HP **8920, 8921** RF Communications Test Set Family

Product Note

# Using the IBASIC Programming Environment on the HP 8920 Test Set Family

HEWLETT® PACKARD
Expanding Possibilities

Contents

# Table of Contents

# Conventions Used in this Product Note

The generic term “test set” is used interchangeably in this product note for the following products:

HP 8920A RF Communications Test Set

HP 8921A Cell Site Test Set

The generic abbreviation "PC" refers to computers compatible with the IBM® personal computer (PC). The PC is used as the example external controller in this product note. Computers other than the PC can also be used to program the test set.

HP-IB is Hewlett-Packard Company’s implementation of the IEEE 488.2 interface bus standard.

Conventions used in defining program steps:

Screen titles are shown: **RF ANALYZER**

Field names are shown: **Squelch**

Field contents and measurement results are shown: ***RF In***

Front-panel keys are shown: **(TESTS)**

[1] IBM® is a U.S. registered trademark of International Business Machines Corp.

Chapter One

# Chapter 1
# Overview of the Test Set

## Reference Documents

Table 1-1 lists other HP documents which will help you develop programs for your test set. Later versions of the same documents may have the same information but in different chapters or under different headings.

| Document Title | HP Part Number | Useful Programming Sections and Information |
|---|---|---|
| HP 8920A, 8921A Programmer's Guides | 08920-90220<br>08921-90031 | • General HP IBASIC reference with syntax lists<br>• A must for serious Test Set programmers<br>• IBASIC error messages |

Table 1-1. Useful programming documents

## The Test Set

The test set is designed to test mobile radios, base stations, cellular telephones, and other communication systems in service and manufacturing environments. The test set combines the features of 22 individual radio test instruments and a Hewlett-Packard instrument BASIC (IBASIC) controller into one package. With the exceptions of volume and squelch vernier controls, all of the test set's functions can be automatically controlled through application programs running on the built-in IBASIC controller or on an external controller connected via HP-IB. The test set's measurement capability can be extended by adding application specific cellular adapters. Current cellular adapters are TDMA test for the North American dual-mode cellular system (IS-54 and IS-136), PDC test for the Japanese personal digital communication system, PHP test for the Japanese personal handy phone system, and CDMA test for the North American code division multiple access cellular system (IS-95).

## Available Software

Hewlett-Packard has an extensive range of test set application programs available to simplify both digital and analog measurements of cellular mobile telephones, cellular telephone cell sites (base stations), private mobile radios (conventional and trunked), and other equipment. They are available as various options to the HP 11807A radio test software and the HP 11807B cell site test software. The tests are based on North American, European, and Japanese communications requirements and standards, as well as manufacturer recommendations.

# This Product Note

This product note is designed to help you write your own programs using the built-in IBASIC controller. Developing programs for the test set is simplified if the programmer has a basic understanding of how the test set operates. An overview of the test set's operation is best presented in terms of how information flows through the unit. The simplified block diagrams presented in Figures 1-1 and 1-2 show how instrument control information and measurement result information is routed between the test set's instruments, instrument control hardware, built-in IBASIC controller, and other components.

# Test Set Operating Modes

Test set usage can be simply modeled if divided into two operating modes:

**Manual Control Mode**

- Test set operation is controlled through the front-panel rotary knob and keypad. This mode is briefly described in this chapter to help you understand the test set.

**Automatic Control Mode**

- External Automatic Control mode: Test set operation is controlled by an external controller connected to the test set via the HP-IB interface. Some information about this mode can be inferred from topics covered in this product note, but this mode is not covered in detail.
- Internal Automatic Control mode: Test set operation is controlled by an application program running on the built-in IBASIC Controller. This product note primarily covers this mode.

# Manual Control Mode

## Instrument Types

The test set's primary instruments are shown on the left side of Figure 1-1. Table 1-2 lists the two basic types of instruments in the test set.

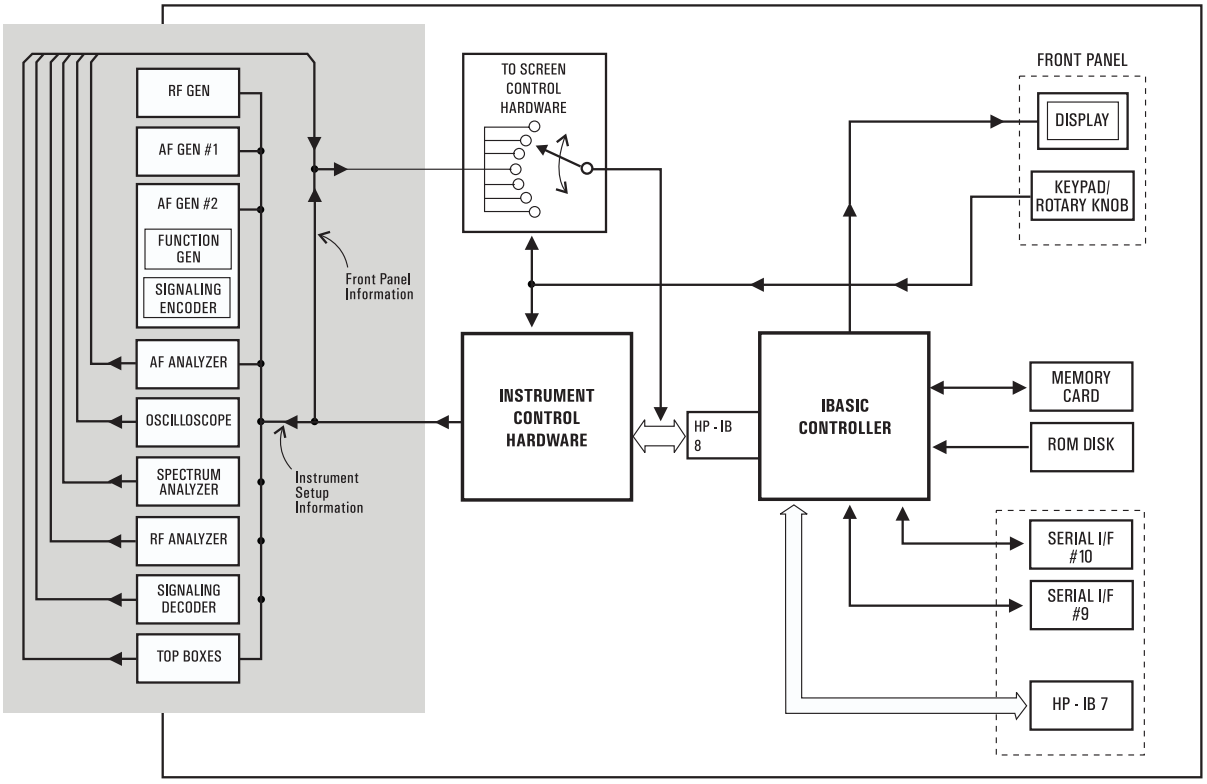


Figure 1-1. Manual control mode functional block diagram

| Instrument Type | Instrument |
|---|---|
| Analyzer | **RF ANALYZER** (power meter, frequency counter, modulation meter)<br>**AF ANALYZER** (ac/dc voltmeter, SINAD meter, audio filters)<br>**OSCILLOSCOPE**<br>**SPECTRUM ANALYZER** (optional in some test sets)<br>**SIGNALING DECODER** |
| Source | **RF GENERATOR**<br>**AF Generator 1**<br>**SIGNALING ENCODER** (AF Generator 2) |

Table 1-2. Test set instrument types

(In addition, all test sets can internally route RF or AF signals among the various connectors and instruments, and can turn device-under-test transmitters on and off with a relay closure. Also, with the optional radio interface card installed, the test set can send and receive 16-bit parallel digital signals to and from the device-under-test).

## Front-Panel Displays

Since so many instruments are integrated into the test set, it is not feasible to have an actual front panel for each instrument. Therefore, each instrument's front panel is maintained in firmware and is displayed whenever the instrument is selected. Only one instrument can be displayed at a time (but up to four measurement results can be seen at once if desired). Just as with stand-alone instruments, instrument front panels in the test set can contain:

- Instrument setting information
- Measurement results
- Data received from the device-under-test (DUT)

## Manual Operation

Using the test set manually is very similar to using a set of bench- or rack-mounted test equipment. To obtain a measurement result with a bench or rack system, you must use the appropriate instrument and make it active. Conceptually, the same measurement process applies to individual instruments of the test set. Table 1-3 compares identical measurements with an individual instrument and a test set.

| Bench- or Rack-Mount Test Equipment | Test Set |
|---|---|
| 1. Choose the RF power meter as the measurement instrument. | 1. Choose the RF power meter as the measurement instrument by selecting the **RF ANALYZER** screen from the **To Screen** menu using the rotary knob. |
| 2. Turn the power meter on. | 2. Turn the power meter measurement on by selecting the field to the right of **TX Power** and pressing the **(ON/OFF)** key. |
| 3. Connect the DUT to the power meter's RF input. | 3. Connect the DUT to the RF IN/OUT connector on the Test Set. |
| 4. Select the correct power meter functions on the front panel. | 4. Change other necessary fields on the **RF ANALYZER** screen (such as **Input Port** to ***RF In.***) |
| 5. Read the measurement result on the power meter's display. | 5. Read the measurement result on the Test Set's CRT under **TX Power.** |

Table 1-3. Steps to make an RF power measurement

Figure 1-1 illustrates that instrument selection is handled by the **To Screen** control hardware which routes the selected instrument's front panel to the display. Once an instrument's front panel has been displayed, the user can manipulate the instrument settings using the rotary knob or keypad. Figure 1-1 also illustrates how instrument setup is handled by the instrument control hardware which routes setup information from the front panel to the individual instruments.

Both HP-IB and serial RS-232 bidirectional interfaces are available in the test set. In manual control mode, they are useful for connecting an external HP-IB or serial printer to print displayed screens.

# Automatic Control Mode

*Note: This section covers internal automatic control mode. Information about external automatic control of the test set by an external computer connected to the test set via the HP-IB interface can be inferred from topics covered in this section, but this mode is not covered in detail.*

In internal automatic control mode, the test set's operation is controlled by an application program running on the built-in IBASIC controller. Hewlett-Packard instrument BASIC is the only programming language supported on the built-in controller of the test set. IBASIC is a subset of the Hewlett-Packard BASIC programming language used on the HP 9000 series 200 and 300 workstation controllers and PCs.

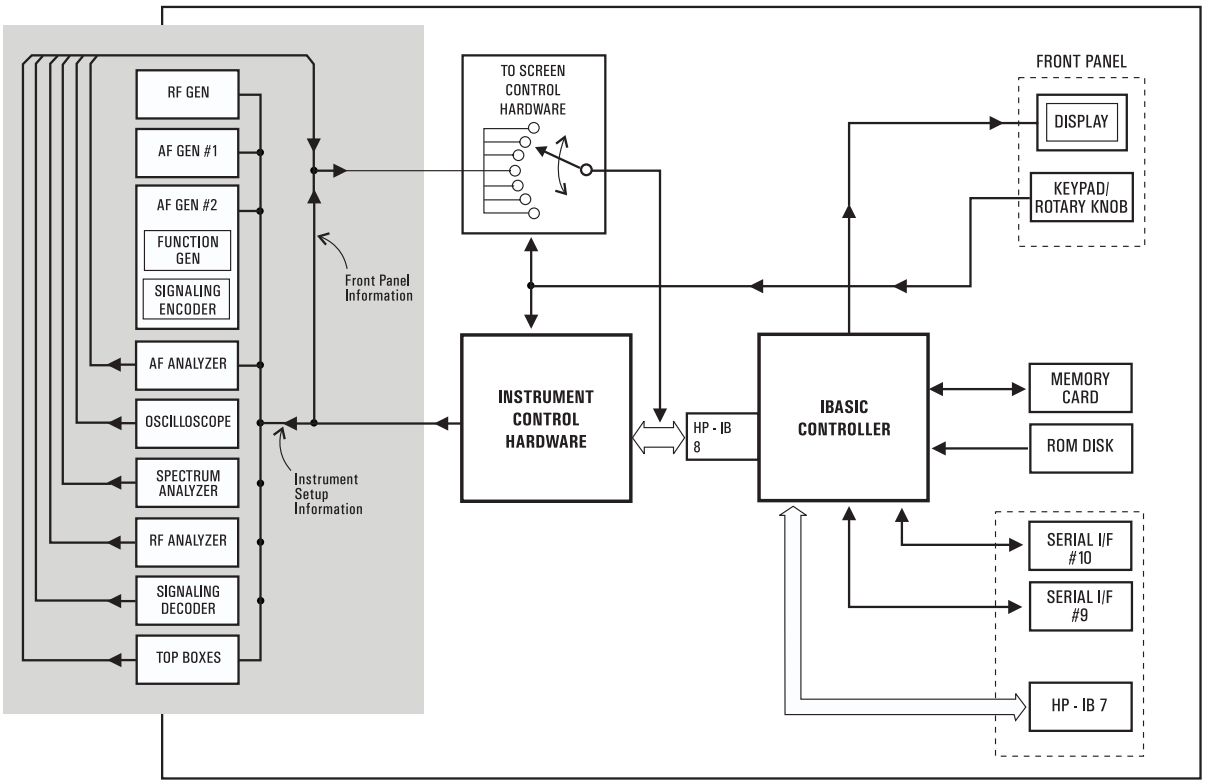


Figure 1-2. Internal automatic control mode functional block diagram

# IBASIC Controller Characteristics

Some characteristics of the built-in IBASIC controller are very similar to that of other computers, and some are quite different. The IBASIC controller can be described as a computer residing in an instrument.

- Only one program can be run on the IBASIC controller at a time. It is single-tasking.
- The program must not exceed the test set's on-board RAM memory space.

## Four Types of Mass Storage

The program is loaded into RAM memory from some type of mass storage device. Four types of mass storage devices are available for the test set:

**Cards**

Memory cards plug into the front-panel slot of the test set.

**SRAM** (Static Random Access)

SRAM memory cards are primarily used for program development and storage. They can be repeatedly written to and over-written, and are battery-backed.

**OTP** (One-time Programmable Read Only Memory)

OTP memory cards contain factory-supplied application programs or data that cannot be over-written or erased by the user. (HP 11807A, HP 11807E, and HP 11807B software comes on this format).

**Disk Drives**

These are typically 3 1/2" flexible disks or hard disks which can be connected to the HP-IB interface.

**RAM Disks**

RAM disks are random access memory in four battery-backed memory locations inside the test set's RAM that can be used by the programmer very much like SRAM memory cards, but cannot be removed from the test set.

**ROM**

This read only memory inside the test set contains factory-supplied programs accessible to the user that cannot be erased.

More detail on mass storage is given in the Appendix.

## Four Types of I/Os

Four types of input/output (I/O) interfaces are available for connecting to external computers, instruments, and other equipment:

**HP-IB**

HP's implementation of the IEEE 488.2 standard; available as Option 103 on the HP 8920A; a standard feature on other HP test sets.

