89400-Series
Vector Signal Analyzers

Using Agilent Instrument BASIC

For Instruments with Firmware Revision
A.02.00

Agilent Technologies

Part Number 89441-90013

Printed in U.S.A.

Print Date: April 1994

Notice

Hewlett-Packard to Agilent Technologies Transition

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

Contacting Agilent Sales and Service Offices

The sales and service contact information in this manual may be out of date. The latest service and contact information for your location can be found on the Web at:

`http://www.agilent.com/find/assist`

If you do not have access to the Internet, contact your field engineer or the nearest sales and service office listed below. In any correspondence or telephone conversation, refer to your instrument by its model number and full serial number.

United States
(tel) 1 800 452 4844
(fax) 1 800 829 4433

Canada
(tel) +1 877 894 4414
(fax) +1 888 900 8921

Europe
(tel) (31 20) 547 2323
(fax) (31 20) 547 2390

Latin America
(tel) (305) 269 7500
(fax) (305) 269 7599

Japan
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

Australia
(tel) 1 800 629 485
(fax) (61 3) 9210 5947

New Zealand
(tel) 0 800 738 378
(fax) 64 4 495 8950

Asia Pacific
(tel) (852) 3197 7777
(fax) (852) 2506 9284

Agilent Technologies

Printed in USA July 2004

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MANUAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

WARRANTY

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Use of this manual and flexible disc(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs can be made for security and back-up purposes only.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013.

HEWLETT-PACKARD COMPANY

3000 Hanover St.

Palo Alto, CA 94303

Table of Contents

Chapter 4: Saving and Recalling Programs

Chapter 5: Developing Programs

Chapter 6: Debugging Programs

Chapter 7: Graphics and Display Techniques

Chapter 8: Interfacing with the HP-IB

Chapter 9: Example Programs

Chapter 10: Instrument-Specific HP Instrument BASIC Features

Chapter 1

Introduction

1

Introduction

Welcome to HP Instrument BASIC

This manual will help you learn about using your HP Instrument BASIC software on the HP 89400-series Vector Signal Analyzers. It shows you how to use the programming, editing and debugging features of HP Instrument BASIC. It also describes how to save and recall programs and how the HP 89400-series Analyzers implement HP Instrument BASIC features.

An additional aid is online help which provides key-specific information on HP Instrument BASIC features. This help is accessed in the same manner as it is for other features of the HP 89400-series analyzers. Press the [**Help**] hardkey followed by the desired hardkey or softkey or use the index.

HP Instrument BASIC Applications

HP Instrument BASIC can be used for a wide range of applications, from simple recording and playback of measurement sequences, to remote control of other instruments.

HP Instrument BASIC is a complete system controller residing inside your analyzer. It communicates with your analyzer via HP-IB commands and can also communicate with other instruments, computers and peripherals over the HP-IB interface.

Using HP Instrument BASIC

You need not be proficient in a programming language to successfully use HP Instrument BASIC. With keystroke recording, HP Instrument BASIC automatically builds an executable program by capturing measurement sequences as they are performed. With little or no editing of this generated code, you can put your program to work immediately controlling and automating your HP 89400-series analyzers.

HP Instrument BASIC's programming interface includes an editor, a program debugger, and a set of programming utilities. The utilities allow you to set memory size as well as renumber, secure or delete your program. The remaining softkeys allow you to run or continue a program, print a listing or configure the display.

The HP Instrument BASIC command set is similar to the command set of HP Series 200/300 BASIC. HP Instrument BASIC programs can run on any HP BASIC workstation with few, if any, changes. Refer to chapter 8, "Interfacing with the HP-IB," for information on interfacing the HP Series 200/300 BASIC and HP Instrument BASIC environments. Porting information is located in the "HP Instrument BASIC Language Reference" section of the *HP Instrument BASIC Users Handbook.*

How to Use This Manual

Read chapters 1 through 4 to learn how to record, run, save and recall programs with a minimum of editing and programming. This information is generally adequate for those who only need HP Instrument BASIC to record their measurement tasks.

Read chapter 5, "Developing Programs," and chapter 6, "Debugging Programs" to learn how to edit programs with the front panel or with a keyboard.

Read chapter 7, "Display and Graphics Techniques," to understand how HP Instrument BASIC's graphics features apply to the HP 89400-series analyzers.

Read chapter 8, "Interfacing with the HP-IB," to understand how the HP Instrument BASIC controller in the HP 89400-series analyzers interfaces with external devices (such as plotters) and external controllers (such as HP Series 200/300 controllers).

Refer to chapter 9 for example programs written in HP Instrument BASIC to run on the HP 89400-series analyzers.

Chapter 10 couples this manual with the *HP Instrument BASIC Users Handbook*. The handbook serves users of HP Instrument BASIC on all instrument platforms. It contains three sections:

- "HP Instrument BASIC Programming Techniques"
- "HP Instrument BASIC Interfacing Techniques"
- "HP Instrument BASIC Language Reference"

Chapter 10 also clarifies which parts of the handbook do not apply to the HP 89400-series analyzers.

Typographical Conventions

The following conventions are used in this manual when referring to various parts of the HP Instrument BASIC and HP 89400-series analyzer's operating environments:

[**Hardkey**]	Brackets [ ] surrounding a bold-faced name indicate the name of a hardkey on the front panel of the HP 89400-series analyzers.
[softkey]	Brackets [ ] surrounding a name indicate the name of a softkey.
[softkey **on** off]	Bolded selection in a softkey indicates the state *after* the softkey is pressed.
[**Hardkey**] [softkey] [softkey]	A series of hardkeys and softkeys represents the path to a given softkey or menu.
[*Key*]	Brackets [ ] surrounding an italic typeface indicate the name of a key on the keyboard which can be used to edit HP Instrument BASIC programs.
Italic	Italic typeface is used when referring to the name of a different manual. It is also used to emphasize a particular word or phrase.
<element>	Angle brackets are used to signify a syntax element in a statement.

Other Sources of Information

- *Getting Started Guide*
- *Operator's Guide*
- Online Help
- *HP Instrument BASIC Users Handbook*:
  - HP Instrument BASIC Programming Techniques
  - HP Instrument BASIC Interfacing Techniques
  - HP Instrument BASIC Language Reference

Need Assistance?

If you need assistance, contact your nearest Hewlett-Packard Sales and Service Office listed in the HP Catalog, or contact your nearest regional office listed at the back of this guide. If you are contacting Hewlett-Packard about a problem with your HP 89400-series Vector Signal Analyzer, please provide the following information:

- Model number:
- Serial number and firmware version:
  (To locate the analyzer's serial number and firmware version, press [**System Utility**] [more] [serial number].)
- Options:
- Date the problem was first encountered:
- Circumstances in which the problem was encountered:
- Can you reproduce the problem?
- What effect does this problem have on you?

Chapter 2

Recording Programs

2

Recording Programs

Keystroke Recording

Of all the available methods of creating HP Instrument BASIC programs, the easiest is keystroke recording. It requires only a couple of steps to set up and run a program. It can be accomplished with very little knowledge of programming.

What is Keystroke Recording?

Keystroke recording is a way to automatically create HP Instrument BASIC measurement sequence programs.

To enable recording, press:

[**BASIC**]

[enable recording]

Press the normal key sequences of a measurement on the analyzer. To stop recording, press the [**BASIC**] hardkey. To run the program, press the [run] softkey in the [**BASIC**] menu.

HP Instrument BASIC programs communicate with the analyzer over an internal bus. HP Instrument BASIC uses the same set of commands that external controllers use for remote operation of the instrument. Keystroke recording works by finding the bus command, called an HP-IB command, that fits each operation you perform from the front panel. It builds a program line that duplicates that operation when executed.

All program lines built by keystroke recording are entered into the analyzer's program memory. If the memory location does not contain any code, a complete executable program is inserted. If program statements exist in the memory location when recording is turned on, the recorded statements are inserted into the existing code. Chapter 5, "Developing Programs," describes how to record into existing programs.

HP Instrument BASIC Programs and the HP-IB Buffer

Recorded programs work by sending HP-IB commands to the analyzer. The analyzer queues the HP-IB commands into its input buffer. An HP Instrument BASIC program generally outputs the commands much faster than the analyzer can execute them. The program often completes before the analyzer finishes executing the commands in the input buffer. The analyzer continues to process these commands until the buffer is empty.

This can be a problem if you are not aware of the possible delay. For example, it may not be obvious that the program has completed, since the analyzer is still functioning. This could cause confusion if you try to pause and continue a program that has actually finished.

You can clear the input buffer by inserting the statement "CLEAR 8" at the beginning of your program. Refer to chapter 5 for more information on developing and editing programs.

What's in a Recorded Program

Any program created with keystroke recording is composed of three fundamental HP Instrument BASIC statements:

- ASSIGN
- OUTPUT
- END

The following simple program demonstrates these statements:

```
1 ASSIGN @Hp89410a TO 800
2 OUTPUT @Hp89410a;"FREQ:SPAN:FULL"
10 END
```

There is only one ASSIGN statement at the beginning of a program and only one END statement at the end, but in a typical program there are many OUTPUT statements. The OUTPUT statement does the actual work of controlling the HP 89400-series analyzers.

The OUTPUT Statement

The HP Instrument BASIC statement

```
OUTPUT  <destination>; <data>
```

essentially tells the internal computer to send some information (*data*) to a device at a specific address (*destination*). The destination can be a device selector (a number), or a name representing a number, called a path name. The data can take several forms but in recorded HP Instrument BASIC programs it is a string containing instructions to the analyzer.

The following command represents a typical OUTPUT statement generated from a recording session:

```
OUTPUT @Hp89410a;"FREQ:SPAN:FULL"
```

The OUTPUT command is followed by a name representing the device selector (@Hp89410a), followed by a semicolon, followed by the data. The data is in quotes ("FREQ:SPAN:FULL") and contains an instruction to the analyzer.

The ASSIGN Statement

The destination in an OUTPUT statement specifies the address of the device. In recorded programs this address is represented by the I/O path name "@Hp89410a." The following line appears in all recorded programs before any OUTPUT statements:

```
ASSIGN @Hp89410a TO 800
```

The ASSIGN statement substitutes an I/O path name (a variable name preceded by the @ symbol) for a device selector number. After the above ASSIGN statement, the program line:

```
OUTPUT @Hp89410a;"FREQ:SPAN:FULL"
```

is equivalent to:

```
OUTPUT 800;"FREQ:SPAN:FULL"
```

The device selector 800 specifies the *host instrument* as the destination of any data sent by the OUTPUT command. The program communicates with the analyzer via select code 8, the internal HP-IB interface. This select code is used solely for communication between HP Instrument BASIC programs and the analyzer. The analyzer responds to any address on the internal interface from 800 to 899. (800 is typically used.)

