variable.
Line 60	Assign the address of the analyzer to a variable.
Line 70	Assign the passthru address of the analyzer to a variable.
Line 80	Define a system timeout of 10 seconds. Perform error trapping.
Line 90	Abort any HP–IB transfers. Perform error trapping.
Line 100	Clear the HP–IB interface of the analyzer. Perform error trapping.
Line 110	Preset the analyzer and the source.
Line 120	Tell the analyzer which device is controlled through the passthru address. Address 19 belongs to the source.
Line 130	Set the analyzer and source to local mode. Perform error trapping.
Line 140	Prompt the user to set up the system. Then wait for the [ENTER] key to be pressed.

Line 150 Disable character matching for the linefeed. The analyzer learn string is 300 contiguous binary bytes (150 for the HP 8757E) that does not end with a cr/lf (since this could actually be part of the learn string information).

Line 160 Program the analyzer to output its learn string.

Line 170 Initialize the variable specifying the number of elements actually read.

Line 180 Allocate string space large enough to hold the learn string of the analyzer (150 bytes).

Line 190 Read the analyzer learn string into the string "Lsna$".

Line 200 Program the source to output its learn string.

Line 210 Initialize the variable specifying the number of elements actually read.

Line 220 Allocate string space large enough to hold the learn string of the source.

Line 230 Read the source learn string into the string "Lswpr$". The computer must read the entire source learn string which, for the HP 8350B Sweeper, is 90 bytes long.

Line 240 Enable character matching; this results in termination on a linefeed when a string is read.

Line 250 Preset the analyzer and source to clear the instrument states.

Line 260 Prompt the user, then wait for the [ENTER] key to be pressed.

Line 270 Program the analyzer to accept its learn string, then send it.

Line 280 Program the source to accept its learn string, then send it.

Line 290 Re−address the analyzer to exit passthru mode and continue sweeping.

Line 300 Set the analyzer and source to local mode.

Line 310 End program execution.

Line 320 Define a subroutine that outputs commands to the analyzer.

Line 330 Perform error trapping.

Line 340 Return from the subroutine.

Line 350 Define a subroutine that outputs commands to the source through the passthru address of the analyzer.

Line 360 Perform error trapping.

Line 370 Return from the subroutine.

Running program 10

1. Clear the computer CRT and type in the program.
2. Press **[ALT] [R] [S]** on the computer.
3. When the computer stops and displays:

   `SET UP SYSTEM, PRESS CONTINUE.`

   Adjust the analyzer and source to a preferred instrument state, then press the **[ENTER]** key on the computer.
4. The computer will save the learn strings of both the analyzer and the source. After completing this, the analyzer and source will be preset to destroy your original instrument state.
5. The computer stops and displays:

   `TO RESTORE SETUP, PRESS CONTINUE.`

   Press the **[ENTER]** key. The computer will restore your original instrument state via the two learn strings. Verify on the displays of the analyzer and the source that your state has been restored.

This example is designed to work with the HP 8350B Sweep Oscillator, which has a learn string of 90 bytes. The program can be easily modified to work with the HP 8340B and 8341B Synthesized Sweepers which have learn strings 123 bytes in length. To do this, change line 40 to be:

```
40   Maxswpr% = 123
```

To work with the HP 8360 Series Synthesized Sweeper, the modifications are more extensive due to its variable length learn string. To do this, change and/or add the following lines:

```
40   Maxswpr% = 700
212  Lswpr0$ = SPACE$(3)
214  CALL IOENTERS(Passthru&, Lswpr0$, 3, Actual%):
     IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
216  Maxswpr% = 256 * ASC(MID$(Lswpr0$, 2)) +
     ASC(MID$(Lswpr0$, 3))
218  Actual% = 0
280  A$ = "IL" + Lswpr0$+ Lswpr$: GOSUB IOOUTSP
```

The following should explain the above actions:

Line 212 Allocate string space large enough to hold the header portion of the HP 8360 learn string (3 bytes).