**Serial**

RS-232 compatible; available as Option 103 on the HP 8920A; a standard feature on other HP test sets.

**Radio Interface**

16-bit parallel data I/O; available as Option 020 (radio interface card) on all HP test sets; this interface is treated as an internal instrument and has its own screen.

**Parallel Port**

This port is used with printers requiring a parallel interface; available as Option 103 on the HP 8920A; a standard feature on other HP test sets.

Figure 1-2 shows how information is routed inside the test set when it is in internal automatic control mode. In manual control mode, certain test set resources are dedicated to manual operation. In internal automatic control mode (that is, when an IBASIC program is running), these resources are switched to the IBASIC controller. Table 1-4 shows some of the relationships.

| Test Set Feature | Manual Control Mode (Front-Panel Control) | Internal Automatic Control Mode (IBASIC program running) |
|---|---|---|
| Front-Panel Display | The display shows instrument settings, measurement results, and data received from the DUT through the **To Screen** control hardware | The display is dedicated to the IBASIC Controller for program and graphics display; instrument front panels cannot be displayed while a program is running |
| Serial Interface (select code 9; available at rear-panel connector) | serial port 9 is under manual control; Example: The user can print a CRT display to a printer if connected to this serial port | Under IBASIC, a program can control other devices using the serial port (like a DUT or printer) |
| HP-IB Interface (select code 7; available at rear-panel connector) or Parallel Interface (select code 15) | HP-IB is under manual control; Example: The user can print a CRT display to a printer if connected to the HP-IB port or Parallel Port | Under IBASIC, a program can control other devices over HP-IB (like a power supply or printer, but will only print on the parallel port) |

Table 1-4. Allocation of controller resources in manual and automatic mode

## Dedicated Internal HP-IB Interface Bus

In internal automatic control mode, the measurement results and data input from the device under test (DUT) are routed to the IBASIC controller through a dedicated internal interface bus (shown in Figure 1-2 as HP-IB 8).

Key features of the dedicated internal bus:

- It communicates with the internal instruments of the test set.
- It is available only to the built-in IBASIC controller.
- No external connector is available for this bus.
- No external instruments may be added to this bus.
- It conforms to the IEEE 488.2 standard in all respects but one: each internal instrument on the internal bus does not have a unique address.
- The test set control hardware determines which internal instrument is being addressed on the bus through the command syntax.

The externally-accessible HP-IB interface bus, select code 7 in Figure 1-2, is used to connect the test set to external instruments or to an external controller. Refer to the *Programmer's Guide* for your test set for a listing of the HP-IB command syntax.

## Important Notes for Writing Programs

The test set is designed to operate the same way under automatic control as it does under manual control. This has several implications when designing and writing programs for the test set:

- To automate a particular task, determine how to do the task manually and then duplicate the steps in your program.
- In manual control mode, an analyzer function must be displayed and active to make a measurement or receive DUT data. Therefore, if you want to make a measurement in your program, you must follow these basic steps:

1. Use the **DISPlay** command to select the instrument whose front panel contains the desired measurement result or data field (such as **AF ANalyzer**).
2. Set the measurement field (such as ***SINAD***) to the **(ON)** state.
3. Then trigger a reading and read the result.

# Program Development

There are three recommended approaches to developing IBASIC programs. They are outlined below and discussed in more detail in Chapter 3. As shown in Figure 1-3, all three development methods employ an external computer (or terminal). Your choice of development method will typically be driven by the available equipment and by the extent of the development task. If you have a large development task, it is strongly recommended you use a BASIC language computer as outlined in development Method #1, by using a PC based tool, HP 83224A.

Method #2 is for program modification or smaller program development. Since the test set only has the knob and numeric keypad for program entry, developing programs on the test set alone is not recommended. Method #3 is least preferred for program development or modification because no syntax checking occurs until the program is first run making it difficult to debug long programs. Details of each development method are given in Chapter 3.

**NOTE:**

Another development method, writing programs for the TESTS subsystem, is referenced in the *HP 8920A Instrument BASIC Programmer's Guide*. This method uses the DEV_PL programs found on the Hewlett-Packard Test Set Program Development Tool (on a 3 ½ inch flexible disk). The DEV_PL programs will **NOT** operate with a typical PC, and therefore will not be described in this product note.

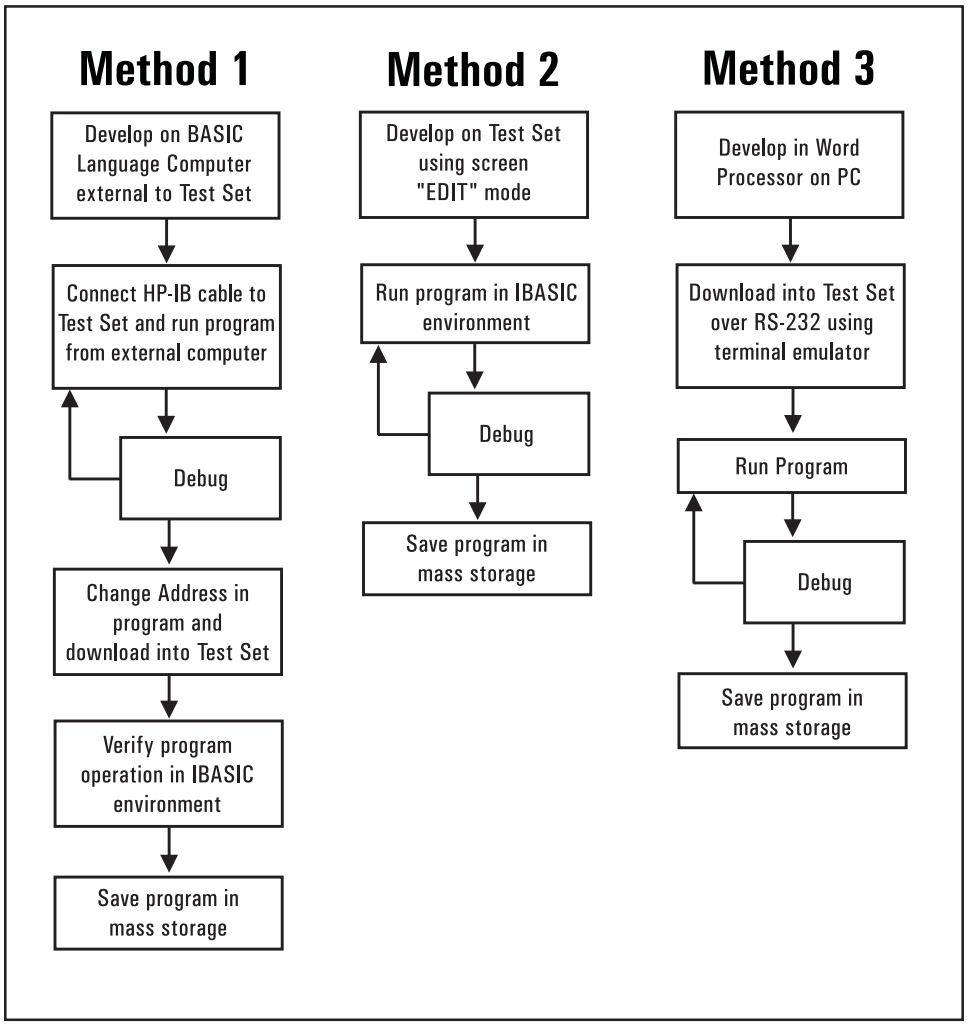


Figure 1-3. Program development methods

# Chapter 2
# Interfacing with the IBASIC Controller

This chapter describes how to interconnect and set up the test set and peripheral devices so you can develop IBASIC programs. To determine which programming environment best fits your application, refer to Chapter 3.

## Test Set Configuration

To prepare for IBASIC programming, you must first configure the test set to operate with your PC or terminal.

This includes:

- Hardware setup
- Cables
- Communication settings: **I/O CONFIGURE** and **TESTS (IBASIC CONTROLLER)** screens

## Serial Port Configuration

In general, this chapter assumes that you will be using the serial port to develop programs to use the internal automatic control mode of the test set.

There are actually two independently controllable serial interfaces on each test set, each using a 3-wire transmit, receive, ground implementation of the RS-232 standard. The IBASIC controller can send and receive data from either port by using its assigned select code.

**Information about the two serial ports**
The rear-panel RJ-11 connector has 6 conductors. (Note that this jack appears the same as a common 4-conductor RJ-11 telephone jack, but the test set jack uses 6 conductors). Three of the wires are designated as serial I/O port address 9, and the other three wires are designated serial I/O port address 10 (also referred to as serial port A and B, respectively). The user cannot change these select codes.

**Serial port 9**
Serial port 9 is used for developing and editing programs because it is the one connected directly to the **IBASIC Command Entry** field in the **TESTS** (IBASIC controller) screen. It is also useful for data I/O from an IBASIC program. Port settings can be changed from the **I/O CONFIGURE** screen, from IBASIC commands, or from an IBASIC program.

**Serial port 10**
Serial port 10 is useful for data I/O from an IBASIC program to an external device. Settings can be changed from IBASIC commands or from an IBASIC program (not from the **I/O CONFIGURE** screen).

**Why does the test set have two serial ports?**
A typical application involving both ports uses port 10 to send and receive data to and from a DUT while it simultaneously uses port 9 to print or log test results to a serial printer or PC. In this example, you can edit your IBASIC program from the PC without switching ports. This is schematically shown in Figure 2-2. If you do not have simultaneous serial I/O needs, you should use only serial port 9.

Figure 2-1 and Table 2-1 show the cables and adapters that are available from Hewlett-Packard. (See Figure 2-2 for a wiring diagram to construct your own cables. RJ-11 cables and adapters can be wired several ways. If you buy a cable or adapter other than the HP parts listed in Table 2-1, verify the connections for the pins indicated before connecting the cables to the test set).

| Device (all for RS-232 Serial connections) | Typical Uses | Description | HP Part Number |
|---|---|---|---|
| **Single to Dual RJ-11 Adapter Cable** | To connect to serial ports 9 and 10 simultaneously | Single 6-pin RJ-11 (male) to dual 6-pin RJ-11 (female); 0.6 meter cable | 08921-61031 |
| **Cable with Connectors** | Test Set to PC | 6-pin RJ-11 (male) to 9-pin DB-9 (female); 2 meter cable | 08921-61038 |
| **Cable with Connectors** | Test Set to printer or terminal | 6-pin RJ-11 (male) to 25-pin DB-25 (male); 3 meter cable | 08921-61039 |
| **RJ-11 to DB-25 Adapter** | Use with 98642-66505 long cable | 6-pin RJ-11 (female) to 25-pin DB-25 (male) adapter | 98642-66508 |
| **Cable with Connectors** | Long cable from Test Set to PC or printer (use with 98642-66508) | 6-pin RJ-11 (male) to 6-pin RJ-11 (male); 15 meter cable | 98642-66505 |