HP-IB Commands

The data sent to the analyzer by the OUTPUT command is called an HP-IB command. Many of the HP-IB commands for the HP 89400-series analyzers conform to SCPI—the Standard Commands for Programmable Instruments The HP-IB command is found in quotes following the device selector path name and semicolon:

```
2 OUTPUT @Hp89410a;"FREQ:SPAN:FULL"
```

The HP-IB commands used in HP Instrument BASIC are the same ones used to remotely control the analyzer from an external computer. External computers communicate with the analyzer over the *external bus* while HP Instrument BASIC programs communicate with it over the *internal bus*. In our example, "FREQ:SPAN:FULL" tells the analyzer to set the start frequency to its minimum value and the stop frequency to its maximum value.

Note

Many, but not all of the HP 89400-series analyzers HP-IB commands conform to SCPI. Refer to the *HP-IB Programming* documentation shipped with your analyzer for a complete description of HP-IB commands, including compliance to SCPI.

For more information on interfacing HP Instrument BASIC with a bus, see chapter 8, "Interfacing with the HP-IB."

How Recording Works

To fully understand HP Instrument BASIC recording, it is important to understand the relationship between the analyzer's front panel operation and the program that is generated to emulate that operation.

Note

HP-IB commands entered in a program during a recording session do not necessarily have a one-to-one correlation with the actual keys that are pressed during that session.

It is important to know that HP-IB commands correspond to an operation—not to the front panel's hardkeys and softkeys. It may take several keystrokes to perform an operation. Keystroke recording generates the appropriate HP-IB command *after* you have pressed a valid sequence of keys.

In other words, the functional operation of the analyzer is recorded, not the exact series of keystrokes.

For example, recording the key sequence:

[**Frequency**]
[full span]

requires two keystrokes but produces only one command. The command, "FREQ:SPAN:FULL," is generated after the sequence is completed. Keystroke recording automatically formats this operation into the statement:

```
OUTPUT @Hp89410a;"FREQ:SPAN:FULL"
```

and inserts it into the program.

If you accidentally press the wrong key in a sequence, it may not appear in the recorded program. It also means that you cannot exactly mimic keystrokes to leave the analyzer in a specific front-panel state. The analyzer's state appears only as a natural consequence of a *completed operation*.

For example, in the above example, pressing [**Frequency**] in a recording session has the effect of bringing up the [**Frequency**] menu. However, it does not, by itself, generate a line of code. You could not, therefore, set the analyzer to display the [**Frequency**] menu.

Operations That Are Not Recorded

Although in most situations keystroke recording works automatically, there are some operations that are not captured or are only partially captured using this method. These operations fall into one of the following areas:

- Front panel operations with no corresponding HP-IB command such as help text operations, HP-IB controller status, RPG (knob) operations and transitional key sequences.
- Operations requiring additional programming steps, such as passing control to the analyzer for plotting or special handling of measurement operations which arm a trigger.
- HP-IB operations with no equivalent front panel operations such as HP-IB query commands.

Front Panel Operations Without HP-IB Commands

There are some front panel operations which have no corresponding HP-IB commands.

The help text available through the [**Help**] hardkey has no corresponding HP-IB command. Help cannot be accessed from the HP-IB. Therefore, the keystroke is not recorded.

You cannot remotely change the analyzer's controller status. This has two significant consequences:

- You cannot remotely change the state of the HP-IB interface. For example, you cannot change the analyzer from Addressable Only to the System Controller.
- You cannot remotely abort an I/O operation when the analyzer has active control of the HP-IB interface. I/O operations are printing, plotting, and using an external disk drive.

Any front-panel key sequences that perform these operations do not generate an HP-IB command. They are not keystroke recorded.

You may enter numeric values by using the the numeric keys or by incrementing or decrementing the value by turning the knob.

During a measurement sequence it may take several key presses to reach an operation that generates a command. The transitional sequences between actual instrument events are not recorded.

Any default settings you do not select while recording are not recorded.

Note

It is important to remember instrument settings not specifically selected or changed *are not recorded.*

Since default states are not recorded, you must actively select them to generate a program statement. An alternate method is to make sure the analyzer is in the *same exact state* when the program runs as it was when the program was recorded. This is discussed later in this chapter in "Avoiding Recording Errors."

HP Instrument BASIC Operations

Softkeys under the [**BASIC**] key cannot be recorded because pressing this key turns off keystroke recording. In addition, [**Save/Recall**] operations that refer to HP Instrument BASIC programs are not recorded. You can, however, record all the other save and recall operations which do not refer to HP Instrument BASIC programs.

Although operations in the [**BASIC**] menus cannot be recorded, many do have corresponding HP-IB commands that allow an external controller to control and communicate with internal HP Instrument BASIC programs. See the example program, TWO_CTRL, in chapter 9 and the HP-IB Programming documentation shipped with your analyzer for more information.

Operations Requiring Additional Programming

Some operations that work well when performed from the front panel have special circumstances that need additional attention when used in an HP Instrument BASIC program. These operations are synchronization and active control.

Synchronization

You must always anticipate timing and synchronization when one event must complete before another can occur. One example of this is when you need to detect a state in the instrument before issuing the next command. For example, you may want your program to single sweep for several measurements, but only after each measurement has successfully completed. You can record the command to set the analyzer to single sweep mode, and the command to single sweep, by pressing key sequences. However, to detect when the analyzer has completed a measurement, you must edit the program and include a routine that waits for a status register to indicate the event has occurred. (For an example of this kind of program, see the SNGL_SWP program in chapter 9.)

Active Control

The [start plot/print] operation, as well as any external disk drive operation, requires the analyzer to be the active controller on the external bus. This means that the analyzer must be set as the System Controller before the program runs; or, an external controller must pass active control to the analyzer. The instrument's active control of the external interface is automatically passed to the HP Instrument BASIC program when it begins running. Active control must be passed back to the analyzer before it can execute the print/plot or external disk operations.

Although you can keystroke record operations involving an external disk drive or [start plot/print], you cannot successfully run the generated program. You need to add program lines to first pass active control to the analyzer and then wait for the active control of the bus to be passed back to the HP Instrument BASIC program. See "Passing and Regaining Control" in chapter 8 for an example of passing control to the analyzer.

Operations Not Available From The Front Panel

Operations such as querying the analyzer's status, transferring data over the external bus, and setting and clearing status registers are not available from the analyzer's front panel. These operations cannot be keystroke recorded. They are useful for HP-IB programming using HP Instrument BASIC. See the HP-IB Programming documentation shipped with your analyzer for a description of these types of operations.

Avoiding Recording Errors

This section describes ways to minimize the mistakes you can make when using keystroke recording.

Use Preset

You should preset the analyzer before recording a measurement sequence and again before running the recorded program. This sets the instrument to its default state. It avoids the risk of creating a program that depends on instrument settings that were present at the time of the keystroke recording but may be different when the program runs.

To include the command that presets the analyzer, press [**Preset**] immediately after enabling keystroke recording. This inserts the following line before all the other OUTPUT statements in your program:

```
OUTPUT @Hp89410a;"SYST:PRES"
```

This sets the analyzer to its default state.

Selecting Specific Parameters

You may not want to preset the analyzer before a recorded program runs because you are recording a section of a larger measurement sequence. In this case, be sure to activate every instrument setting you need in your automated sequence. For example, if you want the format to be SINGLE, press [**Display**] and then [single grid], even though SINGLE is already the default setting. This generates a program line which specifically sets the format to SINGLE.

In some cases you may have to select another setting first and then re-select the original setting in order to generate the correct program line. For example, you want to generate a program line to select a linear magnitude data format. While you are recording you discover that linear magnitude is already selected. Press the [magnitude log linear] toggle switch twice — once to select log and then again to select linear. You can delete any unwanted program lines generated by this procedure by using the editor. Editing is covered in detail by Chapter 5.

Use SCPI Echo

You can review the HP-IB commands before you actually record them. While this is not essential, it can be very useful when you are in doubt as to what a particular key sequence will record, or precisely when a key sequence corresponding to an HP-IB command is completed.

SCPI Echo is a facility that allows you to view HP-IB commands corresponding to any operation executed from the front panel. A command appears in the upper left corner of the display (the third line) as you complete any key sequence that has a matching HP-IB command. See figure 2-1. This command is the same as those generated in your recorded program during a recording session.

At power-up, the default status of SCPI Echo is off. To turn on SCPI Echo, press:

[**Local/setup**]
[SCPI cmd echo **on** off]

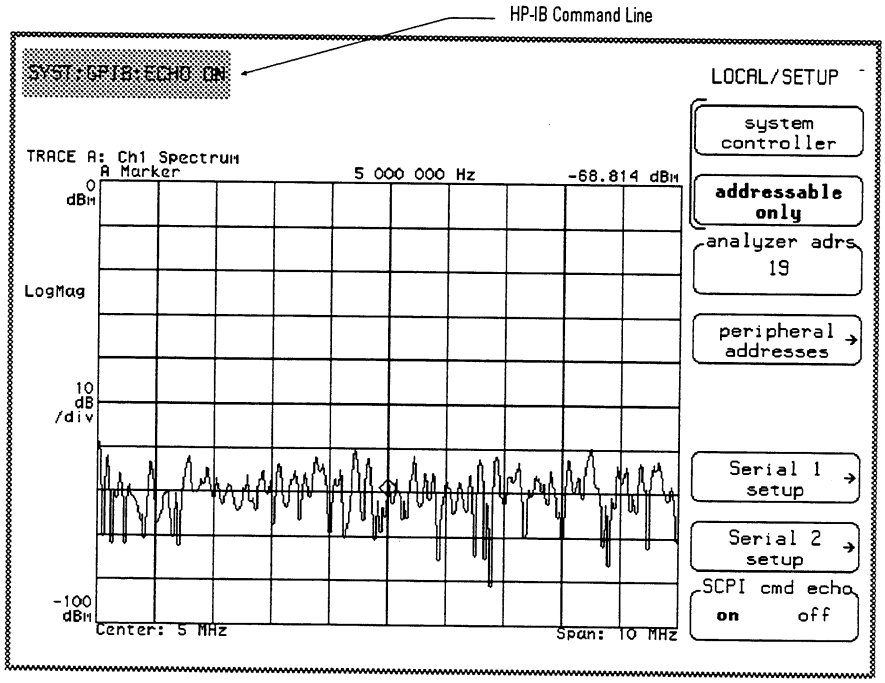

Figure 2-1. SCPI Echo

Chapter 3

Controlling Programs

3

Controlling Programs

You can start, pause and stop an HP Instrument BASIC program from the front panel of the HP 89400-series analyzers using various hardkeys and softkeys. This chapter describes how to control an HP Instrument BASIC program.