Line 214 Read the 3 header bytes. Bytes 2 and 3 indicate the number of bytes to follow.

Line 216 Compute the number of bytes to follow and change Maxswpr% to reflect this.

Line 220 Allocate string space large enough to hold the remainder of the HP 8360 learn string.

Line 230 Read the remainder of the HP 8360 learn string.

Line 280 Program the source to accept its learn string, then send it. For the HP 8360, the complete learn string is Lswpr0$+Lswpr$.

Program 11: guided instrument setup with CRT graphics

As was illustrated by program 9, it is possible to utilize the CRT of the HP 8757D/E as a plotter. This program goes one step further by utilizing the CRT to create a simple connection diagram which may be recalled by the user, at any time, from the front panel of the analyzer.

This program draws the same hypothetical connection diagram that was drawn by program 9. It blanks most of the analyzer's standard display, including the graticle and all annotation except the soft keys. In addition, it adds one softkey under both the save and the recall hardkey menus. This softkey will allow the user to toggle the state of the CRT graphics off and on.

To use the graphics off/on capability of the HP 8757D/E, change "BL5" in line 90 of program 9 to "BLA", and make the necessary changes in the size of the background grid. These, and other changes are illustrated in the following listing.

The same principle can be used to save anything stored to disk on the HP 8757D in the first seven pages of user graphics. By having the softkeys available, the user can store CRT graphics onto a disk for later recall.

Program 11 listing

```
10    REM $INCLUDE: 'QBSETUP'
20    CLS
30    ISC& = 7
40    Sna& = 716
50    Passthru& = 717
60    CALL IOTIMEOUT(ISC&, 10!):
      IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
70    CALL IOABORT(ISC&):
      IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
80    CALL IOCLEAR(ISC&):
      IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
90    A$ = "IP BLA PT15" GOSUB IOOUTS
100   A$ = "EP; GP1,1; DEC" GOSUB IOOUTSP
110   A$ = "SP 6" GOSUB IOOUTSP
120   FOR Col = 0 TO 25
130 A$ = "PU;PA" +STR$(Col * 100) +",0;PD;PA"
      +STR$(Col * 100) +",2000;" GOSUB IOOUTSP
140   NEXT Col
150   FOR Row = 0 TO 20
160 A$ = "PU;PA 0," +STR$(Row * 100) +";PD;PA 2500,"
      +STR$(Row * 100) +";" GOSUB IOOUTSP
170   NEXT Row
180   A$ = "SP 8" GOSUB IOOUTSP
190   A$ = "PU; PA 600,1600; PD" GOSUB IOOUTSP
200   A$ = "SI0.28,0.34; LBCONNECTION DIAGRAM"
      +CHR$(3): GOSUB IOOUTSP
210   A$ = "PU; PA 1200,250; PD" GOSUB IOOUTSP
220   A$ = "SI0.28,0.34; LBDUT" +CHR$(3): GOSUB
      IOOUTSP
230   A$ = "PU; PA 300,800; PD; PA
      1100,800,1100,1100,300,1100,300,800" GOSUB
      IOOUTSP
240   A$ = "PU; PA 800,800; PD; PA 800,1100" GOSUB
      IOOUTSP
250   A$ = "PU; PA 1500,800; PD; PA 2300,
      800,2300,1200,1500,1200,1500,800" GOSUB IOOUTSP
260   A$ = "PU; PA 1950,800; PD; PA 1950,1200" GOSUB
      IOOUTSP
270   A$ = "PU; PA 875,850; PD; PA 875,500,1200,500"
      GOSUB IOOUTSP
280   A$ = "PU; PA 1400,500; PD; PA
      2050,500,2050,850" GOSUB IOOUTSP
290   A$ = "PU; PA 1200,400; PD; PA
      1400,500,1200,600,1200,400" GOSUB IOOUTSP
300   A$ = "PU; PA 0,0" GOSUB IOOUTSP: A$ = "" GOSUB
      IOOUTS
310   CALL IOLOCAL(Sna&):
      IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
320   END
330   IOOUTS: CALL IOOUTPUTS(Sna&, A$, LEN(A$))
340   IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
350   RETURN
360   IOOUTSP: CALL IOOUTPUTS(Passthru&, A$, LEN(A$))
370   IF PCIB.ERR <>NOERR THEN ERROR PCIB.BASERR
380   RETURN
```