Table 2-1. Available HP RS-232 serial cables and adapters

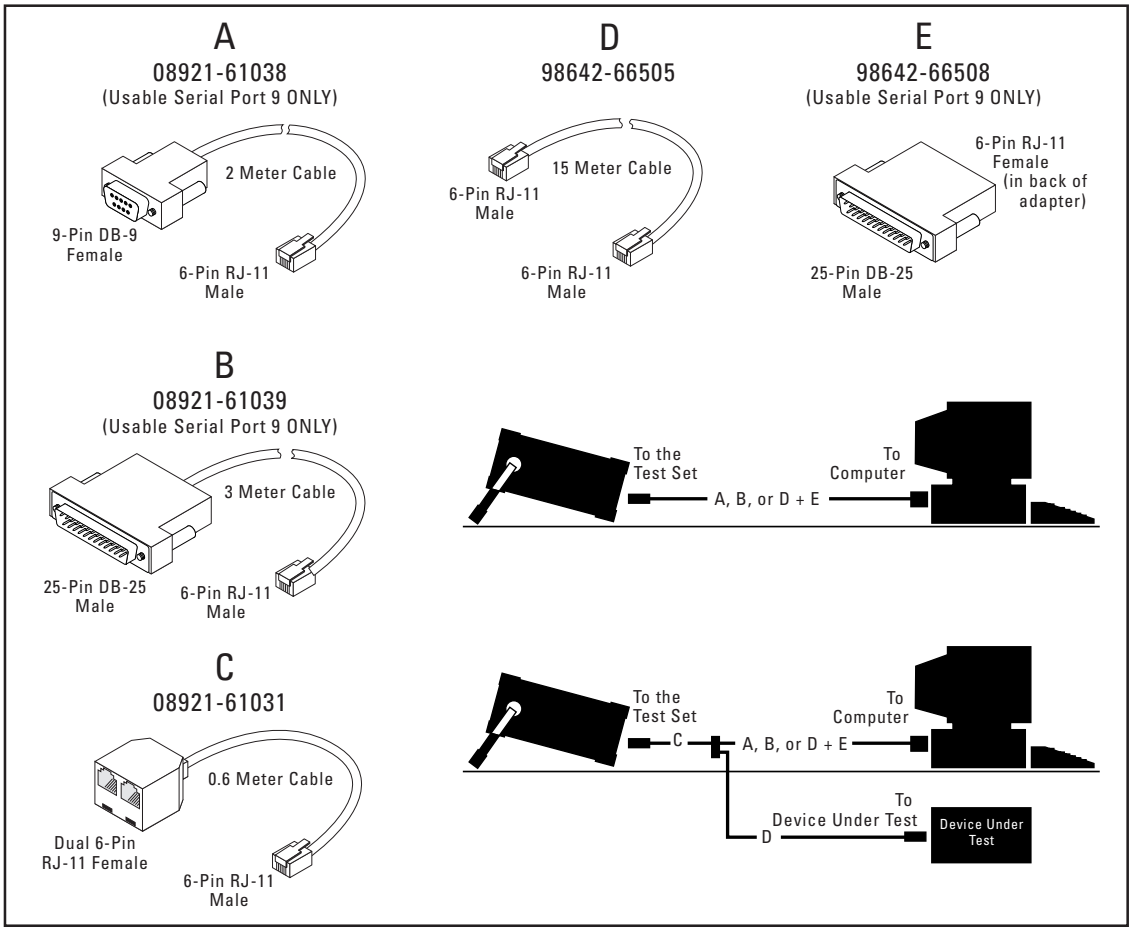


Figure 2-1. Available HP RS-232 serial cables and adapters

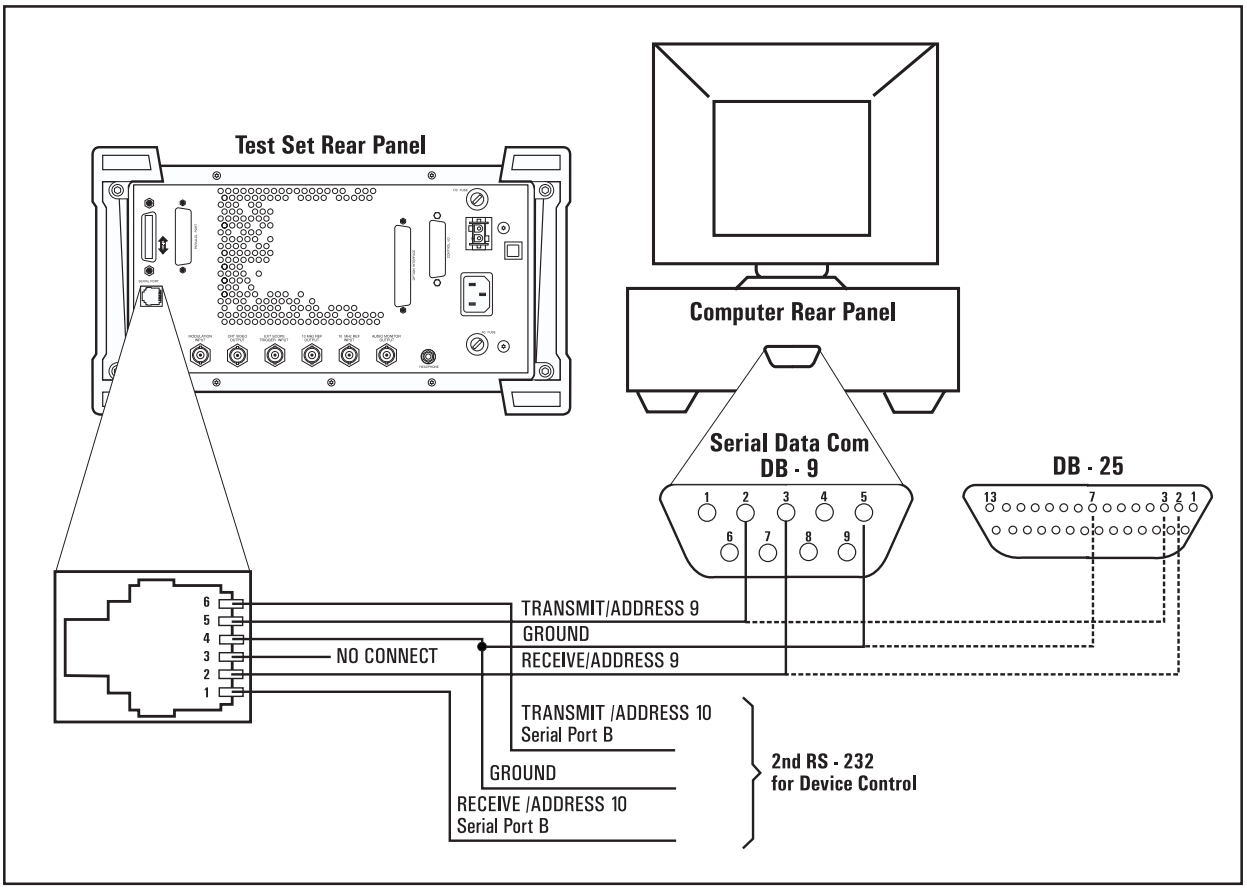


Figure 2-2. Connecting the test set serial port to a PC or terminal

## Test Set Serial Port 9 Screen Configuration

Table 2-2 and the following paragraphs describe how to set up the test set and the implications of the various choices.

1. Under the **To Screen** menu, select ***MORE***, then select ***IO CONFIG.)***
2. This will take you to the **I/O CONFIGURE** screen.
3. Using Table 2-2, set the fields to match your PC or terminal settings for serial port 9. (These test set settings will not change even if you press the **(PRESET)** key, send the *RST programming command, or turn the power on and off). Make sure you set up your PC or terminal with a matching configuration (except for the character echo feature as mentioned for **Inst Echo**).
4. Set **Serial In** to ***Inst*** to configure serial port 9 to the **IBASIC Command Entry** field.
5. Set **IBASIC Echo** to ***On*** to enable IBASIC character output commands like List or Print to always echo characters to serial port 9.
6. If you leave **IBASIC Echo *On***, you can see your programs on your PC or terminal when you list.
7. Another method to output characters to your PC or terminal when listing is to execute the IBASIC command PRINTER IS 9. After executing this command, whenever you execute other printer commands, characters will come out serial port 9. I/O address 9 is the test set's serial port 9 address. I/O address 1 is the test set's display address. I/O address 1 is also the default value for the PRINTER IS command, so all program printer output defaults to the test set's display.

8. If you set **Inst Echo** to ***On***, you can see characters echoed back to your PC or terminal as they are sent to the test set.
9. With test set's **Inst Echo** ***On***, do not use the echo feature of your PC or terminal serial port configuration or all the characters on your display will be duplicated.

## More about Receive and Transmit Pacing

When receiving characters into the **IBASIC command entry** field, the test set's microprocessor responds to each entry so very little buffer space is used even at the highest baud rate. Therefore, when using your PC or terminal to send characters to the **IBASIC command entry** field, it is permissible to set **Rcv Pace** and **Xmt Pace** to ***None***.

When sending characters over the serial port from the test set to slower devices like printers that may have small buffers, it is important to set **Rcv Pace** and **Xmt Pace** to ***Xon/Xoff***. This allows the printer to stop the test set data flow when the printer's buffer is full and then start it again when the printer is ready.

The test set has a serial port input buffer length of 2000 characters (with firmware revision A.09.04 and greater). This becomes important when you write IBASIC programs that expect to receive large amounts of data over the serial port with a single ENTER statement. See Chapter 3 of this product note for more information.

| Field | Available Settings | Recommended Setting |
|---|---|---|
| **Serial In** | Inst/IBASIC | ***Inst*** |
| **IBASIC Echo** | On/Off | ***On*** |
| **Serial Baud** Rate | 150, 300, 600, 1200, 2400, 4800, 9600, 19200 | ***9,600*** |
| **Parity** | None, Odd, Even, Always 1, Always 0 | ***None*** |
| **Data Length** | 7 bits, 8 bits | ***8 bits*** |
| **Stop Length** | 1 bit, 2 bits | ***1 bit*** |
| **Rcv Pace** (receive pacing) | None, Xon/Xoff | ***Xon/Xoff*** |
| **Xmt Pace** (transmit pacing) | None, Xon/Xoff | ***Xon/Xoff*** |

Table 2-2. Test set serial port 9 configuration

# PC Configuration

To prepare for IBASIC programming, you must first configure your PC to operate with the test set. This includes:

- Hardware
- Terminal emulator software

## PC Serial Port Configuration

Refer to Figure 2-2 for connection details. Connect the test set's serial port 9 to a serial I/O (input/output) port on your PC. On many PCs, a serial port is available as either a 25-pin DB-25 (female) connector or a 9-pin DB-9 (male) connector. This port can be configured as COM1, COM2, COM3, or COM4 (communications port 1, 2, 3, or 4) depending on the installed PC hardware and user-defined setup. Follow the instructions with your PC for this hardware and software configuration information.

## PC Configuration of Four Available Terminal Emulators

One convenient method to send ASCII characters to the test set uses a "terminal emulator" program which runs on your PC. A terminal emulator is an application program running on the PC that communicates with one of the serial communication ports installed in most PCs. It provides a bi-directional means of sending and receiving ASCII characters to the test set's serial port.

In general, a terminal emulator enables the PC to act like a dedicated computer terminal. This type of terminal was used before PCs to allow remote users to communicate via RS-232 with central mainframe computers. An ANSI-compatible terminal like the Digital Equipment Corporation VT-100 can be used to directly communicate with the test set. PC terminal emulation application programs have been designed to have setup fields much like these older technology terminals.

## Setting up Microsoft®[2] HyperTerminal on your PC (Windows 95, 98)

1. From Start select Programs, Accessories, Hyperterminal then HyperTerminal.
2. Enter a name and choose an icon of your choice, select OK.
3. In the Connect To screen, change Connect Using to COM1 or COM2 depending on which serial port you are using on your PC.
4. The next screen will allow you to set up the port settings. The default instrument settings are: Baud 9600, Data Bits 8, Parity None, Stop Bits 1, Flow Control Xon/Xoff.
5. Then choose File, Properties, choose the Settings Tab.
6. In the Settings Tab select VT 100 for the emulation field.
7. Select ASCII Setup, check Echo typed characters locally, then OK, then OK again.
8. Then save this file for future use.

## Setting up Microsoft Windows® Terminal on your PC (version 3.1)

1. From the Program Manager, select the Accessories Group
2. Select the Terminal icon
3. From the Settings menu, make the following choices:

**Select Terminal Emulation**

1. DEC VT-100 (ANSI).

[2] Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corp.

**Select Terminal Preferences**

1. Terminal Modes
   Line Wrap: **Off**
   Local Echo: **Off**
   Sound: **Off**
2. Columns: **132**
3. CR->CR/LF
   Inbound: **Off**
   Outbound: **Off**
4. Cursor
   **Block**
   Blink: **On**
5. Terminal Font: **Fixedsys**
6. Translations: **None**
7. Show Scroll Bars: **On**
8. Buffer Lines: **100**
9. Use Function, Arrow, and Ctrl Keys for Windows: **Off**

**Select Text Transfers**

1. Flow Control: **Standard Flow Control**
2. Word wrap Outgoing Text at Column: **Off**

**Select Communications**

*(the first five of the following communications choices for your PC's serial port should match your test set's settings)*

1. Baud Rate: **9600**
2. Data Bits: **8**
3. Stop Bits: **1**
4. Parity: **None**
5. Flow Control: **Xon/Xoff**
6. Connector: **COM1, COM2, COM3,** or **COM4** depending on your PC setup
7. Parity Check: **Off**
8. Carrier Detect: **Off**

## Setting up ProComm[3] on your PC (revision 2.4.3)

ProComm is a general purpose telecommunications software package for PCs with MS-DOS®[4]. One of its functions is to provide an RS-232 terminal function on a typical PC.

Running ProComm in MS-DOS. (You can use ProComm's built-in help function to learn more about setting it up).

1. To access the help and command functions, press the **(Alt)** and **(F10)** keys simultaneously [abbreviated as **(Alt) (F10)**]).
2. Press the **(SPACE BAR)** to move among the choices for a particular field.
3. Press **(ENTER)** to accept the displayed choice.

**Setting up the ProComm software**

1. Press **(ALT) (P)** to access the LINE SETTINGS window.
2. Enter the number **11**.

   This will automatically set:
   Baud rate: **9600**
   Parity: **None**
   Data Bits: **8**
   Stop Bits: **1**
   Selected communications port: **COM1**
   (This may be different on your PC)

[3] ProComm is a product of DATASTORM TECHNOLOGIES, INC.
[4] MS-DOS® is a U.S. registered trademark of Microsoft Corp.

3. To select a different communications port, enter the following numbers:
   20: **COM1**
   21: **COM2**
   22: **COM3**
   23: **COM4**
4. Enter the number **24** to save changes, to make the new configuration your default, and to exit LINE SETTINGS.
5. Press **(Alt) (S)** for the SETUP MENU
6. Enter the number **1** for MODEM SETUP
7. Enter the number **1** for the Modem init string
8. Press **(Enter)** to set a null string
9. Press **(Esc)** to exit MODEM SETUP back to the SETUP MENU
10. Enter the number **2** for TERMINAL SETUP
11. Terminal emulation: **VT-100**
    Duplex: **FULL**
    Flow Control: **XON/XOFF**
    CR translation (in): **CR**
    CR translation (out): **CR**
    BS translation: **NON-DEST**
    BS key definition: **BS**
    Line wrap: **ON**
    Scroll: **ON**
    Break length (ms): **350**
    Enquiry (CNTL-E): **OFF**
12. Press **(ESC)** to exit Terminal Setup back to the Setup Menu
13. Enter the number **4** for General Setup
    Translate Table: **OFF**
    Alarm sound: **OFF**
    Alarm time (secs): **1**
    Aborted downloads: **KEEP**
14. Press **(ESC)** to exit General Setup back to the Setup Menu
15. On the Setup Menu, press **(S)** to save your entries
16. Press **(ESC)** to exit the Setup Menu
17. Press **(ALT) (X)** to exit ProComm back to MS-DOS

## Setting Up HP AdvanceLink on your PC (HP 68333F version B.02.00)

HP AdvanceLink is software that enables you to use your PC as an alphanumeric or graphics terminal. It can also automate terminal and file-transfer functions. The version described will work with PC's with the MS-DOS or PC-DOS operating systems. (AdvanceLink for Windows is also available, and configuration is very similar).

**Running AdvanceLink in MS-DOS**

1. Press the **(TAB)** key to move from one field to the next, which also accepts the displayed choice.
2. Press the **(NEXT CHOICE)** and **(PREVIOUS CHOICE)** keys to move among the choices for a particular field.

**Setting up the AdvanceLink software**

1. Press the **(TERMINAL)** function key
2. Press **(CONFIG KEYS)**
3. Press **(GLOBAL CONFIG)**
   Keyboard: **USASCII**
   Personality: **ANSI**

Language: **ENGLISH**
Terminal Mode: **Alphanumeric**
Remote To: **enter your PC's selected serial port number, often Serial 1**
Printer I/F: **None**
Memory Size: **32K**
Video Type: **select your display type**
Forms Path: **no entry**
Screen Size: **select your size — 23 or 24**

4. Press **(DONE)** to return to the Config Screen
5. Press **(REMOTE CONFIG)** (to set up the Serial port you selected above in Remote To)

   Baud Rate: **9600**
   Parity/DataBits: **None/8**
   Enq Ack: **NO**
   Asterisk: **OFF**
   Chk Parity: **NO**
   SR(CH): **LO**
   Recv Pace: **Xon/Xoff**
   CS(CB)Xmit: **NO**
   XmitPace: **Xon/Xoff**
6. Press **(DONE)** to return to the Config Screen
7. Press **(TERMINAL CONFIG)**

   Terminal Id: **2392A**
   LocalEcho: **OFF**
   CapsLock: **OFF**
   Start Col: **01**
   Bell: **ON**
   XmitFnctn(A): **NO**
   SPOW(B): **NO**
   InhEolWrp(C): **NO**
   Line/Page(D): **LINE**
   InhHndShk(G): **NO**
   Inh DC2(H): **NO**
   Esc Xfer(N): **YES**
   ASCII 8 Bits: **YES**
   Fld Separator: **down arrow or US**
   BlkTerminator: **up arrow or RS**
   ReturnDef: **musical note or CR**
   Copy: **Fields**
   Type Ahead: **NO**
   Row Size: **160**
   Host Prompt Character: **left arrow or D1**
   Horiz. Scrolling Increment: **08**
6. Press **(DONE)** to return to the Config Screen
7. Press **(DONE)** to return to the Terminal Screen
8. Press **(MAIN)** to return to the Main Screen
9. Press **(EXIT ADVLINK)** to exit

## Terminal Configuration

Use the cable information in Table 2-1 and Figure 2-1 earlier in this chapter for connecting to your terminal. Terminals typically have a DB-25 (male) connector.