HP-IB commands can control HP Instrument BASIC programs over the external bus. You can use an external controller to run HP Instrument BASIC programs. For information on running, pausing and stopping programs from an external controller see chapter 8, "Interfacing with the HP-IB."

Running and Continuing a Program

The [**BASIC**] menu displays a [run] softkey. See figure 3-1. This key runs the program which is currently stored in program memory.

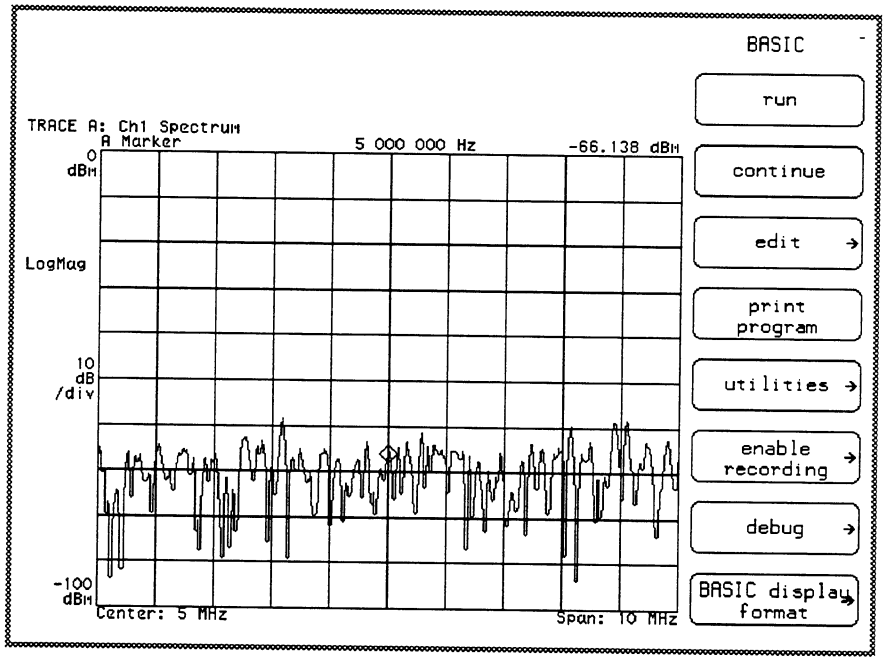

Figure 3-1. The [BASIC] Menu

To run the program, press:

[**BASIC**]
[run]

There is also a [run] key in the [**BASIC**] [debug] menu. This key allows you to run the current program during program debugging (see chapter 6, "Debugging Programs"). Both of these softkeys perform the same HP Instrument BASIC RUN command.

The RUN command is executed in two phases: prerun initialization and program execution.

The prerun phase consists of:

- Reserving memory space for variables specified in COM statements (both labeled and blank).
- Reserving memory space for variables specified by DIM, REAL, INTEGER, or implied in the main program segment. Numeric variables are initialized to 0; string variables are initialized to the null string.
- Checking for syntax errors that result from multiple program statements. Incorrect array references and mismatched parameters or COM lists are examples of these types of syntax errors.

After the prerun phase successfully completes, the program continues executing until one of the following events occurs:

- The program encounters an END or STOP statement.
- The program encounters a PAUSE statement.
- You press the [**Local/Setup**] hardkey to stop the program.
- You press the [**BASIC**] hardkey to pause the program.
- You press the [**Preset**] hardkey to stop the program.
- The program encounters a run-time error.

Pausing a Program

You can pause a program by pressing the [**BASIC**] hardkey. Another way to pause a program is to insert a PAUSE statement into your program. (Refer to chapter 5, "Developing Programs," to learn how to insert statements into your recorded program.) In either case, the analyzer temporarily stops executing the program.

To continue a paused program, press the [continue] softkey in the [**BASIC**] menu or in the [debug] menu. Continuing a paused program resumes the operation from where it was paused in the program. The program retains the values for any variables.

Pausing a program does not close any files that have been opened by the program. You will not be able to perform any of the following disk operations from the front panel after pausing a program that has left a file open on that medium:

- RENAME FILE
- DELETE FILE
- DELETE ALL FILES
- COPY FILE
- COPY ALL FILES
- FORMAT DISK

An HP Instrument BASIC "RESET" closes all open files. Press the [**Local/Setup**] hardkey while the program is running or press the [reset] softkey in the [**BASIC**] [debug] menu when the program is paused.

Keystroke recorded programs do not open files and therefore avoid this problem.

Stopping a Program

To completely stop a program, press the [**Local/Setup**] hardkey at any time while the program is running. This causes an HP Instrument BASIC "RESET." If the analyzer is under remote control, pressing the [**Local/Setup**] hardkey *twice* also resets an HP Instrument BASIC program. (The first press bring the instrument back to local and second press resets the program). A STOP statement in your program terminates the program but does *not* perform the reset operation.

Note

While the program is executing an INPUT statement, pressing the [**Local/Setup**] brings the program under local (front panel) control. This enables the front panel's alpha keys. Pressing [**Local/Setup**] again, enters a "/" on the input line. In this case, press the [**Preset**] hardkey to abort the program.

Variables retain their value after an HP Instrument BASIC RESET. Press [**BASIC**] [debug] [examine variable] to examine variable values.

Pressing [**Preset**] also stops a running program.

For more information on the PAUSE and STOP statements see the "HP Instrument BASIC Language Reference" section of the *HP Instrument BASIC Users Handbook.*

Chapter 4

Saving and Recalling Programs

4

Saving and Recalling Programs

HP Instrument BASIC programs can reside in memory, on disk, or in an external computer.

Transferring Programs

From the front panel you can transfer a program between memory and disk with the [**Save / Recall**] menus. Within a program, you can use the GET, SAVE and RE-SAVE statements to transfer program files to and from disk. The HP 89400-series analyzers have an autoload feature which automatically recalls and runs a program from disk at power-up.

You can transfer a program file between the analyzer and an external controller. You can keystroke record a measurement sequence and then upload the program to the external controller for further editing. Programs developed on an external controller can be downloaded as well. Chapter 8, "Interfacing with the HP-IB," describes methods of transferring programs between the HP 89400-series analyzers and an external controller.

This chapter describes transferring HP Instrument BASIC programs between program memory and the HP 89400-series analyzers's volatile, non-volatile, internal, and external disk drives. The autoload feature is described at the end of this chapter.

Disk Formats and File Systems

To successfully transfer an HP Instrument BASIC program file, you must first understand the disk formats and file systems recognized by the HP 89400-series analyzers.

HP Instrument BASIC in the HP 89400-series analyzers recognizes two disk formats: LIF (Logical Interchange Format), and DOS (Disk Operating System). Formatting or initializing a disk determines the format of a disk or file system.

A LIF disk contains only one directory. This format should be used to exchange programs and data with other HP BASIC computers.

A DOS disk has a hierarchical structure of directories. The DOS format should be used to exchange data with DOS computers.

The *HP Instrument BASIC Users Handbook* refers to a third format, HFS (Hierarchical File System). The HP 89400-series analyzers do not support HFS.

File Types

The HP 89400-series analyzers support three file types:

- ASCII
- BDAT
- untyped (referred to as DOS or HP-UX files)

"File type" is independent of disk format. ASCII, BDAT and untyped files exist on either LIF or DOS disks. Untyped files appear as HP-UX in the catalog of a LIF disk, or as DOS in the catalog of a DOS disk. To view the catalog of the analyzer's default disk, press:

[**Disk Utility**]
[catalog on off]

In HP Instrument BASIC, the "CREATE ASCII" command creates an ASCII file, the "CREATE BDAT" command creates a BDAT file, and the "CREATE" command creates an untyped file.

For more information, refer to "Disk I/O" in chapter 10 and the "Data Storage and Retrieval" chapter in the "HP Instrument BASIC Programming Techniques" section of the *HP Instrument BASIC Users Handbook*.

Note

The [copy file] operation in the [**Disk Utility**] menu *does not* translate file types when you copy files across different file systems (DOS/LIF). Verify you are using the appropriate file type *before* copying a file.

DOS Conventions

On DOS disks, file names must conform to DOS conventions. File names are limited to 8 characters followed by a period and a three character extension. The period and extension are not required. File names are *not* case sensitive. For example, the following file names are equivalent:

```
PROG.ASC = Prog.ASC
```

The HP 89400-series analyzers donot allow "wild card" characters in file names. You can use a wild card in disk operations.

The HP 89400-series analyzers recognize a directory path. For example:

```
ASSIGN @File to "\DATA\TEST1\BEFORE"
```

opens the file named "BEFORE" in the sub-directory "TEST1" under the directory "DATA". Use a "\" or a "/" to separate directory and file names. The file specifier can include a directory path. However, you cannot use the HP 89400-series analyzers front panel keys to create a directory. You can create a directory with the CREATE DIR statement.

Using a DOS Disk to Transfer Data With a PC

You can transfer data from the HP 89400-series analyzers to an IBM-compatible personal computer by writing an HP Instrument BASIC program that outputs the data to a DOS disk.

To ensure a successful transfer, remember:

- *Specify the correct disk format.* Either format the disk on the PC, or use the HP 89400-series analyzers and the correct format option with the [format disk] operation in the [**Disk Utility**] [more utilities] menu. You can determine the format of a disk by looking at its catalog. Press [**Disk Utility**] [catalog on off].

- *Create untyped DOS files with the CREATE command.* Untyped files on DOS disks are extensible. They "grow" to the size needed. ASCII and BDAT files are not extensible. They usually cannot be read by other DOS applications.

- *Open files with the FORMAT ON option of the ASSIGN command.* FORMAT ON directs HP Instrument BASIC to store the data as ASCII characters.

You can also transfer HP Instrument BASIC programs to a personal computer using a DOS disk. The GET statement recalls HP Instrument BASIC programs from a DOS file into the analyzer's memory. The [re-save IBASIC program] softkey and the SAVE or RE-SAVE statement create untyped DOS files on a DOS disk.

For additional information about input and output operations, refer to "Front Panel Operations versus Keyword Statements" later in this chapter and "Disk I/O" in chapter 10, "Instrument Specific HP Instrument BASIC Features."

LIF Conventions

On LIF disks, file names must conform to LIF conventions. File names are limited to 10 characters which include all characters except ":", "<"and "|". Some LIF implementations do not allow lowercase letters.

LIF does not allow directories but you can label the disk with a volume name. The volume name is assigned at initialization.

Using a LIF Disk to Transfer Data with an HP BASIC computer

You can transfer data from the HP 89400-series analyzers to an HP BASIC computer by writing an HP Instrument BASIC program that outputs the data to a LIF disk.

To ensure a successful transfer, remember:

- *Specify the correct disk format.* Use the [format disk] operation in the [**Disk Utility**] menu. You can determine the format of a disk by looking at its catalog. Press [**Disk Utility**] [catalog on off].