Program 11 explanation

Line 10	Call the QuickBASIC initialization file "QBSETUP".
Line 20	Clear the computer screen.
Line 30	Assign the interface select code to a variable.
Line 40	Assign the address of the HP 8757D/E to a variable.
Line 50	Assign the passthru address of the HP 8757D/E to a variable.
Line 60	Define a system timeout of 10 seconds. Perform error trapping.
Line 70	Abort any HP–IB transfers. Perform error trapping.
Line 80	Clear the HP–IB interface of the HP 8757D/E. Perform error trapping.
Line 90	Preset the analyzer and blank all the CRT display except the softkeys. Define the CRT graphics as the target of passthru commands. The graphics address is always one less than the analyzer's HP–IB address.
Line 100	Erase all graphics pages. Turn graphics page 1 on to ensure that the graphics start in it.
Line 110	Select to plot with pen 6, the lowest intensity for the analyzer CRT.
Line 120	Repeat a loop 25 times to draw vertical part of the grid.
Line 130	Draw a vertical line down the CRT screen.
Line 140	End of the loop.
Line 150	Repeat loop 20 times to draw horizontal part of the grid.
Line 160	Draw a horizontal line across the CRT.
Line 170	End of the loop.

Line 180 Select to plot with pen 8, the brightest intensity for the analyzer CRT.

Line 190 Move the pen to title the display.

Line 200 Specify the width and height of each character, indicate what the title is, terminate the title with an end of text character.

Line 210 Move the pen to label the device under test.

Line 220 Specify the width and height of each character, indicate what the title is, terminate the title with an end of text character.

Line 230 Move the pen and draw the outline of the source.

Line 240 Draw the plug–in of the source.

Line 250 Move the pen and draw the outline of the analyzer.

Line 260 Draw the CRT of the analyzer.

Line 270 Draw the connections from the source to the DUT.

Line 280 Draw the connections from the DUT to the analyzer.

Line 290 Draw the DUT (an amplifier.)

Line 300 Move to the bottom left corner of the CRT.

Line 310 Place the analyzer and the source in local mode. Perform error trapping.

Line 320 End program execution.

Line 330 Define a subroutine that outputs commands to the analyzer.

Line 340 Perform error trapping.

Line 350 Return from the subroutine.

Line 360 Define a subroutine that addresses the analyzer as a plotter.

Line 370 Perform error trapping.

Line 380 Return from the subroutine.

Running program 11

1. Clear the screen of the computer and type in the program.
2. Press **[ALT] [R] [S]** on the computer.
3. After the analyzer and source are preset, the CRT is blanked, except for softkeys. First a grid is plotted on the CRT. While this isn't necessary for our connection diagram, it does give you a good indication of where the X and Y coordinates are on the analyzers' CRT.
4. The labelling is added. The labels "CONNECTION DIAGRAM" and "DUT" are written using the analyzer CRT's internal character set.
5. All of the lines are plotted on the analyzer's CRT. If brighter lines are desired, draw each line twice or, select different pen numbers.
6. The analyzer is placed in local mode with the front panel and the softkeys active. To access the graphics on/off capability, press **[SAVE]** on the analyzer to show the save menu. Press the **[STORE TO DISK]** softkey. Note the **[GRAPHIC ON/OFF]** softkey, it does not appear unless the "BLA" command is used. Press the **[GRAPHIC ON/OFF]** softkey so that it is "off." The connection diagram will now disappear from the CRT display. Press the **[GRAPHIC ON/OF]** softkey again and the diagram will reappear. If you store this setup to the external disk drive at this time, the analyzer will remember this graphics on/off mode later upon recall from disk.