Set your terminal for DEC VT-100 ANSI emulation. Many ASCII terminals will also function properly.

To set up your terminal, use the field settings found in the HP AdvanceLink terminal emulator section found earlier in this chapter. As a minimum, make sure your terminal's basic setup information matches the fields on the test set's **I/O CONFIGURE** screen (see Table 2-2 earlier in this chapter for recommended settings).

Chapter Three

# Chapter 3
# Choosing your Development Method

There are three fundamental ways of developing programs for the test set. See Figure 3-1 below.

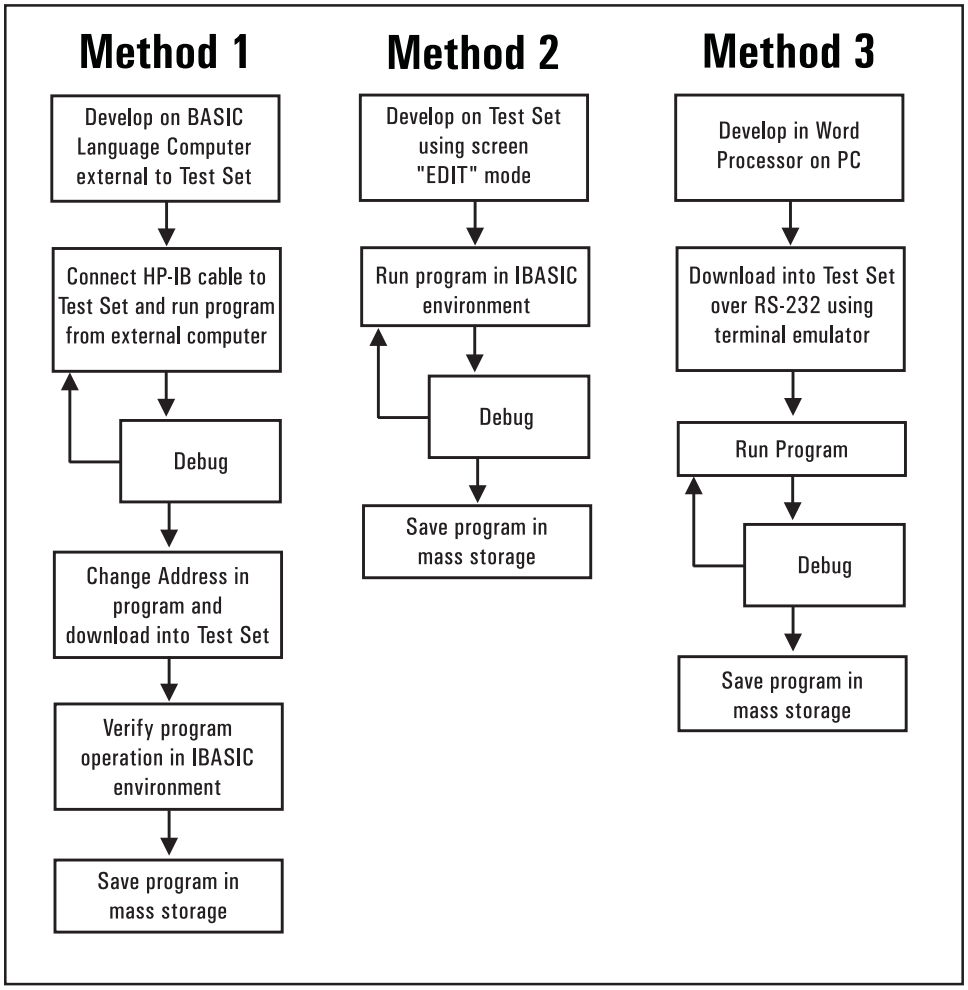


Figure 3-1. Three possible development methods

**Method 1**

If you have a BASIC language computer (either an HP technical computer or a PC running BASIC with HP-IB), this is the best method for developing any program. This is primarily true because it is easier to debug your program on the external computer before downloading it into the test set. When doing this, you can watch the test set's display show changes in state and easily verify the correct measurements. In contrast, when running IBASIC programs on the test set's internal controller, the test set displays only the IBASIC screen, not the individual screen states as the program progresses. This makes troubleshooting larger programs more difficult using Methods 2 or 3.

**Method 2**

If you do not have a BASIC language computer, you will most likely be using a PC with DOS interfacing over RS-232 as explained in Chapter 2. In general, it is easier to use a PC to develop on-line using the test set and a full-screen "EDIT" mode because you can easily and quickly run your program at various stages of development. In this case, the program always resides in the test set and can be run at any time. Mass storage is usually RAM memory.

**Method 3**

The third method of developing is to use a word processor on a PC with RS-232, and then download your program into the test set when you want to run. This is the last choice for development because downloading code into the test set over RS-232 requires a loader utility program running in the test set and a RAM memory card present as an intermediate storage location before running the program. Also, no syntax is checked until you run the program after downloading. For shorter programs, the intermediate storage location is not necessary.

# Method 1

## Developing Programs on a BASIC Language Computer Outside the Test Set

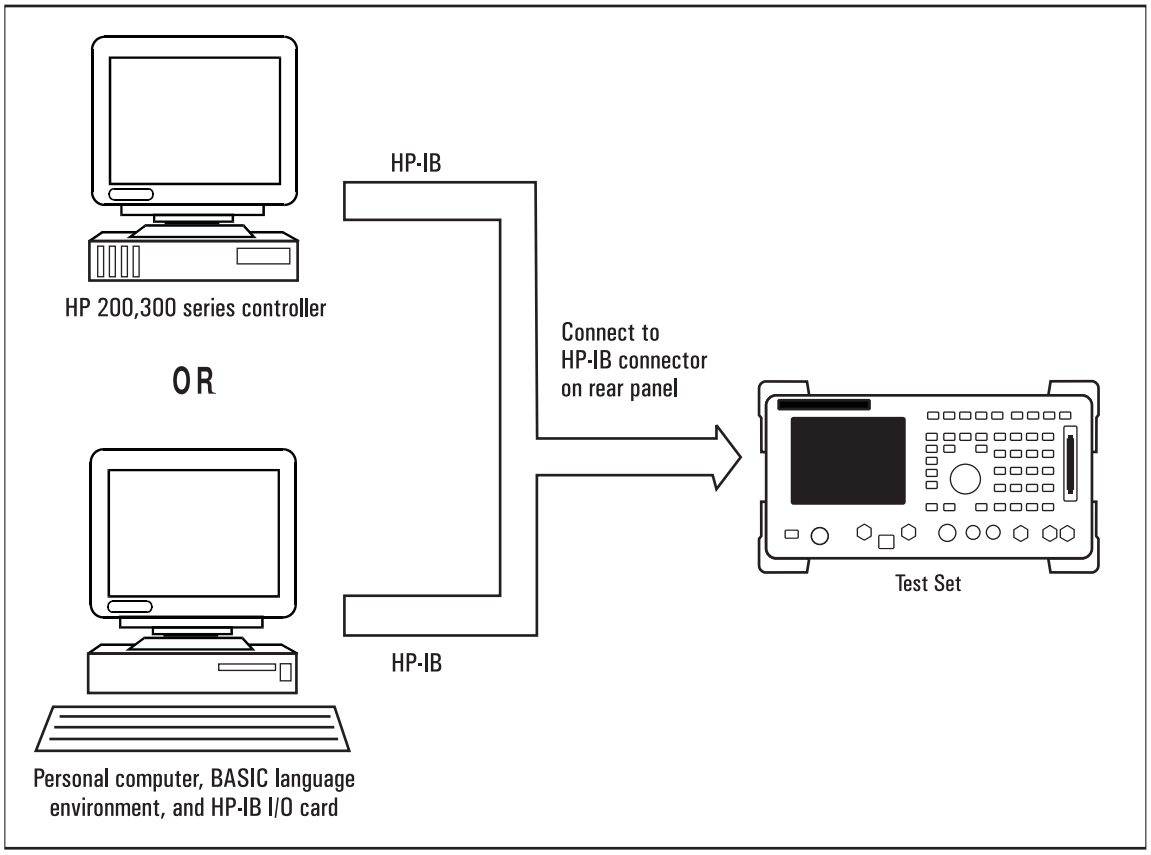


Figure 3-2. Connecting BASIC language computers to the test set

## HP-IB Interface Hardware and Screen Configuration

To use HP-IB (the IEEE 488.2 interface bus) as a means of communicating with the test set, connect a standard HP-IB cable (such as the HP 10833B) between the test set's rear-panel HP-IB connector and the HP-IB connector on your PC.

**On the test set:**

1. Select the **I/O CONFIGURE** screen.
2. Set the **Mode** field to ***Talk&Lstn***.

   **Note:** If this field is set to ***Control***, there will be a controller conflict between your computer and the test set, probably resulting in either an interface status error or "lock up" of your computer.
3. Set the **HP-IB Adrs** field to the desired address for the test set. The default value is ***14***.

## Connecting BASIC Language Computers to the Test Set via HP-IB

As shown in Figure 3-2, there are two choices for computers with this development method. The preferred computer is the HP 200,300 series technical computer. If this is not available, the recommended computer is any PC running a BASIC language environment such as HP BASIC for Windows. Also available is the HP 83224A IBASIC Developer's Tool Kit which runs in the HP BASIC for Windows environment and is used to develop programs specifically for the HP 8920A or HP 8921A.

**Note:** IBASIC is a subset of HP Rocky Mountain Basic. All IBASIC commands are compatible with RMB and thus will execute from a remote BASIC language computer. There are features unique to the test set's IBASIC controller.

They include:

- User keys
- Knob
- Line drawings on the test set display
- Screen printing on test set display
- RS-232 I/O
- Test set command line address (19)
- Dedicated internal HP-IB bus (select code 8)

You first develop your program on the computer using IBASIC compatible commands, then run it from the external computer over HP-IB for debugging. Once the program is working properly, you then download it into the test set so that it can run on the internal IBASIC controller. Because the IBASIC controller communicates with the test set's measurement functions over an internal bus with select code 8, you need to make sure to provide a means of easily changing the HP-IB select code from 7 to 8. Before downloading the program into the test set, change all test set HP-IB commands to address 800 from the test set default address of 714 used with the external computer.

There are two ways of allowing easy conversion of all HP-IB commands to a different address. The first way is to establish a variable to which you assign the correct 3-digit address number.

For example:

10 **Addr = 714!** Sets the value of variable Addr to be 714.

20 **OUTPUT Addr;"*RST" !** Commands the test set to reset at address 714.

To change the address, simply change the value of variable **Addr** to **800**.

A second method is to assign an I/O path to the desired I/O port.

For example, to control device #14 on the port with select code 7:

10 **ASSIGN @Device TO 714!** Establishes I/O path to select code 7 address 14.

20 **OUTPUT @Device;"*RST"!** Commands test set to reset at address 714.

To change the address, simply change **line 10** to:

10 **ASSIGN @Device TO 800**.

## Downloading your Program into the Test Set over HP-IB

IBASIC PROGram subsystem commands have been developed to allow the external BASIC controller to download your program to the test set over HP-IB. Four commands from the remote controller are necessary to transfer the program. You execute these commands serially allowing enough time for each command to finish executing. (The test set's **HP-IB Mode** must be set to ***Talk&Lstn***, and the **TESTS (IBASIC CONTROLLER)** screen must be displayed).

1. OUTPUT 714**;"PROG:DEL"**
   Deletes any programs that reside in test set RAM.
2. OUTPUT 714**;"PROG:DEF #0"**
   Defines the address in test set RAM where the downloaded program will be stored.
3. LIST **#714**
   Causes all program lines to transfer over HP-IB to the test set which is at address 714.
4. OUTPUT 714**;" "END**
   Defines end of download process by generating an EOI command.

At this point the program code will be in the test set ready to run. If you find bugs at this point, you can upload the program back into the remote controller or use the full screen "EDIT" function over RS-232 explained later.

After the program is working properly in the test set IBASIC environment, you should **Save** it to your choice of **Mass Storage**.

## Uploading your Program from the Test Set to a Remote BASIC Language Computer over HP-IB

To upload a program, you need to enter into your external controller the following short program that uses a command from the PROGram subsystem to initiate the upload. The uploaded program is directed to a file on your PC in the HP BASIC directory. If you are using an HP 200,300 series BASIC language computer, change the target file appropriately. After running this program, the program code will be in the designated file and you will have to GET it for editing in the HP BASIC (or RMB) environment.

```
10   ! PROGRAM TO UPLOAD IBASIC CODE FROM A TEST SET TO BASIC
       COMPUTER  VIA HP-IB.
20!###########################################################################
30   !
40   ! The file for uploaded code will be "C:\htb386\code".
50   ! If you want to use a different file or directory, modify the two lines
60   ! with the labels "File_name_1" and File_name_2".
```

```
70  !
80  !#############################################################
90  Addr=714                                    !HP 8920 address on HP-IB
100 ALLOCATE Line$[200]
110 PRINTER IS 1
120 CLEAR SCREEN
130 DISP "It may be several minutes before code begins transferring if your
      program is long"
140 OUTPUT Addr;"*RST"                          !Reset the HP 8920
150 OUTPUT Addr;"DISP TIB"                      !Displays the IBASIC screen
160 OUTPUT Addr;"PROG:EXEC 'CLS'"               !Clears the 8920 display
170 OUTPUT 714;"PROG:DEF?"                      !Initiates the upload of whole program
180 ENTER Addr USING "X,D,#";Count_len          !Number of lines in program
190 ENTER Addr USING VAL$(Count_len)&"D,#";Char_count
                                                !Number of characters
200 !
210 File_name_1:  CREATE ASCII "C:\htb386\code",(1.05*Char_count/256)+5
                                                !Number of records reserved for upload.
220 File_name_2:  ASSIGN @File TO "C:\htb386\code"
230 !
240 DISP "Transferring code from HP 8920"
250 LOOP                                        !Program transfer loop.
260   ENTER Addr;Line$                          !CR/LF terminates each line.
270   PRINT Line$                               !Displays new lines on HP 8920 CRT.
280   OUTPUT @File;Line$                        !Transfer new line to file.
290   Char_count=Char_count-LEN(Line$)-2 !Reduces Char_count by the number of
                                                characters in current line.
300 EXIT IF Char_count<=0
310 END LOOP
320 !
330 ASSIGN @File TO *                           !Cleans out file buffer.
340 ENTER Addr;Line$                            !Close off reading
350 CLEAR SCREEN
360 DISP "Transfer complete."
370 LOCAL Addr
380 END
```

# Method 2

## Developing Programs on the Test Set Via RS-232 Using the "Edit" Mode

The test set's IBASIC controller has a built-in editor that is interactive with a terminal or PC over the RS-232 serial port. (It does not work unless you have a terminal or PC terminal emulator connected to serial port 9.) The editor, referred to here as the "edit" mode, allows you to develop code directly in the test set with no uploading or downloading. You can use the edit mode to begin writing programs from scratch or to modify existing programs. Refer to Chapter 2 for information about connecting a PC or terminal to the test set.