- *Use the CREATE ASCII command to create an ASCII file.* A LIF protect code is not allowed on an ASCII file.

- *Use the CREATE BDAT command to create a BDAT file.* HP Instrument BASIC allows and supports a LIF protect code on a BDAT file.

- *Open files with the FORMAT option of the ASSIGN command.* FORMAT ON directs HP Instrument BASIC to store the data as ASCII characters. FORMAT OFF, which is faster and takes less space, defaults to BDAT representation.

You can also transfer HP Instrument BASIC programs to an HP BASIC computer using a LIF disk. The GET statement recalls HP Instrument BASIC programs from a LIF file into the analyzer's memory. The [re-save IBASIC program] softkey and the SAVE statement create ASCII files on a LIF disk.

For additional information about input and output operations, refer to "Front Panel Operations versus Keyword Statements" later in this chapter and "Disk I/O" in chapter 10, "Instrument Specific HP Instrument BASIC Features."

Memory

HP Instrument BASIC in the HP 89400-series analyzers supports four mass storage devices:

- A volatile RAM disk (:MEMORY,0,0).
- A non-volatile RAM disk (:MEMORY,0,1).
- An internal disk drive (:INTERNAL).
- An external disk drive (Hewlett-Packard Subset/80) (:EXTERNAL,7XX, uu).

To specify the default storage device, use the MASS STORAGE IS command.

Front Panel Operation versus Keyword Statements

There are two ways to transfer program contents between disk and memory. You can transfer programs by either:

- Using the [**Save / Recall**] key in the SYSTEM group on the front panel

or

- Using the keyword statements SAVE, RE-SAVE and GET

The choice of which to use requires some knowledge of the advantages of each as well as your own particular requirements. Both methods are discussed in the following section.

The [Save / Recall] Menu

With the [**Save / Recall**] menu you can perform a variety of disk and file operations, as well as transfer complete programs between the program buffer and any disk file. These menus have the following advantage:

- You can transfer programs directly between any file on disk and the program buffer.
- The analyzer allocates memory automatically when you recall a program.
- The utilities are similar to the other save and recall operations in the analyzer.
- You can select a file in the catalog without typing in the name.

The Keyword Statements (SAVE, RE-SAVE and GET)

In an HP Instrument BASIC program, the keyword statements SAVE, RE-SAVE, and GET, save all or part of that program to disk. They also merge a program with a program from disk.

The SAVE, RESAVE, and GET keyword statements have the following advantages over the [**Save / Recall**] menu:

- You can store parts of a program to disk.
- You can recall programs and append them at any line in the current program.
- They are familiar to HP BASIC programmers.

Saving a Program to Disk

To save the current contents of the program buffer to a disk file, use the [**Save / Recall**] menus. This is the same system used for all disk access in the HP 89400-series analyzers.

If you are saving a program to a new file name, press:

[**Save / Recall**]
[save more]
[save IBASIC program]

Enter the name of the disk file using your keyboard or the front panel alpha keys in the entry window. HP Instrument BASIC programs are stored as ASCII files on LIF disks and as untyped files on DOS disks.

Re-saving a program is similar to saving a file to a disk. In this case however, the disk already contains a file with the same file name. To overwrite the file, use [re-save IBASIC program] in the [save more] menu. Enter the file name with your keyboard or the front panel alphanumeric keys.

To make the re-save process easier, use the disk catalog to select a file name. The catalog describes the contents of the default disk. To use the catalog, press the following keys on your HP 89400-series analyzer:

1. [**Save / Recall**]
2. [catalog on off]
3. Use the knob to highlight the desired file name. The name appears in the entry window.
4. [save more]
5. [re-save IBASIC program]
6. [enter].

The analyzer automatically re-saves the file with the file name you selected.

Note

The [re-save IBASIC program] softkey overwrites the contents of the file. No backup file is created; the contents of the existing file are lost.

Recalling a Program from Disk

When you recall a program file from the disk, it is loaded into the program buffer. Any program recalled to the program buffer using the [**Save / Recall**] menus *overwrites* the current contents of the program buffer.

To recall a program file from the disk to the program buffer, press the following keys on your HP 89400-series analyzer:

1. [**Save / Recall**]
2. [recall more]
3. [recall IBASIC program]
4. Enter the file name in the entry window.
5. [enter].

As with any recall operation, you can use the catalog. Press the following keys on your HP 89400-series analyzer:

1. [**Save / Recall**]
2. [catalog on off]
3. Move the knob to select the file name. The name appears in the entry window.
4. Press [recall more].
5. [recall IBASIC program]
6. [enter].

The recalled program file is entered into the program buffer one line at a time and checked for syntax errors. Lines with syntax errors are commented out. The HP Instrument BASIC syntax error is displayed briefly in a pop-up message window. The error message is also written to the CRT. See chapter 5, Developing Programs, for information on allocating display partitions to view error messages.

Memory is allocated for the program's variables and working space (called the stack). When you use the [**Save / Recall**] menus to recall a program, memory is allocated automatically. For certain kinds of programs, the memory size may need to be increased.

See chapter 5 for more information on memory size.

Appending Program Files from Disk

To append program files from disk to the current program in memory, use the GET statement *within a program.* The GET statement recalls a specified file from the disk and appends it at a specified line in the current program (or at the beginning of the program if a line is not specified).

The following example program appends three program files to itself to build one functional program. It demonstrates how to merge files. It also provides a set of error-handling routines for your recorded programs.

The example program builds a shell composed of:

- an initialization program section.
- a typical keystroke recorded section.
- a cleanup section that contains error-traps and timeout-traps.

The core five line program (program lines 40 - 80) chains the other three programs segments to itself. *These five program statements must be deleted or commented out before you run the program.*

All of these files are on the HP 89400-series analyzers, Example Programs disk:

- SHELLBEG provides the setup and initialization.
- SHELLDEM is a typical keystroke recorded program.
- SHELLEND provides error-handling routines and cleanup.
- SHELLCHA pulls all files together using GET statements.

The file SHELLCHA contains the following program:

```
10      ! SHELLCHA: Program demonstrates chaining program segments
20      !-----------------------------------------------------------
30      ! NOTE: Delete Lines 30 thru 90 immediately after
40      !       running SHELLCHAIN program
50      GET "SHELLBEG:,4",X,60
60      GET "SHELLDEM:,4",X,70
70      GET "SHELLEND:,4",X,80
80      DISP "Delete lines 30 - 90 before running program"
90      GOTO Endlabel
100     X:END
```

Line 50 performs a GET of the file "SHELLBEG" from the internal disk drive. It appends the file at line 100 (labeled "X:") and overwrites that line. It then instructs the program to continue at line 60.

The "SHELLBEG" file has a label "X:" as its last line. The program in memory also has that same label as its last line. Line 60 performs a GET of the file "SHELLDEM," appends it at the current label "X:" and then continues the program at line 70.

The SHELLDEM file also contains a label "X:" as its last line. Line 70 of the SHELLCHA program performs a GET, which appends the SHELLEND program file to the end of the program in memory.

Finally, line 90 of the SHELLCHA program skips to the label "Endlabel". This label, which was at the end of the SHELLEND file, is now at the end of the program in memory. The program would go on to execute the SHELLBEG program if the ENDLABEL statement had been omitted. That is, without the ENDLABEL statement, the program would run itself immediately after appending the three program files.

Note

Remember to comment out or delete program lines 30-90 before running the combined program.

To use the SHELLCHA program with your own recorded program do the following:

1. Insert the label "X:" in the line containing the END statement of your recorded program. Make sure you have not used the label "X:" elsewhere in your program.
2. Recall the SHELLCHA program and change the file name in line 60 from "SHELLDEM" to the name of your program file.

Autoloading a Program

HP Instrument BASIC allows you to automatically load and run a designated program when you turn on the analyzer.

To make an autoloading program, save it with the file name "AUTO_BAS" either in non-volatile memory or on a floppy disk in the internal drive. At power-up, HP Instrument BASIC searches first the internal disk drive and then the non-volatile RAM for the file "AUTO_BAS." If it is found, HP Instrument BASIC loads and executes the program immediately.

Pressing the [**Preset**] hardkey while you turn on the analyzer, disables the AUTO_BAS program.

Chapter 5

Developing Programs

5

Developing Programs

Overview

For many applications, you can easily record and run programs without altering the program code that is generated with keystroke recording. However, with some knowledge of the HP Instrument BASIC language and the program development capabilities in the HP 89400-series analyzers, you can add immeasurable power to your recorded programs. You can also create programs without using the keystroke recording feature.

This chapter describes the operation of the keys under the [**BASIC**] menu. See figure 5-1. At the end of the chapter the [BASIC display format] softkey (in the [**BASIC**] menu) is discussed. This softkey presents a menu that lets you manage a part of the screen display for output from the program.

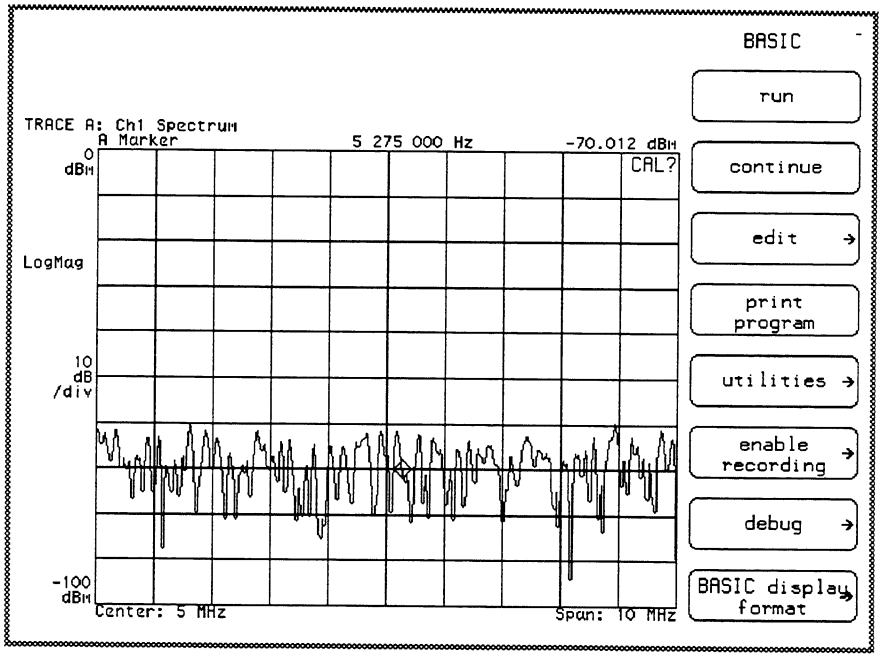

Figure 5-1. The [BASIC] Menu

The ability to change and enhance your program and its operating environment is found primarily under the [edit] and [utilities] menus.