Table 2. Alphabetical Listing of HP 8757D/E Programming Codes (1 of 3)

Code	Action	Code	Action
A0	Averaging off	CLS	Color list, salmon[1]
AB	A/B ratio measurement	CLW	Color list, white[1]
AC	A/C ratio measurement[2]	CLY	Color list, yellow[1]
AFd	Averaging on and factor d	CN	Cursor to minimum
ANm	Adaptive Normalization on/off	COBd	Brightness adjust, one color[1]
AR	A/R ratio measurement	COCd	Color adjust, one color[1]
AS	Autoscale	COTd	Tint adjust, one color[1]
AZ2	Autozero the DC detectors once	CR	C/R ratio measurement[2]
AZm	Autozero repeat on/off of the DC detectors	CS	Clear status bytes
BA	B/A ratio measurement	CTm	Auto system calibration on/off
BC	B/C ratio measurement[2]	CUm	Cursor on/off
BFm	Plotter buffer on/off[3]	CWm	CW mode on/off
BL0	Restore CRT to normal mode	CX	Cursor to maximum
BL1	Blank frequency labels (secure frequency mode, frequency labels cannot be restored)	DAd	Detector A amplitude offset set to d
BL2	Blank all labels	DBd	Detector B amplitude offset set to d
BL3	Blank active channel trace	DCd	Detector C amplitude offset set to d[2]
BL4	Blank softkey labels	DEC	Set default colors[1]
BL5	Blank all (except user CRT graphics)	DFA	Set disk format to ASCII[1]
BL6	Blank title	DFB	Set disk format to binary[1]
BL7	Blank mode labels	DFE	Set Disk format to extended binary[1]
BL8	Blank the active entry area	DHm	Display Hold on/off of the active channel trace
BL9	Blank the limit lines	DIAd	Set disk HP−IB address[1]
BLA	Blank all (except user CRT graphics and softkeys)	DIUd	Set disk unit number[1]
BR	B/R ratio measurement	DIVd	Set disk volume number[1]
BTNd	Overall display brightness	DLF	Delete file from disk[1]
BW	Display the search bandwidth on the CRT[1]	DM0	All inputs set to DC detection
C0	Channel off	DM1	All inputs set to AC detection
C1	Channel 1 on/active	DN	Step down (decrement)
C2	Channel 2 on/active	DOAd	Measure Detector A amplitude offset
C3	Channel 3 on/active[1]	DOBd	Measure Detector B amplitude offset
C4	Channel 4 on/active[1]	DOCd	Measure Detector C amplitude offset[2]
CA	C/A ratio measurement[2]	DORd	Measure Detector R amplitude offset
CB	C/B ratio measurement[2]	DRd	Detector R amplitude offset set to d
CC1	Set channel 1 color[1]	DS0	Display trace data in log magnitude
CC2	Set channel 2 color[1]	DS1	Display trace data in standing wave ratio (SWR) format
CC3	Set channel 3 color[1]	DTSTPAs	Enter stop frequency for detector A
CC4	Set channel 4 color[1]	DTSTPBs	Enter stop frequency for detector B
CDm	Cursor delta on/off	DTSTPCs	Enter stop frequency for detector C[2]
CGL	Set labels color[1]	DTSTPRs	Enter stop frequency for detector R
CGN	Set background color[1]	DTSTRAs	Enter start frequency for detector A
CGR	Set grid color[1]	DTSTRBs	Enter start frequency for detector B
CGW	Set warning label color[1]	DTSTRCs	Enter start frequency for detector C[2]
CL	Perform system configuration of detectors and channels	DTSTRRs	Enter start frequency for detector R
CLB	Color list, black[1]	EO	Enter measured detector amplitude offset
CLG	Color list, green[1]	ER0	Erase all save/recall registers
CLL	Color list, blue[1]	FAs	Start frequency label
CLR	Color list, red[1]	FBs	Stop frequency label
		FD0	Format data ASCII