## Selecting the IBASIC Command Entry Field

To use serial port 9 for program development, you must access the **IBASIC command entry** field. Serial port 9 is directly interfaced with the **IBASIC command entry** field. An IBASIC command of ASCII characters sent into serial port 9 will appear on the **IBASIC command entry** field. When a carriage return/line feed is encountered, the command will be executed. To select the **TESTS (IBASIC CONTROLLER)** screen on the test set:

1. Press the **(TESTS)** key. The **TESTS** (main menu) screen is displayed.
2. Position the cursor and select the **IBASIC (IBASIC Cntrl)** field on the screen's lower right.
3. The screen titled **TESTS (IBASIC CONTROLLER)** is displayed.


To access the **IBASIC command entry** field:

1. Position the cursor on the screen's upper left. This is the **IBASIC command entry** field.
2. The **IBASIC command entry** field is not titled like others in the test set — it is the highlighted horizontal 2-line "bar" just below the screen title **TESTS (IBASIC CONTROLLER)**.

To use the **IBASIC command entry** field with the test set knob:

1. Position the cursor at the **IBASIC command entry** field and push the knob.
2. By rotating the knob, you can scroll through a long list of ASCII characters displayed on the right side of the screen.
3. When the cursor is next to the character you want, push the knob to select that character.
4. No external hardware is required for this entry method, but it is tedious and is recommended only for short commands. Use this method when doing simple tasks such as initializing memory cards or CATaloging a memory card.
5. Program development using the knob alone is not recommended.

## Entering and Exiting the Edit Mode

You enter the edit mode by placing the cursor on the test set's IBASIC command line, typing the word "edit" on the connected PC, and pressing the PC's **(ENTER)** key. At this point the test set will fill the PC screen with 22 lines of IBASIC code from the current program in the program memory. No program line will display on the test set screen. If no program is present, it will display 10 for line #10 to allow you to begin writing IBASIC lines beginning at line #10. The test set IBASIC screen will display "*" in the upper right hand corner indicating that the IBASIC controller is running to support the full screen edit mode.

After your editing is complete, you exit the edit mode by pressing the PC's **(ESCAPE)** key twice or pressing **(SHIFT)** **(CANCEL)** on the test set.

## IBASIC Edit Commands (Function Keys)

A variety of editing commands are supported by the test set edit mode. These commands are activated in the test set as escape code sequences. Most terminals and PC terminal emulator programs allow you to configure softkeys (function keys) with user defined escape code sequences and user defined labels for the keys. An escape command (when received by a peripheral device like a printer or the test set) causes the peripheral to recognize subsequent ASCII characters differently. In the case of the test set, escape sequences are used for executing IBASIC program editing commands.

For instance, ESCAPE [ L makes the test set insert a new line number where the cursor is positioned. Table 3-1 lists editing escape codes for the test set. Choose and use the ones you need. There is no escape code for DELETE CHARACTER. Use the **(Backspace)** key for deleting. Also, use the arrow keys to position the cursor.

| Function Key Names | Windows Terminal Escape Codes | HP AdvanceLink Escape Codes |
|---|---|---|
| INSERT LINE | ^[[L | (-[L |
| DELETE LINE | ^[[M | (-[M |
| GO TO LINE | ^[g | (- g |
| CLEAR LINE | ^[[K | (-[K |
| PAGE UP | ^[OQ | (-OQ |
| PAGE DOWN | ^[OR | (-OR |
| RECALL LINE | ^[r | (-r |
| BEGIN LINE | ^[OP | (-OP |
| END LINE | ^[OS | (-OS |

Table 3-1. Edit mode escape code commands

## Setting up Function Keys in Microsoft Windows Terminal

When in the TERMINAL mode, click on **Settings**, then **Function Keys**.
^[ is ESCAPE in Windows Terminal. See Table 3-1 for the escape codes.

**Note:** Windows Terminal seems to work best when you use a mouse to access the function keys, not the keyboard. Also, scrolling a program works best when you maximize the terminal window display.

Refer to page 20 for instructions on setting up Microsoft HyperTerminal on your PC (Windows 95, 98 version).

## Setting up Function Keys in HP AdvanceLink

- From the "Main" (highest level) screen, set up the 8 softkeys as follows:
  1. Display User Definition Screens: by pressing **(Cntrl)** **(F9)**
  2. Enter all the LABEL titles for **K1** through **K8**.

3. Activate the **"Display Function"** feature by pressing softkey (**F7**).
4. Now you can enter the escape codes for each edit command aligned with the soft key definitions you just entered. With the **"Display Functions"** key pressed, when you press the escape key, a left arrow will be displayed.

- Once you have set up all 8 keys, you activate them by pressing **(Shift) (F12)**. To deactivate your user defined softkeys, press **(F12)**.
- (- is ESCAPE in HP AdvanceLink. See Table 3-1 for the escape codes.

## Setting up Function Keys in ProComm

ProComm does not have function keys. However, you can assign escape sequences to number keys **1** through **0** by using the Keyboard Macro function. You access this function by keying **(Alt) (M)**. There is no way to display a label so you have to record the labels elsewhere. See the ProComm manual for further information.

# Method 3

## Developing Programs in a Word Processor on a PC

The third way of developing an IBASIC program for the test set is to write the program using a text processor on a PC, save it as an ASCII file, then download it into the test set through the serial port. The benefit of this method is that you can do it on your PC alone without connecting a test set until download and you do not need a BASIC language compiler. The primary drawback is that no syntax checking occurs until you download the complete program and first run it. A second drawback is that, especially for longer programs (>100 lines), it is time-consuming to transfer the code into the test set when you want to run it.

## Setting up a Word Processor

The word processor on which you develop IBASIC code must be able to save the file in ASCII format. This is necessary because word processors use a variety of escape codes to mark all the special display formats such as bold face, font size, indented text, and the like. When you store your text processor file as ASCII format, all escape codes are stripped off. (As explained in the Appendix, the test set stores IBASIC code either as PROGRAM files or as ASCII files. The ASCII format is available in most text processors.)

**Note:** If you have a BASIC language operating system running on DOS such as TransEra® HT Basic™, you can use the GET statement to bring in programs developed on text processors which were saved as an ASCII file. Then follow the steps for downloading described earlier in this chapter in Method 1.

When writing the program, follow these steps to make sure the test set will accept the code when you download it.

## Writing Lines of IBASIC Code on the Word Processor

1. Always begin new lines at the far left margin. Never use a leading space or tab.
2. Number each consecutive line just like an IBASIC language program. Typically begin with 10 and increment by ten for each consecutive line.
3. Do not leave any space or double space between lines.
4. Make sure to use hard carriage return, line feeds (CR/LF) at the end of each line.

5. When saving the completed program, save it as an ASCII file. Some word processors have ASCII options where you need to specify CR/LF at the end of each line. It is important that each line end with a carriage return, line feed.
6. Experiment with a short program first to make sure everything is working correctly.

## Transferring Programs from your Text Editor into the Test Set

### Programs 100 Lines or Shorter

For short (<100 lines) programs, it is convenient to use an ASCII file transfer program running on your PC to send your program down to the test set over RS-232 a line at a time directly into the IBASIC program entry field. In this case, you set up the test set to receive serial ASCII characters by positioning the test set cursor at the **IBASIC command entry** field as explained in Method 2 in the section titled "Selecting the IBASIC Command Entry Field". With this setup, when ASCII characters are sent, they appear on the **IBASIC command entry** field. When a carriage return, line feed is received, the test set will parse the line into the IBASIC program memory. Each line takes about two seconds to scroll in and be parsed. As you can see, this becomes very time consuming for long programs. An alternative for longer programs is discussed later.

Start off by making sure there is no program in the IBASIC program memory by executing a **SCRATCH A** command.

For the "short" program transfer method, using Microsoft Windows Terminal, select the Hyperterminal application in the Accessories Group. Set it up as described in Chapter 2 with the following additions:

1. Select **Settings**, **Text Transfers, Flow Control, Line at a Time, Delay Between Lines 25/10 Sec, Word Wrap Outgoing Text at Column: Off.**
2. Select **Transfers, Send Text File, Following CR: Strip LF** selected; **Append LF** not selected.
3. Select the text file you want to transfer and begin the transfer by selecting **(OK)**.
4. Make sure the test set display cursor is in the upper left of the **IBASIC command entry** line.

At this point, the **IBASIC command entry** line will intensify and characters will scroll in left to right. As each line is finished you will see the IBASIC run time indicator (upper right) come on for about 0.5 seconds as the line is parsed. If another line is sent before this parsing is complete, the test set will beep indicating an error, and the next line of the transfer will be rejected. If this happens, you need to halt the transfer, increase the delay between line to a slightly higher number, and start the transfer again from the beginning. When all lines have transferred, you can list the program to verify that it was completely received. At this time, the program is ready to run. You can either key in RUN from your terminal emulator or press the **(K1)** Run user key in the **TESTS (IBASIC CONTROLLER)** screen. Make sure to not press the Run Test user key in the **TESTS** (main menu) screen as this will scratch the program you just loaded and look to the memory card for a different program.

### Programs Longer Than 100 Lines

For longer programs (greater than 100 lines), transferring the ASCII text files directly into the IBASIC program memory via the RS-232 serial bus is too time consuming. To speed the process up, perform a two step process.

1. Transfer the ASCII text file directly to a test set mass storage location (typically a RAM card).
2. Perform a **"GET"** command to bring the program from mass storage into the IBASIC program memory.

To perform the ASCII text file transfer, an IBASIC program running in the test set is required to manage the transfer. A suitable program titled “ASCII_DN” (for ASCII downloader) is listed below.

```
10  ! ASCII_DN
20  ! Program to download ASCII program file from PC to the test set via RS-232
30  ! ###################################################################
40  !
50  ! This program must be loaded into the test set and run on the test set.
60  ! It directs ASCII characters that come in the serial port 9 to a file
70  ! named “TEMP_CODE” on an SRAM card. After the transfer is complete,
80  ! you must SCRATCH this program and GET the transferred program from
90  ! the “TEMP_CODE” file.
100 !
110 ! ###################################################################
120   COM /File_name/ File_name$[10]
130   DIM In$[200]
140   File_name$=”TEMP_CODE”                    ! File name on RAM card
150   CLEAR SCREEN
160   CLEAR 9                                   ! Clears HP 8920 serial bus
170   OUTPUT 800;”*RST”
180 ! Set up HP 8920 serial bus to receive ASCII text file
190   OUTPUT 800;”CONF:SPORT:BAUD ‘9600’;PAR ‘None’;DATA ‘8 Bits’”
200   OUTPUT 800;”CONF:SPORT:STOP ‘1 Bit’;RPAC ‘Xon/Xoff’;XPAC ‘Xon/Xoff’”
210   OUTPUT 800;”CONF:SPORT:SIN ‘IBASIC’;IBECHO ‘OFF’”
220 CALL Code(File_name$,In$)
230 END
240 Purge_it:SUB Purge_it                       ! Purges File_name on card
250     COM /File_name/ File_name$
260     OFF ERROR
270     PURGE File_name$&”:INTERNAL”
280   SUBEND
290 Code:SUB Code(File_name$,In$)
300     ON ERROR CALL Purge_it                  ! Branches if CREATE statement
                                                returns error
310     CREATE ASCII File_name$&”:INTERNAL”,650
                                                ! Creates file on card
320     OFF ERROR
330     ASSIGN @File TO File_name$&”:INTERNAL”
340     PRINT TABXY(1,5);”Ready to receive ASCII file data.”
350     PRINT
360 Begin:ON TIMEOUT 9,1 GOTO Begin             !Loops until data begins coming
370     ENTER 9;In$
380     OUTPUT @File;In$
390     PRINT In$
400 Transfer:LOOP                               !Loops to bring in ASCII file line
                                                at a time
410     ON TIMEOUT 9,5 GOTO Done                !Exit loop if data stops for >5 sec.
420     ENTER 9;In$
430     PRINT In$
440     OUTPUT @File;In$
450     END LOOP
460 Done:ASSIGN @File TO *
470   CLEAR SCREEN
480 ! Returns HP 8920 serial port input to “instrument” allowing serial comm
490 ! via the IBASIC entry field.
500     OUTPUT 800;”CONF:SPORT:SIN ‘Inst’;IECHO ‘ON’;IBECHO ‘ON’”
510     PRINT TABXY(1,5);”Down load of ASCII file is complete.”
520   SUBEND
```

This program runs on the test set to direct ASCII characters coming in serial port 9 directly to a file on a RAM card titled **"TEMP_CODE"**. This program creates the **TEMP_CODE** file on the card with a size of 650 records (166 KB or enough for about 6600 lines of ASCII text — see the Appendix for memory considerations). When this program is running, it displays **"Ready to receive ASCII file data".** At that point, you need to initiate sending the ASCII text file representing your program from your computer over the RS-232 serial bus. Below are two ways to initiate sending an ASCII file from your PC. They both require you to have entered the **"ASCII_DN"** program into the test set's IBASIC controller memory. This can be done by either typing it in using the **EDIT** mode described earlier, or downloading it from an ASCII text file a line at a time as explained earlier.

## Sending ASCII Text Files over RS-232 with Windows HyperTerminal

Set up your Windows HyperTerminal software on your PC as explained in Chapter 2 of this product note. On the test set, enter and run the **ASCII_DN** download program in the IBASIC program memory (see "Transferring Programs from your Text Editor into the Test Set" immediately above in this chapter). You should now see the instruction **"Ready to receive ASCII file data"** on the test set. Now make sure your Terminal or HyperTerminal settings are configured correctly (see page 20).

At this point, you should see each line of your program rapidly scroll across the screen of the test set. When it is finished, the download program will display "**Down load of ASCII file complete.**"

Before running your transferred program, you must **SCRATCH** the IBASIC program memory to remove the **ASCII_DN** download program from the test set's memory.

Next, execute a **GET "TEMP_CODE"** command on the IBASIC command entry line. This will bring the ASCII text into the IBASIC program memory. It is now ready for execution.

## Sending ASCII Text Files over RS-232 with ProComm Communications Software

Set up your ProComm terminal emulator software on your PC as covered in Chapter 2 of this product note. On the test set, enter and run the **ASCII_DN** download program into the IBASIC program memory (see "Transferring Programs from your Text Editor into the Test Set" above in this chapter). You should now see the instruction **"Ready to receive ASCII file data"** on the test set. Now make the following settings in ProComm on your PC:

**Note:**
The way ProComm views the remote computer (test set controller) defines an upload as sending data from the host computer "up" to the test set. This is backwards from the way the download was described earlier in the Terminal example. So with ProComm we will be using its ASCII upload capability.