- Pressing [edit] places you in the HP Instrument BASIC editor. You can make changes to your program on a line-by-line basis using a keyboard or the front panel alpha-numeric keys.

- Pressing [utilities] presents a menu of helpful utilities. You can make global changes to the program and its environment. You can renumber lines, allocate memory size, indent lines, secure the program, delete subprograms, and remove the program.

- Pressing the [print program] softkey prints a hard copy program listing to an attached HP-IB printer. The address of the printer is defined under [**Local/Setup**] [peripheral addresses].

Using the HP Instrument BASIC editor

The HP Instrument BASIC editor allows you to create and alter program text. If you are familiar with the HP 9000 Series 200/300 BASIC editor, you will find it similar. If not, you should find the HP Instrument BASIC editor easy to learn and to use. This section tells you how to enter and edit an HP Instrument BASIC program.

To start the editor, press the [edit] softkey in the [**BASIC**] menu. The program, if one exists, usually appears on the display with the cursor on the first line of the program. If the program buffer is empty, the first line number (10) appears with the cursor positioned to enter text.

The current program line (the line containing the cursor) always appears as two lines on the screen, allowing you to enter up to 102 characters. The other lines display the first 48 characters (excluding line numbers.

The first 6 columns of each line are a numeric field specifying the program line number. Line numbers are right justified. Program lines are automatically numbered by the editor. You can manually edit the current line number to copy or move it a to different location in the program. Line numbers can also be renumbered in blocks with the [utilities] [renumber] softkey menu. Line numbers range from 1 to 32766.

Once in the [edit] menu you can use your keyboard or the front panel alpha-numeric keys.

Using the HP Instrument BASIC Editor With a Keyboard

Using a keyboard makes developing HP Instrument BASIC programs easy.

All of the "typewriter" keys are enabled. Letters can be entered in lower or upper case. All punctuation marks and special characters can be entered using the HP approved PC keyboard. See figure 5-2.

A Typewriter keys
B Cursor keys
C Nurmeric keys
D Function keys
E Computer control keys
F Indicator lights

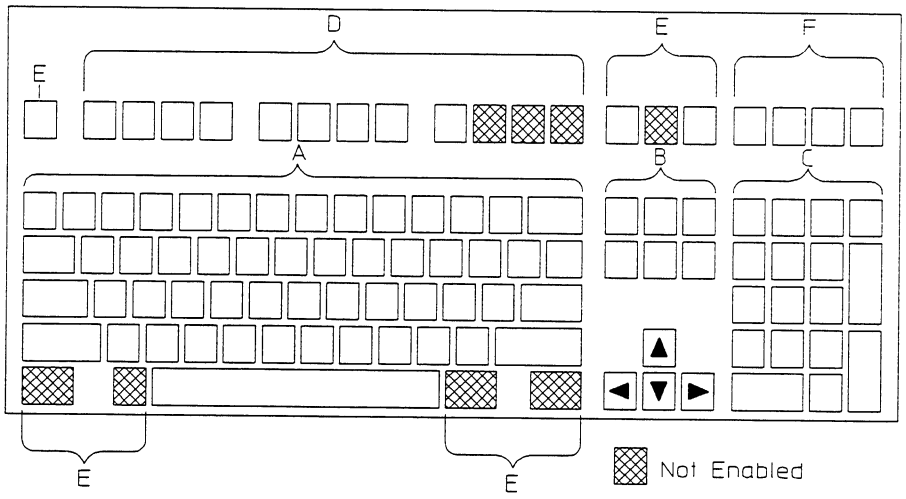

Figure 5-2. Using a keyboard with the HP 89400-series analyzers

The [**Enter**] key is used to store each line of program code and completes each alpha-numeric entry. The analyzer checks the line for syntax errors. If it detects an error, a pop-up message window displays the syntax error. If the analyzer does not detect an error, it stores the line.

Note

If you edit or enter text on the current program line and then move off the line without pressing the [**Enter**] key, all editing on the line is lost.

The [**Tab**] inserts two spaces. Pressing [**Shift**] [**Tab**] moves the cursor backwards two spaces.

The softkey menus in the HP 89400-series analyzers load into the keyboard function keys, [**F1**] through [**F8**]. The [Return] key loads into [**F9**]. [**F10**], [**F11**], and [**F12**] are not enabled. See figure 5-3.

The "cursor" keys are enabled. The arrow keys indicate the direction in which they move the cursor.

The [**Home**] key moves the cursor to the beginning of the current line. The [**End**] key moves the cursor to the end of the current line.

The [**Page Up**] key moves the cursor a maximum of 15 lines upward. The [**Page Down**] key moves the cursor a maximum of 15 lines downward.

The [**Insert**] key inserts a new line of text. To get out of the insert mode, press the [**Insert**] key again or move the cursor off the current line. Remember, to save an edit you must press [**Enter**] while the cursor is on the current line.

The [**Delete**] key erases the character where the cursor is positioned. In addition, all characters to the right of the deleted character move one character to the left.

Pressing the [**Shift**] [**Delete**] keys deletes the current program line.

Pressing the [**Alt**] [**Delete**] keys ([**Alt Gr**] [**Delete**] keys on a non-U.S. English keyboard) deletes all characters from the current cursor position to the end of the line.

The [**Alt**] key is not enabled except when used with the [**Delete**] key or to preset the analyzer.

Key presses made with the keyboard that have no meaning in a given operating context are ignored, just as they are when pressed from the front panel.

Caution

Pressing the [**Del**] key with the [**Alt**] key and the [**Ctrl**] key, presets the analyzer. (Just like a soft reboot in an IBM-compatible PC!)

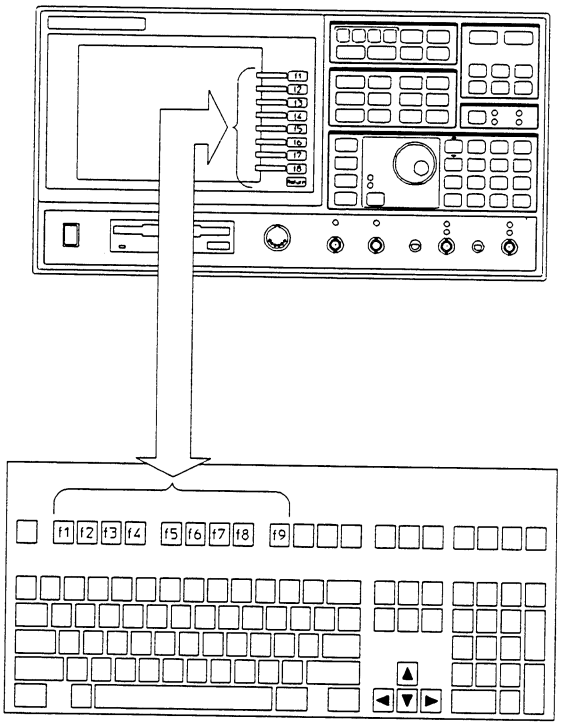

Figure 5-3. Mapping of the HP 89400 Series Softkeys

To end an editing session, press the [**f8**] key, which corresponds to pressing the [end edit] hardkey while in the [edit] menu. You can also exit the editor at any time by pressing the [**BASIC**] softkey. This returns you to the [**BASIC**] menu.

Connecting your keyboard

To connect the keyboard to your HP 89400-series analyzer, plug the round connector into the front of the analyzer. See figure 5-4.

Caution

Use only the HP approved keyboard on this product. HP does not warrant damage or performance loss caused by a nonHP approved keyboard.

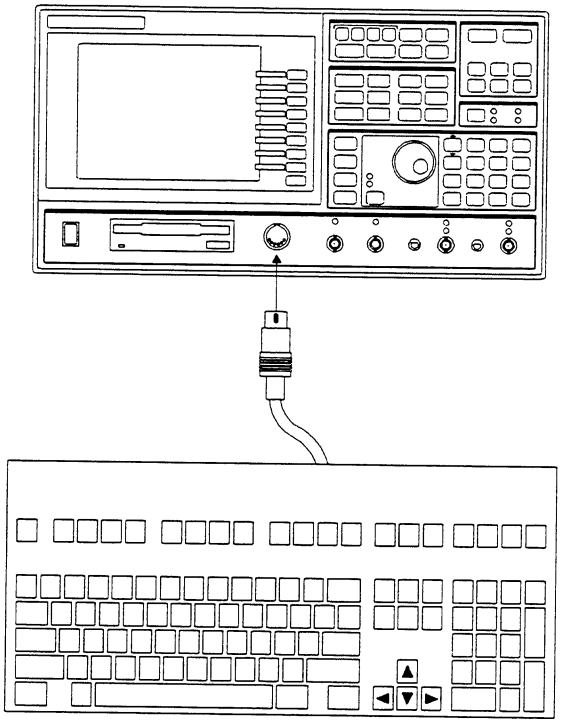

Figure 5-4. Connecting the keyboard.

If you are using an international keyboard, specify the type of keyboard with the [keyboard type] softkey in the [**System Utility**] menu under [more].

Using the [edit] Softkeys

The [edit] menu contains the softkeys shown in figure 5-5.

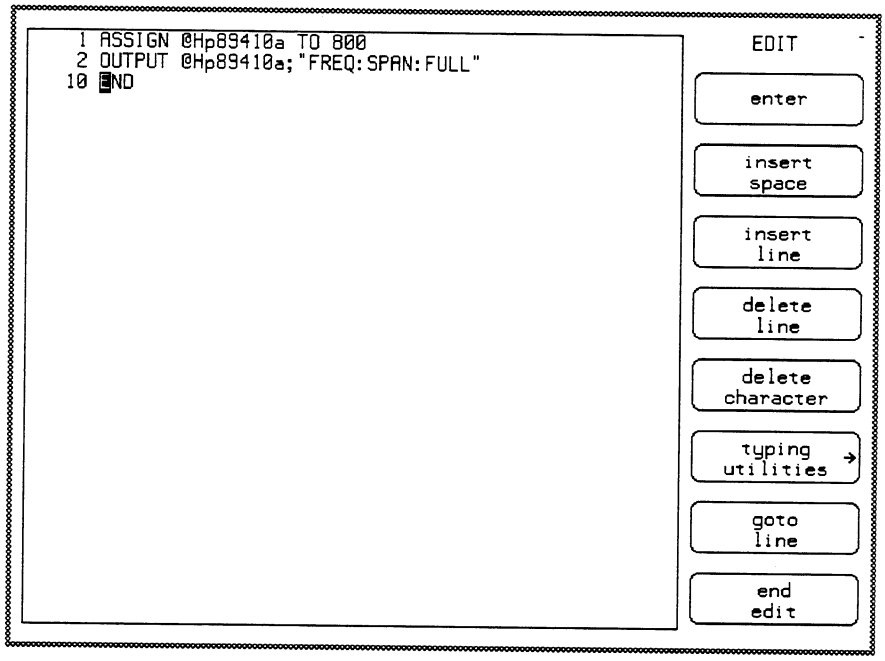

Figure 5-5. The [edit] menu

You can move the cursor around in the program using the knob on the front panel or the cursor keys on the keyboard. When you get to a line you want to change, make the change and press the [enter] softkey or the [**Enter**] key on the keyboard. The analyzer checks the line for syntax and then stores it if the syntax is correct.