1. HP 8757D only
2. HP 8757D Option 001 only
3. Revision 3.1 or above for HP 8757E.

Table 9. Alphabetical Listing of HP 8757D/E Programming Codes (2 of 3)

Code	Action
FD1	Format data binary (HP BASIC compatible)
FD2	Format data extended ASCII
FD3	Format data binary (PC compatible)
FD4	Format data extended binary (HP BASIC compatible)
FD5	Format data extended binary (PC compatible)
FR0	Logarithmic (dB) cursor format[3]
FR1	SWR cursor format[3]
FSm	Step sweep on/off[3,4]
FTAm	Detector A frequency on/off
FTBm	Detector B frequency on/off
FTCm	Detector C frequency on/off[2]
FTRm	Detector R frequency on/off
IA	Input A absolute power measurement
IB	Input B absolute power measurement
IC	Input C absolute power measurement[2]
ILs	Input Learn string
IND	Initialize disk format[1]
IP	Instrument preset
IR	Input R absolute power measurement
IX	External ADC input (AUX) voltage measurement1
LE	Erase limit lines for active channel[5]
LFA	Load instrument information file from disk[1]
LFC	Load CRT graphics file from disk[1]
LFD	Load data trace file from disk[1]
LFF	Load measurement file from disk.[1]
LFH	Load instrument information file from disk and place instrument in hold mode.[1]
LFI	Load instrument state file from disk[1]
LFM	Load memory trace file from disk[1]
LFN	Load display trace file from disk.[1]
LFs	Enter limit test flat line data[5]
LL	Store lower limit line into memory[5]
LPs	Enter limit test point data[5]
LSs	Enter limit test sloped line data[5]
LTm	Limit line test on/off[5]
LU	Store upper limit line into memory[5]
M−	Display normalized data (measurement − memory)
MDm	Modulation on/off
ME	Display measurement data
MM	Display the channel menu(main menu)
MN	Display normalized data (same as M−)
MOC	Monochrome display[1]
MR	Marker (or cursor) to reference line
MSm	Manual sweep mode on/off
MU0	Display the measurement menu
MU1	Display the display menu
MU2	Display the scale menu
MU3	Display the reference menu
MU4	Display the cursor menu
MU5	Display the average menu
MU6	Display the calibration menu
MU7	Display the special menu
MU8	Display the system menu
MY	Display memory data
MZ	Manual calibration of DC detectors
NSm	Non−standard sweep mode on/off
OC	Output cursor value
OD	Output trace data
OE1	Output error status of display channel 1
OE2	Output error status of display channel 2
OI	Output identity
OK	Output keycode of last key pressed
OL	Output learn string
OM	Output memory data
ON	Output normalized (measurement —memory) data
OPDO	Output measured detector amplitude offset
OPxx	Output interrogated parameter value xx= AF, BW, DA, DB, DC, DR, RL, RP, SD, SL, SO, SP, SR, SS, ST
OR	Output rotary knob value (—32768 ≤ value ≤ +32767)
OS	Output status bytes
OT1m	Control output #1 on/off
OT2m	Control output #2 on/off
OV	Output CW value
P1	Plot channel 1 trace on external plotter
P2	Plot channel 2 trace on external plotter
P3	Plot channel 3 trace on external plotter[1]
P4	Plot channel 4 trace on external plotter[1]
PA	Plot all on external plotter
PBm	System interface control on/off
PC	Plot labels on external plotter
PD	Plot custom plot
PG	Plot grid on external plotter
PR1	Print all to monochrome printer, except softkeys and CRT graphics
PR2	Print tabular display data in monochrome
PR3	Print tabular marker/cursor data to external printer
PR4	Print all to color printer, except softkeys and CRT graphics[1]
PTd	Passthrough address set to d
PWRA	Execute a detector A power calibration
PWRB	Execute a detector B power calibration
PWRC	Execute a detector C power calibration[2]
PWRR	Execute a detector R power calibration
R1	R/A ratio measurement
R2	R/B ratio measurement

1. HP 8757D only
2. HP 8757D Option 001 only
3. Revision 3.1 or above for HP 8757E.
4. HP 8340, HP 8341, or HP 8360 series synthesized sweeper only with 8757 SYSTEM INTERFACE connected and active.
5. Limit line functions valid only for channels 1 or 2,. HP 8757D only.