1. Press **(Alt)** **(F10)** to display the ProComm help screen.
2. Press **(Alt)** **(P)** to display the **SETUP MENU**.
3. Select item **6**, **ASCII TRANSFER SETUP**.
4. Set Echo locally **NO**
5. Expand blank lines **YES**

6. Pace character **0**
7. Character pacing **15**
8. Line pacing **10**
9. CR translation **NONE, LF**
10. Translation **NONE** (This is important since the default setting will strip line feeds and this will cause the transfer to never begin).
11. Now use the **(Escape)** key to get out of the set up mode and back to the main screen.
12. Press **(Alt)** **(F10)** to access the help menu.
13. To begin sending the file, select **PgUp.**
14. In the **UPLOAD screen,** select **7** ASCII protocol.
15. Run the **ASCII_DN** downloader program on the test set.
16. When the test set displays **"Ready to receive ASCII file data"**, press **(Enter**) on your PC and the transfer will begin. At this point, you should see each line of your program rapidly scroll across the screen of the test set. When it is finished, the download program will display "**Down load of ASCII file complete.**", and your program file will be on the memory card in the **"TEMP-CODE"** file.
17. Before running your transferred program, you must **SCRATCH** the IBASIC program memory to remove the **ASCII_DN** download program from test set memory.
18. Next you must execute a **GET "TEMP_CODE"** command on the IBASIC command entry line to bring the ASCII text into the IBASIC program memory. It is now ready for execution.
19. To run this program on the test set IBASIC controller, execute the **RUN** command.

## Uploading Programs from the Test Set to your PC

As an overview, you have to do the following:

1. The test set must output the program over the serial bus.
2. Your PC must receive the data through its serial port and direct the data to a file on disk. This can be done by a terminal emulator application such as Windows Terminal, ProComm, or HP AdvanceLink. This requires having the serial port connection established as explained in Chapter 2.

On the test set, set the cursor to the **IBASIC command entry** line. Execute the command **PRINTER IS 9** which sets serial port 9 as the default printer port. Later when you execute the **PRINT** command, ASCII characters will come out serial port 9.

On your PC, select **"Receive Text File"** in Windows Terminal or **"Receive Files"** (PgDn which is called Download) in ProComm. Enter a file name, then initiate the file transfer. The PC is now looking for ASCII text to come in the serial port.

The final step is to execute the IBASIC **LIST** command on the test set IBASIC command line. At this point, the program will list its contents out serial port 9 and be received by the terminal emulator software on the PC. When the listing is finished, you need to terminate the file transfer by selecting **Stop** on Windows or **Escape** on ProComm.

## I/O from IBASIC Programs

There are three ports providing I/O (input/output) to peripherals external to the test set: two serial ports and one HP-IB (IEEE-488.2) port. To bring data in, use the IBASIC **ENTER** command. To send data out, use the **OUTPUT** command. Please reference the *Test Set Instrument BASIC Programmer's Guide* for a complete description of these and other IBASIC commands.

## The Two Serial Ports

The test set uses a small RJ-11 female connector on the rear panel for two serial ports. This connector has six wires, 3 for serial port address 9 and 3 for serial port address 10. For basic information about the serial ports, see Chapter 2 of this product note. For a connection diagram, see Figure 2-2 in Chapter 2.

Before using either port, you have to set up the RS-232 protocol by setting baud rate, pacing, and the other settings explained in Chapter 2. Functionally from an I/O point of view, the two ports are identical. However, operationally there is one major difference. The serial port address 9 settings are adjustable on the **I/O CONFIGURE** screen or via IBASIC commands, while the serial port 10 settings are adjustable only via IBASIC commands. There is no screen for serial port 10 settings. For more information, see Chapter 3, Operating Overview of the Test Set's *User's Guide* which gives the syntax for serial port 10 commands.

## Example IBASIC Program using Serial Port 10

```
10  !......ASCII CHARACTER CYCLER.....
11  !......Prompts user over Serial Port 9. To see this prompt, you need to
12  !......be connected to a terminal at 9600 baud.
20  !......Outputs ASCII characters on Serial Port 10 beginning with ASCII
30  !......character 32 (space) ending with ASCII character 126 (~).
40  !......Characters are output with no CR/LF.
50  OUTPUT 9;"When you are ready for data to be sent on 10"
60  OUTPUT 9;"port, press Enter"
70  OUTPUT 800;"CONF:SPOR:SIN 'IBASIC';BAUD '9600'"!Allows IBASIC to
            read port 9.
80  DIM A$[10]
90  ENTER 9;A$  !Program hangs here until CR/LF comes in.
100 !.....................................................................
110 I=32
120 WHILE  I<=126
130 OUTPUT 10 USING "K,#";CHR$(I) ! Outputs characters all on one line.
140 I=I+1
150 END WHILE
160 OUTPUT 800;"CONF:SPOR:SIN 'Inst'" !Sets port 9 to IBASIC entry field.
170 EXECUTE ("CURSOR HOME")  !Places cursor at left of IBASIC entry field.
180 END
```

## More Serial Port 10 Information

Serial port 10 is sometimes called serial port B in test set documentation and programs.

The default serial port 10 settings are the same as serial port 9. They are:

1. Serial Baud rate: **9600**
2. Parity: **None**
3. Data Length: **8 Bits**
4. Stop Length: **1 Bit**
5. Receive and Transmit Pacing: **Xon/Xoff**
6. Serial in: **Not available for Port 10**
7. IBASIC and Instrument Echo: **Not available for Port 10**

There is no test set screen that shows serial port 10 settings. Therefore, to know serial port 10 settings, you must either set them or query them.

For example, you can query the baud rate of the serial port 10 with the following IBASIC program:

```
10 DIM Setting$[20]
20 OUTPUT 800;"CONF:SPB:BAUD?"  !Initiates a query.
30 ENTER 800;Setting$
40 DISP Setting$
50 END
```

This program returns a quoted string. For example, if the baud rate is set to 9600, the returned ASCII character string is "9600". These settings are held in non-volatile memory. They remain in place until you change them.

Chapter Four

# Chapter 4
# Common Mistakes and Helpful Hints

Because of the test set's wide functionality, a large body of IBASIC syntax has been defined to provide complete HP-IB control. The *Programmer's Guide* lists this syntax in a hierarchical presentation to help you understand the sequence you must follow in building command strings. The following section lists several helpful programming guidelines you can use to develop your program more quickly.

## Guideline #1:

### Suggested Measurement Sequence and Example Program

When taking readings from the test set always follow this general sequence:

1. Start from the **PRESET** state. This is done manually with the green **(PRESET)** key and over HP-IB with the ***RST** command. This initial step allows you to start making settings beginning with most fields in a known state. If you do not do this in your program, you will have to make many more settings to assure the desired instrument state.
2. Make your measurement manually on the front panel of the test set and record in sequential order the screens you select and settings within each screen.
3. Begin writing your program using settings you recorded from above. As your manual procedure requires changing screens, use the **DISPlay** command to select the desired screen followed by the correct syntax to set the desired field. As discussed in Chapter 3 of this product note, when programs are running in the IBASIC controller, you will not see the **"displayed"** screen since the IBASIC screen is always displayed. However, the IBASIC **DISPlay** command assures all settings and measurements you want in a particular screen are available. If you try to take a reading from a screen that is not made available by using the **DISPlay** command, the test set will return **HP-IB Error:-420 Query UNTERMINATED,** or **HP-IB Error: -113 Undefined header**.
4. The **ENCoder and DECoder** require a further level of syntax to display the desired screen. To display the desired **ENCoder** or **DECoder** screen, you first have to send **DISP ENC** then send **ENC: MODE 'DTMF'** for example.
5. Make sure measurement state is **ON**. This is the default preset condition for most measurements but if a previous program has set the state to **OFF**, the measurement will not be available. The program will halt with **HP-IB error:-420 Query UNTERMINATED.**
6. Trigger a reading if you are using single trigger. The default **trigger mode** is **repetitive** causing the test set to trigger itself. In this case, a separate trigger command is not necessary.
7. Send the **MEASure query** command to initiate a reading. This will place the desired reading in the HP-IB output register. If you are using **AF Analyzer measurements** of **SINAD**, **Distortion**, **Signal to Noise Ratio, AF Frequency**, **DC Level,** or **Current**, you must first **SELect** the appropriate measurement. (See HP-IB syntax diagrams for **MEASure**.)
8. Use the **ENTER** statement to transfer the reading to a variable or string array in the context of your program.

The following example illustrates how to make settings and then take a reading from the test set. This setup takes a reading from the spectrum analyzer marker after tuning it to the RF generator output frequency.

## Example Program

```
10  Addr=714
20  OUTPUT  714;"*RST"                          ! Preset to known state
30  OUTPUT Addr;"TRIG:MODE:RETR SING"           ! Sets single trigger
40  OUTPUT Addr;"DISP RFG"                      ! Selects the RF Gen screen
50  OUTPUT Addr;"AFG1:FM:STAT OFF"              ! Turns FM OFF
60  OUTPUT Addr;"RFG:AMPL   -66 DBM"            ! Sets RF Gen ampl to -66 dBm
70  OUTPUT Addr;"RFG:FREQ 500 MHZ"              ! Sets RF Gen freq to 500 MHz
80  OUTPUT Addr;"RFG:AMPL:STAT ON"              ! Turns RF Gen output ON if
                                                ! it was OFF.
90  OUTPUT Addr;"DISP SAN"                      ! Selects Spec Anal screen
100 OUTPUT Addr;"SAN:CFR 500 MHZ"               ! Center Frequency 500 MHz
110 ! ————————  MEASUREMENT SEQUENCE————————————
120 OUTPUT Addr;"TRIG"                          ! Triggers reading
130 OUTPUT  Addr;"MEAS:SAN:MARK:LEV?"           ! Query of Spec Anal marker level
140 ENTER  Addr;Lvl                             ! Places measurement result in
                                                ! variable Lvl
150 DISP Lvl                                    ! Displays value of Lvl
160 END
```

After preset, the **RF generator** is set to ***500 MHz***, ***-66 dBm*** from the **RF IN/OUT port** with **modulation turned off.**

Then the **spectrum analyzer** is set to the **same frequency as the generator** with **input from the RF IN/OUT port**. This allows the spectrum analyzer to measure the RF generator with no external connections. In the beginning the **trigger mode** is set to **SINGle** so that a **TRIGger command** is necessary during the measurement sequence. This program illustrates the proper sequence of making settings and taking readings from the test set.

The measurement taken is the level of the spectrum analyzer marker. The marker is always tuned to the center frequency of the spectrum analyzer display after preset. With the RF generator output and spectrum analyzer input both directed to the RF IN/OUT port, the two will internally couple with 46 dB of gain. So the measurement will read a level of approximately -20 dBm. (-66 +46) This is convenient for the demonstration since no external cables are required.

## Guideline #2

### If the Program Stops or "Hangs Up"

If your program stops or "hangs up" when trying to **ENTER** the measurement, it is likely that the measurement you're trying to read is not available. Here are several reasons that can happen.

1. You have not displayed the screen where your measurement is presented before querying the measurement.

2. The squelch control is set too high. If you are trying to make TX measurements and the test set is squelched, most measurements are unavailable.
3. The **RF Input** is set to ***ANT*** while trying to read TX power. TX power is not measurable with the input port set to antenna.
4. The input signal to the test set is very unstable causing the test set to continuously autorange. (You should see this when setting up the measurement manually).
5. While setting up the measurement manually, if the measurement field displays ——, this means the measurement is not available.
6. You have set single trigger mode **(TRIG:MODE:RETRig SINGle)** and have not triggered a new reading before the **ENTER** command.
7. You are trying to make an FM deviation or AM depth measurement while in the RX test screen or after executing instrument preset. In these cases, the test set is in the RX test screen where FM or AM measurements are not available.

One way to make modulation depth measurements available is to:

1. Display the **AF ANALYZER** screen **(DISP AFAN)**
2. Set the AF analyzer input to ***AM*** or ***FM Demod*** **(AFAN:INP 'FM')**.
3. You may also use the **TX TEST** screen to access the **AFAnlIn** field but this automatically changes several other settings and is not recommended for IBASIC programming.

# Guideline #3

## Use of Single Quotes and Spaces

The test set's HP-IB syntax requires proper use of single quotes and spaces. The syntax diagrams in the *Programmer's Guide* show where single quotes are needed and where spaces are needed.

**Example:**
OUTPUT 714;"DISP<space> AFAN"
OUTPUT 714;"AFAN: DEMP<space>'Off'"

Notice the use of the single quotes around the Off command. If you leave out the space in the second line, the test set will display "ERROR 103 Invalid Separator".

# Guideline #4

## Improve Measurement Speed

There are several methods to improve measurement speed. Time-to-first-reading after making new settings is usually much slower than the repetitive reading rate once the first reading has been returned. The main contributor to first-reading measurement time is hardware autoranging. See "Speeding Up HP-IB Measurements" in *The Programmer's Guide* under "Advanced Operation" for more details.

You can eliminate autoranging by first establishing the expected signal levels into the test set (both RF and AF). With these signal levels present, the test set will autorange, allowing you to determine the attenuation and gain settings of the RF input attenuator in the **RF ANALYZER** screen, and the various IF and Audio gains shown in the **AF ANALYZER** screen. In your program, select gain control hold (default is auto) and make the settings you recorded earlier. When your program runs, the signal levels into the test set need to remain relatively constant because autoranging has been disabled.

The second improvement you can do is to set all unused measurements to **STATe OFF**. This will primarily speed up reading repetition rate.

Disable RF autotuning to achieve the third speed improvement.

# Guideline #5

## Combine Output Statements

To reduce the number of **OUTPUT** statements used to make the desired settings within one screen, string together multiple settings within one **OUTPUT** statement. To do this, you first make one setting, then back up one level in the syntax hierarchy and make the second setting and so on. For normal forward hierarchical progression of syntax you use the **:** (colon) separator. To back up one level of syntax hierarchy you use the **;** (semicolon) separator. The **;:** takes you back to the highest level of hierarchy.