Getting Around in the Program

You can move the cursor from line to line within an existing program by:

- Using the knob
- Using the up-arrow and down-arrow keys
- Using the [goto line] softkey to jump directly to a specific line number or label
- Using the [enter] softkey (when not in insert mode) to step one line at a time
- Using the [**Enter**] key on the keyboard
- Using the [**Page Up**] and [**Page Down**] keys on the keyboard
- Using the cursor keys on the keyboard

Using the Knob

You can move the cursor in the EDIT mode with the knob on the front panel.

The line that the cursor is on is always the edited line. Rotating the knob clockwise on the currently edited line moves the cursor to the right. Rotating the knob counterclockwise moves the cursor to the left. When the cursor is at the end (far right) of the edited line, turning the knob clockwise moves the cursor down to the end of the next line. Conversely, when the cursor is at the beginning (far left) of the edited line, turning the knob counterclockwise moves the cursor to the beginning of the preceding line.

Using the Up-Arrow and Down-Arrow Keys

You can move the cursor to a higher or lower line number by pressing the up-arrow or down-arrow front-panel hardkeys. The up-arrow hardkey moves the cursor to a lower line number and the down-arrow hardkey moves the cursor to a higher line number. Pressing and holding either of these keys moves through multiple lines. The cursor maintains the same relative position in each line.

Using the [**Shift**] key with the up-arrow or down-arrow keys moves the cursor upward or downward a page (a maximum of 15 lines).

Using the [goto line] Softkey

To jump immediately to any line or label in the program press the [goto line] softkey ([**f7**] on the keyboard). Enter the line number or the label of the line into the entry window and press [enter]. To specify a label, use the keyboard or use the front-panel keys and the optional [(_)] underscore softkey. You can enter the label in capital letters and it automatically converts to the proper case.

If the specified line exists, it appears in the middle of the display as the current program line. If you have specified a line number that doesn't exist, the cursor is placed on the line number closest to it. Specifying a non-existent line label generates an error message, "Line not found in this context."

A quick way to go to the last line of the program is to enter a number much larger than the largest possible program line number such as 99999 (or any number greater than 32766 or the last line number of your program).

Using the [enter] Softkey

You can use the [enter] softkey to move the cursor down one line at a time. All other softkeys that move the cursor alter program text.

Using the Keyboard

See the previous section, Using the HP Instrument BASIC Editor with a Keyboard, for a description of getting around the program using the keyboard.

Entering Program Lines

When you finish entering or changing a program line, store it by pressing [enter]. The analyzer checks the line for syntax errors and converts letter case to the required form for names and keywords (HP Instrument BASIC commands). If it detects an error, a pop-up message window displays the syntax error. If no errors are detected, it stores the line.

Note

If you edit or enter text on the current program line and then move off the line without pressing [enter], all editing on the line is lost.

Renumbering, Copying and Moving Lines

If you want to change the line number of an edited program line, move the cursor to the line number field and enter a new line number. Changing the line number copies the line. The line does not move. To move the line, change the line number, press [enter] and then delete the original line.

If you want to revise and move the current line, edit the line, change the line number and then press [enter]. The revision only appears in the copied line.

If you change the line number and you are in insert mode, you remain in insert mode at the new line number.

When the cursor is in the line number field, entries operate in an overtype mode rather than in the insert mode as in the text portion of the program line. The [**Back Space**] hardkey in the numeric keypad moves the cursor over line numbers without deleting the number.

Inserting Spaces

Use the [insert space] softkey to place a space at the position of the cursor. The text to the right of the cursor moves one place to the right. This softkey is located in more than one menu.

To insert a space with your keyboard, press the space bar.

Inserting Lines

You can easily insert one or more program lines above any existing line by placing the cursor on the existing line and pressing [insert line]. The [insert line] softkey toggles the insert mode on or off.

If you are using a keyboard, press the [**Insert**] key to insert a line. Pressing the function key [**F3**], also turns the insert mode on or off in the main [edit] submenu.

In the following example we use the [insert line] softkey to insert lines between two adjacent programs lines numbered 90 and 100.

Move the cursor to line 100 and press [insert line]. A new line, numbered 91, appears between line 90 and line 100. Press [Enter] to store the inserted line and another line appears numbered 92. If you continue to insert new lines and the inserted line number increments to 100, the current line 100 is renumbered to 101 to accommodate the inserted line.

To get out of insert mode, press [insert line] again or use the knob to move off of the current line. (Remember, any edits you make to the currently inserted line are lost if you leave insert mode without pressing [enter].) Make sure you have entered any changes to your final inserted line before exiting the insert mode.

Using the Front-Panel Alpha Keys

If you do not have a keyboard, you can use the alpha keys on the front panel of your HP 89400-series analyzer. Nearly every hardkey is labeled with a corresponding letter of the alphabet. These may be familiar to you if you have performed any editing function on an HP 89400-series analyzer, such as specifying a unique filename in a [**Save / Recall**] operation.

The alpha key labels appear below and to the right of the keys. They are arranged in alphabetical order from left to right, descending the front-panel hardkeys. See figure 5-6.

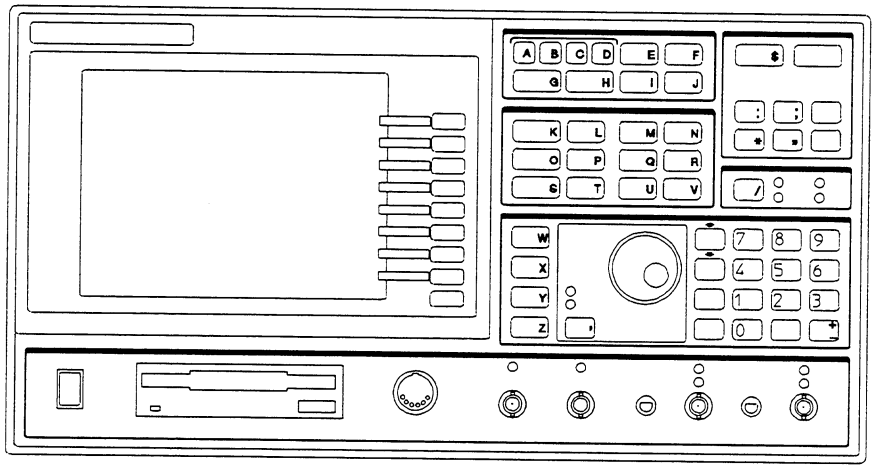

Figure 5-6. HP 89400 series alpha-numeric keys

You do not have to use the alpha keys to enter HP Instrument BASIC keywords. They can be entered via the [typing utilities] [insert keyword] menu. For more details on this feature see the section on entering keywords later in this chapter. The front panel alpha keys are necessary to enter variable names, constants, labels and strings if you are not using a keyboard.

When in the HP Instrument BASIC editor, the front-panel keys are automatically in alpha mode. Pressing an alpha key enters the character at the cursor position in the current program line. When you exit the editor [end edit] the front-panel keys return to their labeled hardkey function.

Recording into an Existing Program

Another way to enter lines into your program is to use the keystroke recording capabilities of HP Instrument BASIC. To record measurement sequence operations into your program, move the cursor to the line where you want the recorded statements inserted. Press [end edit], then press [enable recording] and proceed with your recording as you normally would. Press [**BASIC**] to conclude the recording session as usual.

The inserted recording acts the same as if you had pressed [insert lines] in the editor, and generated OUTPUT statements in insert mode.

The "ASSIGN @Hp89410a TO 800" statement is *not* generated when you are recording into an existing program. The "ASSIGN @Hp89410a TO 800" statement *must* be included in your program prior to any recorded OUTPUT commands. If you initially created the program using keystroke recording, this statement should already exist. If it does not exist, you will need to enter it.

Removing Program Text

The HP Instrument BASIC editor allows you to remove individual characters or entire lines. To learn how to remove the entire program see the description of the [utility][scratch] softkey later in this chapter.

Deleting Characters using a keyboard

The [**Delete**] key on the keyboard erases the character where the cursor is positioned. In addition, all characters to the right of the deleted character move one character to the left.

The [**Backspace**] key also removes text. The cursor moves one space to the left and usually erases any characters in the cursor's path. It does not erase characters in the program line number field.

Pressing the keys [**Alt**] [**Delete**] ([**Alt Gr**] [**Delete**] keys on a non-U.S. English keyboard) deletes all characters from the current cursor positon to the end of the line.

Deleting Characters using the [delete character] softkey

The [delete character] softkey, [**f5**] on the keyboard, removes the character under the cursor and moves all the following characters to the left one place. Repeatedly pressing [delete character] causes text to the right of the cursor to be pulled in and deleted. The [delete character] softkey functions the same in both the line number and program statement fields. However, in the line number field, only line numbers to the right of the cursor are pulled in and deleted. Program statement characters are not deleted when the cursor is in the line number field.

Another way to remove text on a line is with the [**Back Space**] key in the front panel's numeric key pad. Pressing [**Back Space**] removes the letter to the left of the cursor and moves the cursor (and all characters to the right of the cursor) one space to the left. When the cursor is on a line number, pressing the [**Back Space**] key simply moves the cursor back one position without deleting the number.

Deleting Lines using a keyboard

Pressing the [**Shift**] [**Delete**] keys removes the current program line and places it in a buffer. When the current program line disappears, all subsequent lines in the display move up one line, but are not renumbered. The cursor maintains its column-relative position on the next highest numbered line.

If the [**Shift**] [**Delete**] keys are pressed when the cursor is on the last program line, the line text is removed but the line number remains with the cursor resting in the first column. This puts the editor in insert mode on the last line of the program (see "Inserting Lines"). To get out of insert mode, use the knob or up-arrow key and move the cursor up one line.

Pressing the [**Shift**] [**Delete**] keys will *not* remove a subprogram line with a SUB keyword in it unless all program lines belonging to that subprogram are deleted first. You can delete entire subprograms by using utilities described later in this chapter under Delsub and Delsub to End.

Deleting Lines using the [delete line] softkey

The [delete line] softkey, [**f4**] on the keyboard, removes the current program line in the same manner as pressing the [**Shift**] [**Delete**] keys on the keyboard.