Table 9. Alphabetical Listing of HP 8757D/E Programming Codes (3 of 3)

Code	Action	Code	Action
R3	R/C ratio measurement[2]	SR	Cursor search right[1]
RCn	Recall register n	SSd	Cursor search value set to d[1]
RLd	Reference level set to d	STd	Reference level step size set to d
RMd	Service request mask set to d	SUd	Specify custom plot according to d
RPq	Reference position set to vertical division q	SVn	Save register n
RS	Restart averaging	SW0	Non–swept mode; non–swept operation
SCd	Set cursor to horizontal position d	SW1	Swept mode; normal swept operation
SDd	Scale per division set to d	SW2	Sweep hold mode; non–swept mode with HP–IB bus hold off until completion of TSd
SFA	Store all instrument information to disk in file[1]	TCm	Continuous Temperature Compensation on/off
SFC	Store CRT graphics to disk in file[1]	TIFs	Title for file[1]
SFD	Store data trace to disk in file[1]	TSd	Take d sweeps, then hold display
SFI	Store instrument state to disk in file[1]	UP	Step up (increment)
SFM	Store memory trace to disk in file[1]	WKs	Write softkey label
SFN	Store normalized trace to disk in file[1]	WMs	Write to channel memory.
SKq	Select softkey q: q =1 to 8	WTs	Write title, s is an ASCII string of up to 50 characters
SL	Cursor search left[1]	XAs	External detector cal value for detector A
SM	Store measurement into memory	XBs	External detector cal value for detector B
SN	Store normalized data (measurement – memory) into memory	XCs	External detector cal value for detector C[2]
SOd	Smoothing set to d % of frequency span	XRs	External detector cal value for detector R
SPd	Number of points set to d: d=101, 201, 401, 801[1], 1601[1]		

1. HP 8757D only
2. HP 8757D Option 001 only (detector C)

NOTES:
- n = decimal integer 1 to 9
- d = variable length numeric
- m = 0 for off/1 for on
- q = unique value
- s = ASCII or binary string

For more information, call your local HP sales office listed in your telephone directory or an HP regional office listed below for the location of your nearest sales office.

United States:
Hewlett–Packard Company
4 Choke Cherry Road
Rockville, MD 20850
(301) 670–4300

Hewlett–Packard Company
5201 Tollview Drive
Rolling Meadows, IL 60008
(312) 255–9800

Hewlett–Packard Company
5161 Lankershim Blvd.
No. Hollywood, CA 91601
(818) 505–5600

Hewlett–Packard Company
2015 South Park Place
Atlanta, GA 30339
(404) 955–1500

Canada:

Hewlett–Packard Ltd.
6877 Goreway Drive
Mississauga, Ontario L4V1M8
(416) 678–9430

Australia/New Zealand:
Hewlett–Packard Australia Ltd.
31–41 Joseph Street,
Blackburn, Victoria 3130
Melbourne, Australia
(03) 895–2895

Europe/Africa/Middle East:
Hewlett–Packard S.A.
Central Mailing Department,
P.O. Box 529
1180 AM Amstelveen,
The Netherlands
(31) 20/547 9999

Far East:
Hewlett–Packard Asia Ltd.
22/F Bond Centre
West Tower
89 Queensway
Central, Hong Kong
(5) 8487777

Japan:
Yokogawa–Hewlett–Packard Ltd.
29–21, Takaido–Higashi 3–chome
Suginami–ku, Tokyo 168
(03) 331–6111

Latin America:
Latin American Region Headquarters
Monte Pelvoux Nbr. 111
Lomas de Chapultepec
11000 Mexico, D.F. Mexico
(905) 596–79–33

HEWLETT PACKARD

July 15, 1992
HP Part Number 08757–90117
Printed in U.S.A.