As documented in the "Using HP-IB" section of the *Programmer's Guide*, you must follow the given hierarchy to make settings for each field. The syntax diagrams in the *Programmer's Guide* show this hierarchy. You can use the **;** to string together the three settings made to the RF generator in the example program listed under Guideline #1 for lines 60, 70 and 80.

```
OUTPUT Addr;"RFG:AMPL -66 DBM;FREQ 500 MHZ;AMPL:STAT ON"
```

In a second example, the following lines can be used to set the AMPS SAT tone frequency and deviation.

```
OUTPUT 714;"ENC:AMPS:SAT:FM 2.35 KHZ"
OUTPUT 714;"ENC:AMPS:SAT:FREQ 5.970 KHZ"
```

These same settings can be made in a single line like this

```
OUTPUT 714;"ENC:AMPS:SAT:FM 2.35 KHZ;FREQ 5.970 KHZ"
```

In another example, if you want to set the RF generator to accept external modulation from the rear panel input and set the amount of modulation all in the same OUTPUT statement, use the following line:

```
OUTPUT 714;"RFG:MOD:EXT:DEST 'FM (/Vpk)';FM 12.5 KHZ;FM:STAT ON"
```

# Guideline #6

## Ensure Fields are Active with State On

When making settings to fields that can be turned OFF with the **STATe ON/OFF** command (**(OFF/ON)** key on the front panel), make sure the **STATe** is **ON** if you want that function to be active. Note that if the **STATe** is **OFF**, just setting a numeric value in the field will not change the **STATe** to **ON**. (This is different than how it works on the front panel.) You have to explicitly send the **STATe ON** command. The previous example involving the RF Generator external modulation input illustrates the proper use of the **STATe ON** command to activate the function after the desired value is set. **STATe ON** is the preset state for most functions affected by the STATe command. External modulation input is one exception.

In another example, if you want to set a new **AMPS** encoder **SAT** tone deviation and the current **STATe** is **OFF**, use the following line: (note the use of **;** to back up one level in the hierarchy so you can make more than one setting in a single line.)

```
OUTPUT 714;"ENC:AMPS:SAT:FM 2.1 KHZ;FM:STAT ON"
```

If you just want to turn on the **SAT** tone without affecting the current setting, use the following line:

```
OUTPUT 714;"ENC:AMPS:SAT:FM:STAT ON"
```

## Guideline #7

### HP-IB Measurements Return in HP-IB Units Only

When querying measurements or settings over HP-IB, the test set always returns numeric values in HP-IB units, regardless of the current display unit. HP-IB units are units of Hz, volts, watts, and seconds, for example. Display units are what you see on the front panel display. This means that when the test set front panel is displaying TX frequency as 835.02 MHz, and you query this measurement over HP-IB, the value returned will be 835020000. Another way of looking at this is that changing display units will not change HP-IB units. (Note that making numeric settings over HP-IB can be done in a wide variety of units.)

## Guideline #8

### HP-IB Syntax Index

When looking for HP-IB syntax in the *Programmer's Guide*, it helps to have an understanding of the organizational structure of the syntax diagrams. The syntax diagrams are divided into four parts as shown below. This list is in the same order as the pages appear in the HP-IB section of the *Programmer's Guide*.

**FUNCTIONALITY**
(listed alphabetically)

AF ANalyzer
AF Generator 1
AF Generator 2 Pre-Modulation Filters
AF Generator 2/ENCoder
CONFigure, I/O Configure
DECoder
OSCilloscope
RF ANalyzer
RF Generator
Radio Interface
Spectrum ANalyzer

## NUMERIC SETTINGS
(Number Setting Syntax: Integer, Real and Multiple Real)

Integer Number Setting Syntax
Real number setting syntax
Multiple Real Number Setting Syntax
- Make Numeric Settings
- Set Display Units
- Set Increment
- HP-IB Units (UNITs)
- STATE ON/OFF

## MEASUREMENT QUERY
(MEASure)

MEASure
- AF ANalyzer (AFRequency)
- OSCilloscope
- RF ANalyzer (RFRequency)
- Spectrum ANalyzer
- DECoder

Trigger
Multiple Number Measurement Syntax
Number Measurement Syntax
- Units
- Average
- HI / LO limits
- Meter
- Reference Set
- STATe (ON/OFF)

## OTHER
(Miscellaneous)

DISPlay (Screen)
PROGram (Subsystem)
TESTs (Environment)
STATus
Save / Recall Registers

# Appendix
# Description of Memory and Mass Storage

## Memory Systems Overview

As shown in Figure A-1 and Table A-1, the test set has both external and on-board memory locations (on-board means inside the test set on the memory board). Programs in any of these memory locations can be accessed and run from an IBASIC program or command. The programmer can select and run programs from any desired memory location using IBASIC commands. The programmer must also decide which mass storage location to use when storing programs and data.

When shipped from the factory (or after initializing RAM with the **SERVICE** screen **RAM Initialize** function), the test set's default mass storage location is INTERNAL, which means the front-panel memory card slot. The front-panel slot accepts small solid-state media called memory cards.

**Card (SRAM)**
SRAM memory cards are the most convenient media for storing user-written programs.
*Hewlett-Packard recommends this as the primary media for program storage and development.*

**Card (OTP)**
As mentioned in Chapter 1, Hewlett-Packard offers many types of IBASIC programs (the HP 11807A/E radio test software and HP 11807B cell site test software) on the ROM version of the memory card.

**Disk Drive**
External hard or flexible disk drives can be used to store user-written programs and data. They must be connected to the test set via the HP-IB interface.
*Hewlett-Packard does not recommend this media for program storage or development because of slow program storage and retrieval times.*

**RAM**
Part of the test set's on-board RAM memory is used to store IBASIC programs, store data from programs, and store **SAVE/RECALL** register information. (**SAVE/RECALL** registers can also be stored in on-board RAM, memory cards, or disk drives).
*Hewlett-Packard does not recommend this media for permanent program storage because of the chance of unintentional program erasure.*
*It is excellent for dedicated applications where the software will not change often.*

**ROM** (factory use only)
Part of the test set's on-board ROM memory contains IBASIC application programs (such as hardware diagnostics and memory card initialization). However, you cannot write to this memory.

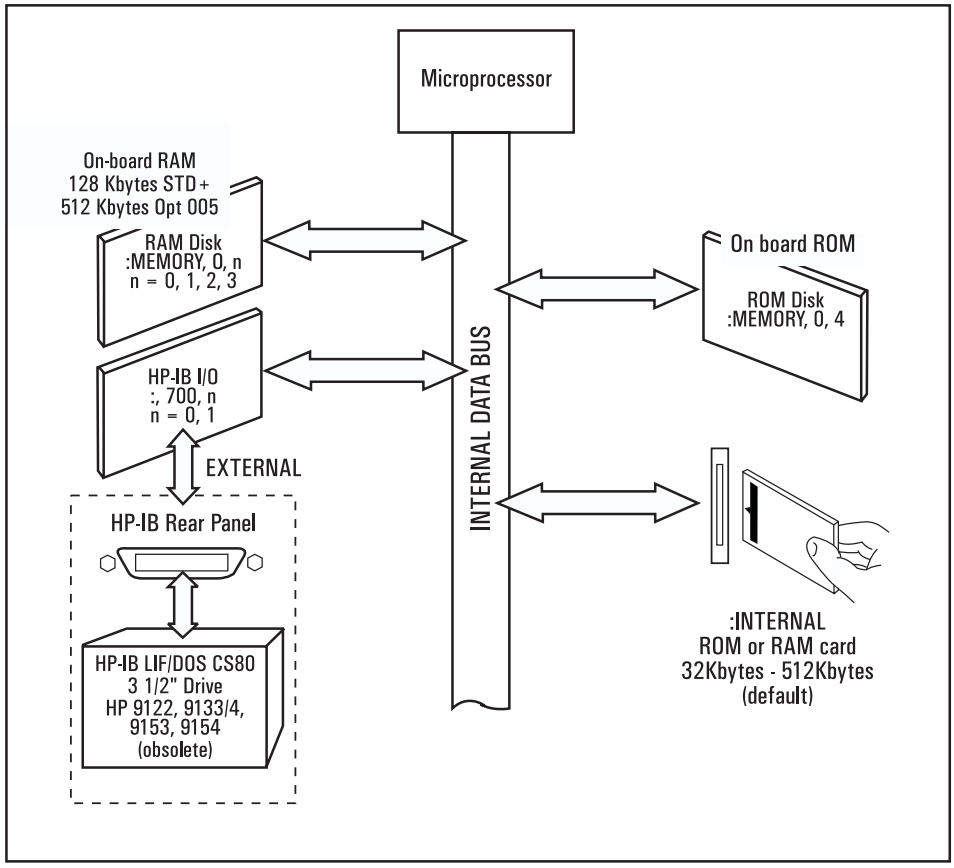


Figure A-1. Test set on-board mass storage locations and size

| Mass Storage Name | Mass Storage Type | Physical Location and Format | Mass Storage Volume Specifier |
|---|---|---|---|
| SRAM Memory Card | Static Random Access Memory | • Plugs into front-panel slot of Test Set<br>• Removable 86x54x2 mm solid-state package | :INTERNAL |

Table A-1. Test set memory overview

### SRAM, Typical Uses and Other Information

- User-written program storage; recommended as the primary method to store and develop programs.
- Accessible to the user from the **TESTS** screen and the **IBASIC command entry** field
- Program or data retained by on-card lithium battery

| Mass Storage Name | Mass Storage Type | Physical Location and Format | Mass Storage Volume Specifier |
|---|---|---|---|
| OTP Memory Card | - OTP = One-Time Programmable Read Only Memory Card<br>- Permanent program storage | - Plugs into front-panel slot of Test Set<br>- Removable 86x54x2 mm solid-state package | :INTERNAL |

**OTP Card, Typical Uses and Other Information**

- Factory-written program storage
- Users cannot erase or write to OTP memory cards
- Accessible to the user from the **TESTS** screen and the **IBASIC Command Entry** field

| Mass Storage Name | Mass Storage Type | Physical Location and Format | Mass Storage Volume Specifier |
|---|---|---|---|
| Disk Drive | - HP-IB Disk Drive<br>- Media is typically 3 1/2" flexible disk or hard disk drive | - Separate unit from Test Set<br>- Connected by HP-IB interface<br>- Flexible disk is removable<br>- Hard disk is usually not removable<br>- HP LIF compatible<br>- DOS compatible | :,7xx,n (where 7xx is HP-IB address, with xx=00 to 30; and n is unit #) |

**Disk Drive, Typical Uses and Other Information**

- User-written program storage; relatively slow program storage and retrieval, not recommended for some applications
- Accessible to the user from the **TESTS** screen and the **IBASIC command entry** field
- Easy storage and retrieval of multiple procedures for different DUT's in manufacturing or service situations
- Good for collection and storage of data in unattended or remote situations

| Mass Storage Name | Mass Storage Type | Physical Location and Format | Mass Storage Volume Specifier |
|---|---|---|---|
| RAM (or RAM Disk) | - Random Access Memory<br>- 4 factory-defined unit numbers (or volumes)[5]<br>- Programs and data protected by Test Set internal battery | - IC's on memory board inside Test Set<br>- Only a portion of Test Set RAM is user-available<br>- Non-removable | :MEMORY,0,n (where n = 0, 1, 2, or 3; the unit number) |

[5] Some test set documentation refers to a section of RAM memory as a "volume" in place of the IBASIC term "unit number." The term "unit" will be used in this product note.

**Typical Uses and Other Information**

- User-written program storage; excellent for dedicated application; not for permanent program storage (easily erased)
- All units accessible to the user from the **IBASIC command entry** field
- Unit 0 accessible to the user from the **TESTS** screen
- Unit 0 can be overwritten by the “RAM_MNG” ROM utility program
- Unit 1 can be overwritten by the “COPY_PL” ROM utility program
- Units 2 and 3 are not overwritten by ROM programs

| Mass Storage Name | Mass Storage Type | Physical Location and Format | Mass Storage Volume Specifier |
|---|---|---|---|
| ROM | - Read Only Memory<br>- Permanent | - IC's on memory board inside Test Set<br>- Non-removable | :MEMORY,0,4 |

**Typical Uses and Other Information**

- Contains factory-supplied utility programs
- Users cannot erase or write to ROM memory
- Accessible to the user from the **TESTS** screen and the **IBASIC command entry** field

## Selecting the Mass Storage Location

The test set’s default mass storage location is INTERNAL, which means the front-panel memory card slot. A common way to select a different mass storage location is to redefine the default location in your program. This is done using the MSI (Mass Storage Is) keyword in IBASIC.

- For example, to change the default location to on-board RAM unit 2, execute the following command:

  MSI **“:MEMORY,0,2”**

- MSI is also a keyboard-executable command, but any changes to the mass storage default are lost when the test set is turned off.

## Listing Memory Contents

By executing the CATalog command, the IBASIC controller will display the contents of memory.

- To display the contents of the default mass storage location (whether SRAM memory card, disk drive, or on-board RAM):

  CAT

- To display the contents of any specific memory location, use CAT followed by the mass storage address. The following example lists the contents of on-board RAM memory unit 0:

  CAT **":MEMORY,0,0"**

## Priority of Memory Use

Memory is allocated in sequence based on the following order:

- The highest precedence is given to space initialized for RAM disk. If you use a RAM disk, remember that RAM is used to contain the current IBASIC executable program, so there must always be sufficient space to contain the whole program or it will not load.
- The second-highest precedence is given to the memory function you first executed since RAM was cleared (typically using the **RAM Initialize** function on the **SERVICE** screen).

## Memory Space Considerations

### Determining the Program Size

1. When programs are **SAVEd** as ASCII files, memory is used at the rate of approximately 25 bytes/line.
2. When programs are **STOREd** as **PROGRAM** files or loaded as executable files, memory is used at the rate of approximately 62.5 bytes/line. This can vary slightly depending on how much COMmon space is dimensioned or how many large arrays are dimensioned in the program.

### Determining Your Test Set's Total On-Board RAM Memory Size

The test set's standard memory size is usually 1 megabyte.

If you have firmware revision 14.XX or higher, press **(TESTS).** Under Select Procedure Location, select "ROM." Under Select Procedure Filename, select "RAM_USAGE." Press k1 (Run Test.) RAM memory usage information will be displayed on the screen. If your firmware revision is lower than 14.XX, there is no direct method to determine total memory space. To indirectly determine the total memory space in your test set, follow steps C & D below:

A. Clearing all on-board **RAM**
   1. Press **(SHIFT)** **(CONFIG)**
   2. Under **To Screen**, select the **SERVICE** screen
   3. Select **RAM Initialize** to clear all RAM

   **CAUTION!** *This will clear all save/recall registers, programs in RAM disk, etc.*

B. Initializing a **RAM** disk to use a known amount of RAM space.
For example:
128 KB = 128 x 1024 bytes = 131,072 bytes
131,072 bytes / 256 bytes/record = 512 records
1. Press **(TESTS)**
2. Under **Location**, select ***ROM***
3. Under **Procedure:**, select ***RAM_MNG*** in HP 8920A and HP 8921B, or ***IB_UTIL*** in HP 8920B and HP 8924C (to use the RAM_MNG utility program)
4. Press **(k1)** (the Run Test softkey)
5. Press **(k1)** (the Yes softkey)
6. Press **(k3)** (the Int RAM softkey)
7. Press the **(5)**,**( 1)**,**( 2)**, and **(ENTER)** keys (to initialize 512 records for unit 0)
8. Press **(k1)** (the Yes softkey)

C. Determine the percentage of total RAM still available (free memory) (Rev. <14.XX)
1. Under **To Screen**, select the **RF GEN** screen
2. Press **(SHIFT)** **(SAVE)**
3. Information at the top of the display will appear like this example:
*(number/name; 65% free memory)*

D. Determine the total memory space in your test set (Rev. <14.XX)
TMKB = Total Memory in KB (kilobytes)
% FM = % Free Memory (65% for this example + 2% for RAM overhead
128 = space initialized in RAM in KB

$$\frac{(65\% + 2\%)\,FM}{100\%\,FM} = \frac{TMKB - 128}{TMKB}$$ ; 0.67 x TMKB = 1.0 TMKB - 1.0 X 128; TM = 387 KB

Thus, total RAM in this test set is 384 KB. Record this total for future use.