To recall the last deleted line, press the [recall line] softkey in the [typing utilities] menu.

Using [typing utilities]

The [typing utilities] softkey allows you to enter non-alphabetic symbols and insert HP Instrument BASIC keywords without a keyboard. See figure 5-7.

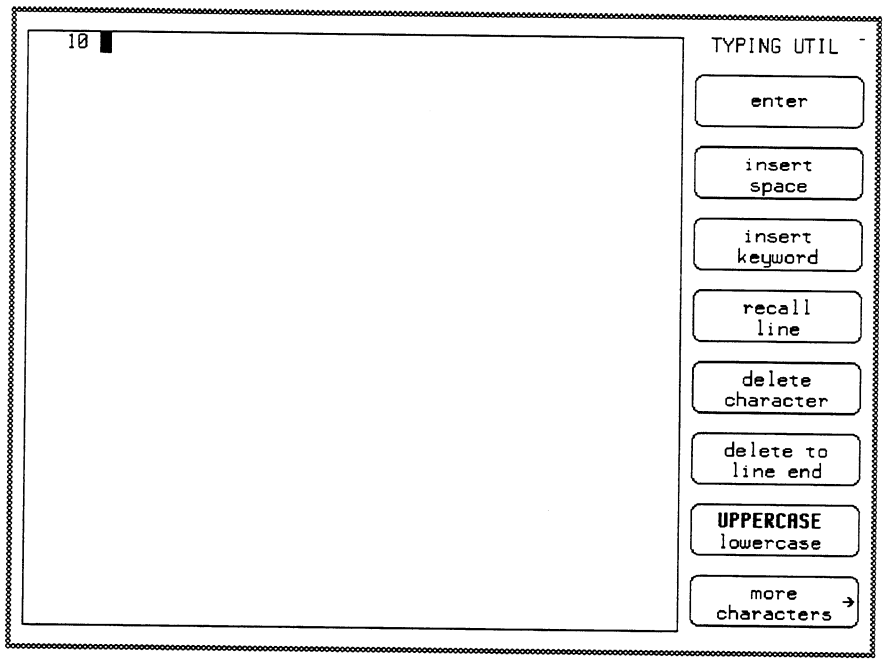

Figure 5-7. The [typing utilities] menu

The [enter], [insert space], and [delete character] keys are carried over from the [edit] menu. In addition, this menu contains the [insert keyword], [recall line], [delete to line end], [UPPERCASE lower case], and [more characters] keys. These keys are described in more detail below.

Entering Keywords

You do not have to type an entire HP Instrument BASIC keyword if you use the [insert keyword] softkey in the [typing utilities] menu. When you are in the [insert keyword] menu, pressing any alpha key presents a menu of keywords beginning with that letter.

For example, pressing [insert keyword] and then the alpha key "A" (the [**A**] hardkey) presents a menu with the following softkeys:

- [ABORT]
- [ABS( ]
- [ACS( ]
- [ALLOCATE]
- [ALPHA]
- [AND]
- [more]
- [cancel]

Pressing any one of the first six softkeys enters the corresponding text into the current program line. Keywords are always inserted in uppercase regardless of the current setting of the [**UPPERCASE** lowercase] softkey.

In cases where there are more than seven keywords starting with a particular letter, the softkey labeled [more] appears which allows you to access the rest of the keywords of that letter. When the last set of keywords is displayed, press [more] to get back to the first set. This allows you to cycle through all the keywords starting with a specific letter.

After pressing [insert keyword] you can skip from one keyword menu to another simply by pressing another front-panel alpha entry key.

Notice that all keywords that require an argument are provided with the beginning parenthesis; for example, [ASN( ]. The parenthesis indicates the keyword as requiring an argument. When this keyword is selected, the keyword and both parentheses are inserted in the program line with the cursor placed automatically between the two. All keywords that require an argument are inserted this way.

To return to the previous menu without selecting a keyword, press [cancel].

Recalling Deleted Lines

If you used the [delete line] key to remove a line, the [recall line] key automatically recalls that line. This is useful for recovering lines deleted by mistake.

It is also useful for moving a line. Use the [delete line] key in the [edit] menu, move to the desired area of the program and press [insert line] ([**F3**] on the keyboard). Press [recall line] under the [typing utilities] softkey ([**F4**] on the keyboard) and then edit the recalled line to the current line number.

Note

Pressing [recall line] automatically aborts any changes made to the currently edited line.

Deleting part of a line

Pressing the [delete to line end] key deletes all characters from the current cursor position to the end of the line. You may then edit the remaining line before pressing [enter]. You can recover the old line by moving off the line any time prior to pressing [enter].

Changing Case

The case of an alpha key is determined by the state of the [typing utilities] [**UPPERCASE** lowercase] key. The default is uppercase. To enter lowercase letters, press the [UPPERCASE lowercase] key.

If you are using the keyboard, the case is determined by the [**Caps Lock**] key.

Entering Symbols

Symbols are available in three menus under [typing utilities] [more characters]. Each menu is labeled "more chars" followed by a list of the available symbols in that menu. See figure 5-8.

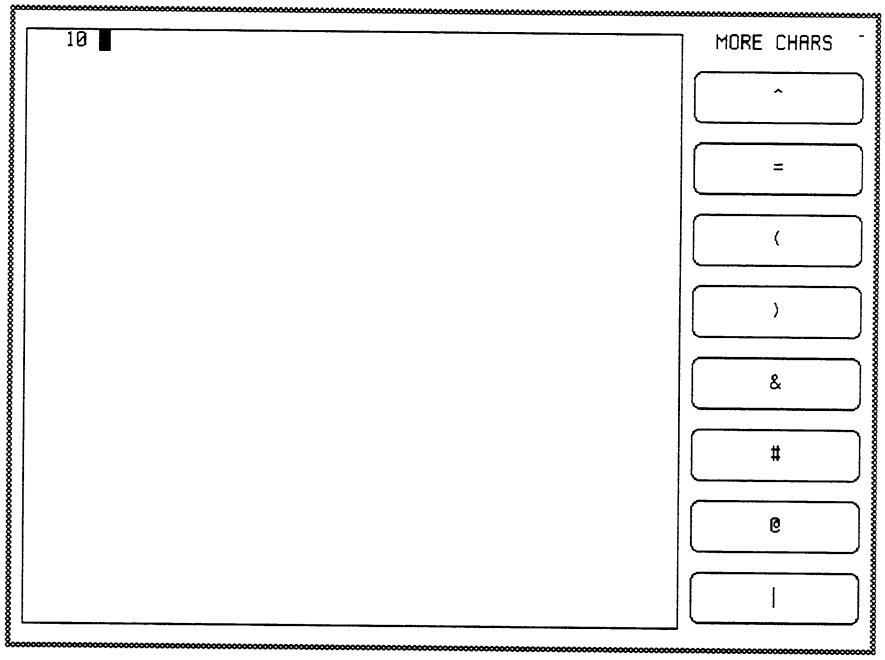

Figure 5-8. A [more chars] menu

For example, to enter a pound sign (#), press [typing utilities] [more characters]. This brings up three [more chars] softkey menus, each containing symbols which are listed in the label as a separate softkey when the appropriate [more chars] softkey is pressed.. Press the softkey labeled [more chars ^=( )&#@ |] softkey then [#]. Return to previous menus by pressing [Return] one or more times.

Note that additional symbols, for example * : ; , " + - $ / are available as remapped hardkeys.

Using [utilities]

There are some activities generally associated with editing that are located outside the [edit] menu, under the [**BASIC**] [utilities] softkey. These editing utilities are more global in nature, rather than pertaining to single characters, words and lines as the editor does. See figure 5-9.

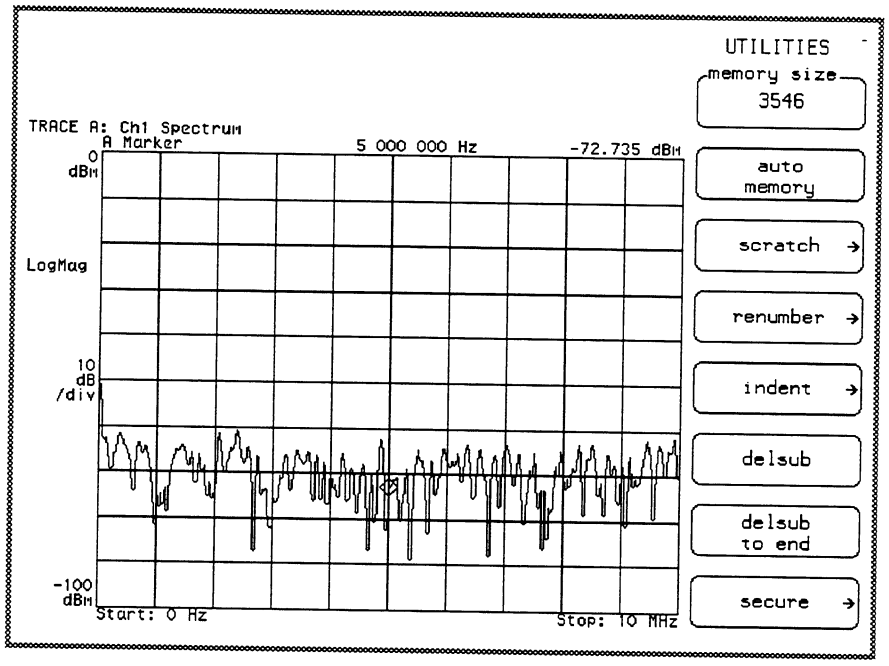

Figure 5-9. The [utilities] menu

If using a keyboard, the [utilities] menu assigns the function keys as follows:

[memory size]	[**F1**]
[auto memory]	[**F2**]
[scratch]	[**F3**]
[renumber]	[**F4**]
[indent]	[**F5**]
[delsub]	[**F6**]
[delsub to end]	[**F7**]
[secure]	[**F8**]

The [utilities] menu is mostly composed of HP Instrument BASIC keywords that can be executed interactively. All but [memory size] and [auto memory] directly affect the contents of the program. The [memory size] and [auto memory] softkeys allow you to directly change the program's operating space.

Memory Size

The [memory size] softkey displays the amount of working space (commonly called the stack) currently allocated for the program. The stack contains all variables not in COM as well as context information for functions and subprogram calls. The stack does not contain program code.

HP Instrument BASIC allocates the size of the stack for the most efficient use of memory resources. If you use recursive subprograms, HP Instrument BASIC may not allocate enough memory. If the analyzer runs out of stack space while the program is running it displays an error message, "Out of Memory" in a pop-up message window.

To increase the amount of memory allocated for the stack, press [memory size]. The entry window is then displayed. Enter the new amount using the numeric keys on the front panel or on the keyboard. The entry window is displayed when you press the first numeric key. Use the [exponent] softkey to enter the size using engineering notation. After entering the new memory size, press [enter]. HP Instrument BASIC may adjust your entry to the closest available increment of memory.