## Determining the percentage of total RAM still available (free memory):

1. Under **To Screen**, select the **RF GEN** screen
2. Press **(SHIFT)** **(SAVE)**
3. Information at the top of the display will appear like this:
***(number/name; 99% free memory)***

## Other Memory Space Considerations

When you GET a program from memory, besides pulling the named file into the current IBASIC program memory location, the **GET** command partially compiles the originally saved ASCII file into machine code. (This is a feature of IBASIC). This machine code takes about 2.5 times the space of the saved ASCII file. It is therefore possible to have an ASCII file which appears small enough to fit in the available RAM, but when retrieved with the **GET** command, will not fit into the available RAM due to the compilation expansion.

If many save/recall registers are saved, then there may not be enough space to load a large IBASIC program.

Also, if a large IBASIC program is in executable memory, there may be insufficient memory space to save many save/recall registers.

## Saving and Retrieving Programs

There are two groups of IBASIC commands used to save and retrieve files.

- **SAVE** and **GET** are used for ASCII files. ASCII files occupy approximately 2.5 times less memory space than PROGRAM files. The drawback is that they transfer into the test set more slowly. All HP 11807A and HP 11807B test programs are saved as ASCII files to conserve space on memory cards.
- **STORE** and **LOAD** are used for **PROGRAM** files. **PROGRAM** files are about 2.5 times larger than the equivalent ASCII files because **PROGRAM** files have been partially compiled. Compiled code occupies more space because IBASIC is a high level language in which a single command (represented in ASCII) actually instructs the computer to execute many machine steps.

The following examples show the command format for the IBASIC keywords **SAVE**, **GET**, **STORE**, and **LOAD**.

1. To save an IBASIC program as an ASCII file to the default mass storage location (where My_file is the unique name of the file):
   SAVE **"My_file"**
2. To retrieve a PROGRAM file (from RAM disk unit 3 in the following example):
   LOAD **"My_file:MEMORY,0,3"**

# Memory Cards

This media was chosen for test set program and data storage because of its small physical size and ruggedness. The SRAM card is recommended for program storage because it is convenient, transportable, and requires no peripheral hardware.

## The Memory Card Location

As shown in Figure A-1, the test set's default mass storage location is INTERNAL, which means the front-panel memory card slot.

- Default mass storage
  If the default mass storage location has changed, you can designate the memory card to be the default memory by using the mass storage is (MSI) command:
  1. MSI **":INTERNAL"**
- Catalog
  To catalog the card if it is not the default mass storage:
  1. CAT **":INTERNAL"**
- Each memory card is treated as one mass storage unit in IBASIC, so there is no mass storage unit number or volume number.

## SRAM Cards

SRAM cards provide removable read/write memory for programs, similar to a flexible disk. Data and files can be stored, re-stored, read, or erased as needed.

### SRAM Card HP Model Numbers:

HP 85700A — 32 KB
HP 85702A — 128 KB
HP 85704A — 256 KB
HP 85705A — 512 KB

### Battery

SRAM memory cards require a battery to maintain stored information. The lithium battery is HP part number 1420-0383 (or non-HP type CR2016). A fresh battery in an SRAM card typically retains data for over 1 year at 25 $^{O}$C. To ensure data retention, replace the battery annually. (See the Test Set's *User's Guide* for more information about replacing the SRAM card battery).

When a SRAM card is plugged into an operating test set, the card draws power from the test set. This allows you to change the battery while the card is powered by the test set. **NOTE:** Don't remove the battery unless the card is in a powered-up test set.

### Card Insertion and Removal

Memory cards may be inserted and removed with the test set powered on or off.

### Initialization

When you initialize an SRAM card, you do not specify the number of records. The initialization process always initializes the whole card. Initialization destroys any existing files on the SRAM card. There are two initialization methods for cards:

1. Use the IBASIC Command — INITIALIZE **":INTERNAL"**
2. Use the **RAM_MNG** program located in on-board ROM. (Note: Do not use this method if you have a program you want to keep in test set memory. Loading the RAM_MNG program will erase your program).

### Write-Protect Switch

Slide the switch on the end of the card to the "SAFE" position to prevent writing to the card (as shown on the label on one side of the HP SRAM cards).

### Saving and Retrieving Programs with SRAM

The SRAM card must be initialized before a **SAVE** or **STORE**.

1. To **SAVE** or **GET** an ASCII file:
   SAVE **"My_file:INTERNAL"**
   GET **"My_file:INTERNAL"**
2. To **STORE** or **LOAD** a **PROGRAM** file:
   STORE **"My_file:INTERNAL"**
   LOAD **"My_file:INTERNAL"**

### Determining Remaining SRAM Space

1. CATalog the card and look at the number of records occupied by the programs listed.
2. Example: a 256 KB SRAM card is used to save one ASCII program and two BDAT (binary data) files. (Also, there are typically 14 records used for the on-card file directory which is not shown when you CATalog the card). The number of bytes used is:

   **Hidden Directory** 14 records
   **ASCII Program** 548 records
   **BDAT File** 21 records
   **BDAT File** 16 records
   (14+548+21+16) * 256 bytes/record = 153 KB

## OTP (ROM) Cards

All HP 11807A and HP 11807B software is shipped on the OTP version of this media (One-Time Programmable ROM).

- OTP cards provide removable read-only storage for HP-supplied programs.
- OTP cards do not require a battery to maintain stored information.

- Memory cards may be inserted and removed with the test set powered on or off.
- OTP cards cannot be programmed by a test set. A special memory card programmer (not available from HP) is required to save files to OTP cards.
- **Catalog**
  To catalog the card if it is not the default mass storage:

  CAT **":INTERNAL"**

# On-Board RAM Memory

On-board RAM serves three primary functions (see Table A-1 for more details of the on-board RAM structure):

## 1. Storage for SAVE/RECALL Registers

Although save/recall registers occupy on-board RAM, the amount of RAM used is usually very little. If you need to maximize the RAM space available for programs and data, delete the save/recall registers or save them to memory cards or disk (see the Test Set's *User's Guide*).

## 2. Storage for Executable IBASIC Programs

To run a program, **GET** or **LOAD** the program from the mass storage memory location. This action will place the program into RAM. There must be sufficient RAM available to hold the program or it will not load. (See "Memory Space Considerations" earlier in this chapter for a discussion of the memory space occupied by programs).

If you encounter an "insufficient memory" error when getting or loading a program, you can:
a. Reduce the size of the program
b. Release space that is being held for one of the other two RAM functions
c. Add more RAM memory to the test set (if possible)

## 3. Storage for IBASIC Programs or Data (RAM Disk)

RAM disk is a term used to designate the test set's on-board RAM used to store IBASIC programs and data.

You can designate RAM disk to be the default memory by using the MSI (Mass Storage Is) command.

MSI **":MEMORY,0,n"**
(where n = 0, 1, 2, or 3; the unit number)

The RAM disk is divided into four IBASIC unit numbers (often called volumes in other test set documentation). The complete RAM disk space is available to each unit if necessary.

All RAM disk units are accessible to the user from the IBASIC command entry field.

Unit 0 is accessible to the user from the **TESTS** screen. Unit 0 can be overwritten by the "RAM_MNG" ROM utility program.

Unit 1 can be overwritten by the **"COPY_PL"** ROM utility program.
Units 2 and 3 are not overwritten by ROM programs.

On-board RAM is different than other test set mass storage media. When INITIALIZEd for use as a RAM disk, that portion is no longer available to the test set for other RAM functions (such as containing an executable program or save/recall registers). The programmer should INITIALIZE only the portion needed to leave enough to allow other functions to operate properly.

### Initialization

Before you save programs on RAM disk, you need to INITIALIZE enough space for your program. Initialization destroys any existing files on the RAM disk unit. A typical INITIALIZE command looks like this:

INITIALIZE **":MEMORY,0,0",50**
(where 50 is the number of records reserved for RAM disk)

There are 256 bytes/record. For a test set with 640 KB total memory available, you could theoretically initialize a RAM disk for a maximum of 640 kilobytes / 256 bytes/record = 2560 records. (However, this will leave no space to GET and run the program).

### Clearing RAM Disks

There are two ways to clear RAM disks.

1. To clear all RAM (including save/recall registers, any loaded program, and all RAM disk space), use the **RAM Initialize** field located in the **SERVICE** screen.

   a. To access the **SERVICE** screen, press the **(SHIFT)** and then the **(CONFIG)** key.
   b. From the **To Screen** menu, select **SERVICE.**
   c. Select the **RAM Initialize** field to clear all on-board RAM.

2. To clear only a specific RAM disk unit, initialize only that RAM disk unit to zero records (example is for unit 0):

   INITIALIZE **":MEMORY,0,0",0**

### Accessing "Unavailable" RAM Disk Space

Unlike most disk drives, RAM disk space is not "packed" after you delete a program. Thus, if you try to load a slightly larger program into the same unit, the previous space will not be reused, but new space at the end of the previous program will be used. It is possible to delete programs from RAM disk yet still not have enough space to save a program that is smaller than the sum of the programs deleted. In this case, you need to re-initialize the RAM disk unit which will erase all the files on the unit.

## Disk Drives

The test set supports HP-IB disk drives that operate under the Hewlett-Packard Logical Interchange File (LIF) format with CS-80 command structure. In the past, Hewlett-Packard has manufactured both flexible and hard disk drives of this type. Here is a partial list of the previously-manufactured HP model numbers which will operate with the test set:

- HP 9122A, HP 9122C, and HP 9122D,S single and dual 3 1/2" flexible disk drives
- HP 9133H,L and HP 9153A,B,C single 3 1/2" flexible disk plus fixed hard disk drive
- HP 9134H,L, HP 9153C, and HP 9154A,B fixed hard disk drives

(The HP 9121 series of disk drives are not compatible with the test set). These disk drives or compatible ones available from other manufacturers can be

connected directly to the HP-IB port on the back of the test set and used almost identically to the other mass storage media described in this chapter.[6]

Because the disk drive is external to the test set and connected via HP-IB, the test set must be configured as a controller on the HP-IB bus before accessing the disk drive. To do this, make the following settings:

1. Access the **I/O CONFIGURE** screen
2. On the **Mode** field, select ***Control***

You can designate the disk drive as default memory by using the **MSI** (Mass Storage Is) command. With an external disk drive, you can leave out the volume number so the command can be shortened to:

1. MSI ":,<device selector>,<unit number>"
2. MSI **":,7xx,m"** (where 7xx = HP-IB address and m = unit number)
3. MSI **":,700,0"**

The HP-IB address for a disk drive is typically 700, but can be selected from 700 to 730.

In most cases, the unit number is either 0 or 1 to indicate a left-hand disk drive or a right-hand disk drive of a two-drive disk system.

Memory access and storage operations typically require more time with a disk drive attached to the test set than with RAM disk or memory cards.

Initialization is just like the card media. All existing files on the media will be destroyed. (For more information, see "INITIALIZE" in the *HP Instrument BASIC Language Reference*. Also, review the documentation with the disk drive to assure compatible initialization is performed since some drives can be configured for CS-80, DOS, and other file format options). For example, you could use this command to initialize the left-hand disk drive at HP-IB address 700 of a dual-floppy drive:

```
INITIALIZE ":,700,0"
```

To save a program to a disk drive when it is the default memory location, execute the following command:

```
SAVE "My_file"
```

To save to a disk drive when it is not the default memory, execute the following command:

```
SAVE "My_file:,700,0"
```

**NOTE:**
On the bottom of the test set's **TEST (External Devices)** screen is a field titled **External Disk Specification**. This field is used to set the external disk location in conjunction with the TESTS environment discussed in the Test Set's *User's Guide* in the "Tests" section.

[6] One currently-manufactured disk drive which has been tested to operate with the test set: Model 7312 Option 400 – 3 1/2" Flexible Disk plus 120 MB Winchester Hard Disk Drive (CS-80 and LIF compatible). Contact: Bering Technology, 1357 Dell Ave., Campbell, CA 95008 USA; (408) 379-6900.

For more information about the HP 8920 Test Set Family visit our web site at:

**http://www.hp.com/hp8920support**

Available literature may include: product overviews, technical specifications, product notes, configuration guides, application notes, and more.

**Warranty** This test set is backed by Hewlett-Packard with a one-year warranty and world class support services. With HP sales and service centers worldwide, you can be assured of getting the global support that you need.

HEWLETT® PACKARD

Expanding Possibilities

For more information about Hewlett-Packard test and measurement products, applications, services, and a current sales office listing, visit our web site at:

**http://www.hp.com/go/tmdir**

You can also contact one of the following centers and ask for a test and measurement sales representative.

**United States:**
Hewlett-Packard Company
Test and Measurement Call Center
P.O. Box 4026
Englewood, CO 80155-4026
(tel) 1 800 452 4844

**Canada:**
Hewlett-Packard Canada Ltd.
5150 Spectrum Way
Mississauga, Ontario
L4W 5G1
(tel) 1 877 894 4414

**Europe:**
Hewlett-Packard Company
European Marketing Organisation
P.O. Box 999
1180 AZ Amstelveen
The Netherlands
(tel) (31 20) 547 9999

**Japan:**
Hewlett-Packard Japan Ltd.
Measurement Assistance Center
9-1, Takakura-Cho, Hachioji-Shi,
Tokyo 192-8510, Japan
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

**Latin America:**
Hewlett-Packard
Latin American Region Headquarters
5200 Blue Lagoon Drive, 9th Floor
Miami, Florida 33126 U.S.A.
(tel) (305) 267-4245
(tel) (305) 267-4220
(fax) (305) 267-4288

**Australia/New Zealand:**
Hewlett-Packard Australia Ltd.
31-41 Joseph Street
Blackburn, Victoria 3130
Australia
(tel) 1 800 629 485 (Australia)
(tel) 0800 738 378 (New Zealand)
(fax) (61 3) 9210 5489

**Asia Pacific:**
Hewlett-Packard Asia Pacific Ltd.
17-21/F Shell Tower, Times Square,
1 Matheson Street, Causeway Bay,
Hong Kong, SAR
(tel) (852) 2599 7777
(fax) (852) 2506 9285


Technical data subject to change
Printed in U.S.A. 4/99
5963-0046E