If you enter a number which exceeds the available memory, the memory size will be set to the largest available stack size. The minimum amount allocated by HP Instrument BASIC for the stack is 1200 bytes.

Memory available for HP Instrument BASIC programs is dependent upon the amount of memory space allocated for other uses. To display the usage of all of the analyzer's memory, press the [**System Utility**] hardkey then press [memory usage]. A table displays the amount of memory allocated for HP Instrument BASIC programs (code and program stacks) as well as memory allocated for other functions of the analyzer.

Auto Memory

The [auto memory] softkey resizes stack space automatically to fit the current program. This is similar to the operation that occurs when a program is loaded with the [**Save / Recall**] menus. This is faster than using the [memory size] key and works well for most programs.

In some cases, [auto memory] may allocate more memory than the HP Instrument BASIC program needs. Use the [memory size] softkey to reduce the amount of memory allocated for your program. If you receive an "Out of Memory" error when you try to run the program you can use the [memory size] softkey to increase the memory size. Programs that use recursive functions or subprograms may need to have memory increased manually with the [memory size] softkey.

Scratch

Pressing the [scratch] softkey brings up a menu that allows you to clear the current program and/or variables. The softkeys load into the keyboard function keys as follows:

[scratch]	[f1]
[scratch C]	[f2]
[scratch A]	[f3]
[perform scratch]	[f5]

You must first select a combination of program and/or variables to clear by pressing [scratch], [scratch C], or [scratch A]. The scratch operation is not executed, however, until you press the [perform scratch] softkey.

scratch

This key selects the current active HP Instrument BASIC program and all variables not in COM.

scratch c

This key selects all variables including those in COM, but does not clear the program.

scratch a

This softkey selects the current active HP Instrument BASIC program and all variables, including those in COM.

The analyzer does not clear the memory until you press [perform scratch]. To cancel a SCRATCH operation, press [Return] at any time prior to pressing [perform scratch].

Renumber

Pressing [renumber] displays a menu that allows you to change the line numbering for the entire active program. The [renumber] menu loads into the keyboard function keys as follows:

[start line #]	[f1]
[increment]	[f2]
[perform renumber]	[f4]

To select the number that is assigned to the first line in the program when renumbering lines in a program, press [Start line #]. If you do not define the starting line number, the first line number defaults to 10.

Press [increment] to specify the increment between the renumbered line numbers. The default is 10. For example, if [start line #] is 10 and [increment] is 5, the line numbers will be 10 . . . 15 . . . 20 . . . 25 . . . and so on.

Once these parameters are defined, press [perform renumber] to execute the command. To cancel the renumbering operation, press [Return] at any time prior to pressing [renumber program].

Indent

Pressing [indent] displays a menu that allows you to change the indentation for the entire program. The [indent] menu loads into the keyboard function keys as follows:

[start column]	[f1]
[increment]	[f2]
[perform indent]	[f4]

Press [start column] to specify the column number in which the first character of the first statement should appear. Press [increment] to specify the number of spaces each line should move to the right or left when the nesting level of the program changes.

Once these parameters are defined, press [perform indent] to execute the command. To cancel the renumbering operation, press [Return] at any time prior to pressing [perform indent].

Delsub and Delsub to end

The [delsub] and [delsub to end] softkeys allow you to delete subprograms and functions from your program. When you press [delsub], you are prompted to enter the name of the *single* subprogram or function you want to delete. Once you have typed the name in the prompt, press [enter] to delete the subprogram or function, or press [Return] to cancel the operation.

When you press [delsub to end], you are prompted to enter the name of the *first* subprogram or function you want to delete. Once you have typed the name in the prompt, press [enter] to delete the subprogram or function—and all subprograms or functions that follow it—or press [Return] to cancel the operation.

Secure

The [secure] menu allows you to "protect" program lines. "Protected" program lines cannot be listed to a printer or viewed in EDIT mode.The [secure] softkeys load into the keyboard function keys as follows:

[start line #]	[f1]
[end line #]	[f2]
[perform secure]	[f4]

To secure a block of the active program:

1. Press [start line #].
2. Enter the beginning line number of the program block. (The value defaults to 1.)
3. Press [end line #].
4. Enter the ending line number of the program block. (The value defaults to 32766.)
5. Press [perform secure].

Since [start line #] value defaults to 1 and the [end line #] value defaults to 32766, you can secure the entire program by pressing [perform secure] without altering the start line and end line values.

Secured lines cannot be printed or viewed in the editor. They appear only as an asterisk following the line number (*). Secured lines can, however, be deleted from the program using the editor. You may leave this menu at any time by pressing [Return].

Caution

Secured program lines cannot be unsecured. Be sure to keep an unsecured version of the program for your own records.

Using [print program]

The [print program] softkey allows you to print the current contents of the program buffer to a printer attached to the HP-IB interface. The printer must be correctly set up under [**Local/Setup**] hardkey and must be selected under the [**Plot/Print**] hardkey. See the *Getting Started Guide* shipped with your HP 89400-series analyzer for details or setting up and printing to peripherals. To enable printing over the HP-IB, the analyzer must be set as the System Controller in the [**Local/Setup**] menu.

Note

If you press [print program] and do not have a printer connected or properly configured, HP Instrument BASIC continues attempting to print until you either press [**Local/Setup**] or [**Preset**].

Using [display format]

Pressing the [**BASIC**] [display format] key allows you to allocate a partition of the analyzer's display to be used by your program. Alternately, HP Instrument BASIC can return any allocated partition of the display to the analyzer.

The display in the HP 89400-series analyzers is divided into two small partition areas (UPPER, and LOWER) and one large area (FULL), which encompasses both the UPPER and LOWER partition areas. See figure 5-10.

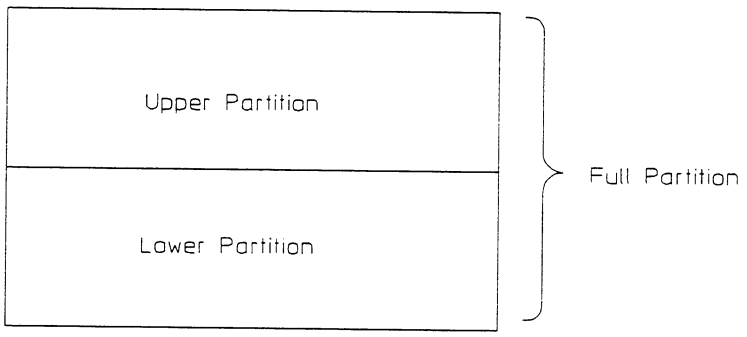

Figure 5-10. The Display Partitions

All screen output commands, such as PRINT and DRAW, require that you allocate a partition of the screen in order to view the results of the command. This can be performed in your program or interactively using the [BASIC display format] softkey. See figure 5-11.

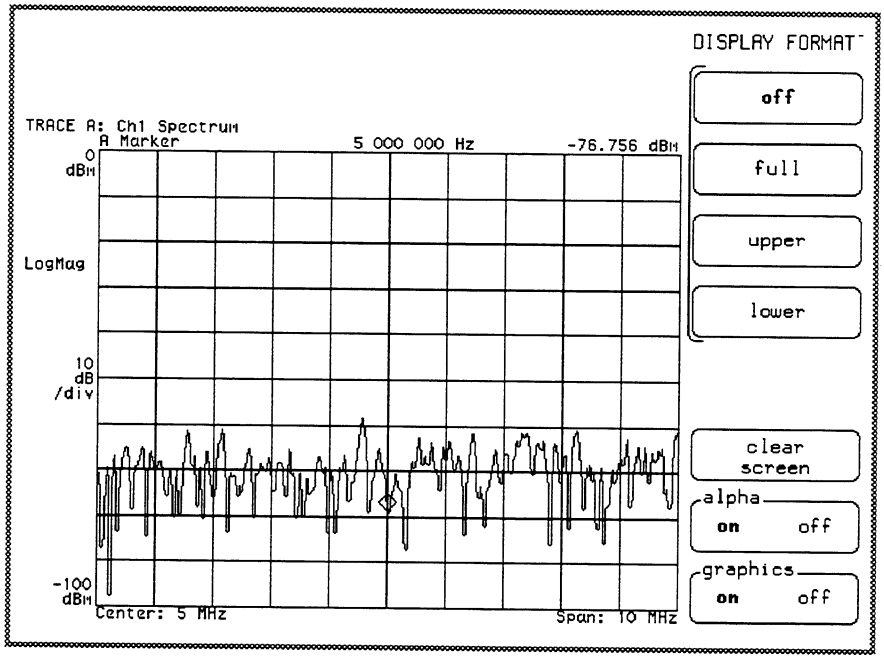

Figure 5-11. The [BASIC display format] menu

The [BASIC display format] menu softkeys load into the following keyboard function keys:

[off]	[**F1**]
[full]	[**F2**]
[upper]	[**F3**]
[lower]	[**F4**]
[clear screen]	[**F6**]
[alpha on off]	[**F7**]
[graphics on off]	[**F8**]

The [BASIC display format] softkey also appears under the [**Display**] hardkey.

You can allocate display partitions from within your program using the HP-IB command "DISP:PROG" and specifying the parameter UPPer, LOWer or FULL. For example the statement

```
OUTPUT 800;"DISP:PROG FULL"
```

allocates the single trace box of the display. Table 5-1 shows the relationship between the [BASIC display format] softkeys and the corresponding HP-IB commands required to program the same functions.

Table 5-1. The Display Partitions

MENU	ALLOCATES	HP-IB Command
OFF	NO DISPLAY	DISP:PROG OFF
FULL	SINGLE TRACE AREA	DISP:PROG FULL
UPPER	UPPER TRACE AREA	DISP:PROG UPP
LOWER	LOWER TRACE AREA	DISP:PROG LOW

Most display allocation should be handled by your program with the HP-IB commands. It is best to use these softkeys during program development.

[clear screen] clears all text and graphics from whichever partition is active.

[alpha on off] enables and disables the display of alpha output in the active partition.
[graphics on off] enables and disables the display of graphics output in the active partition.

For more information about controlling the display, refer to chapter 7, "Graphics and Display Techniques."

Chapter 6

Debugging Programs

6

Debugging Programs

The process of creating programs usually involves correcting errors. You can minimize these errors by using keystroke recording for your measurement sequence program segments and by writing structured, well-designed programs.

Of course bugs can and do appear in even the best designed programs. HP Instrument BASIC contains some useful features to help you track them down.

Overview

The HP Instrument BASIC tools provided for program debugging are simple and, if used properly, can be very helpful. The [**BASIC**] menu contains the [debug] softkey. See figure 6-1.

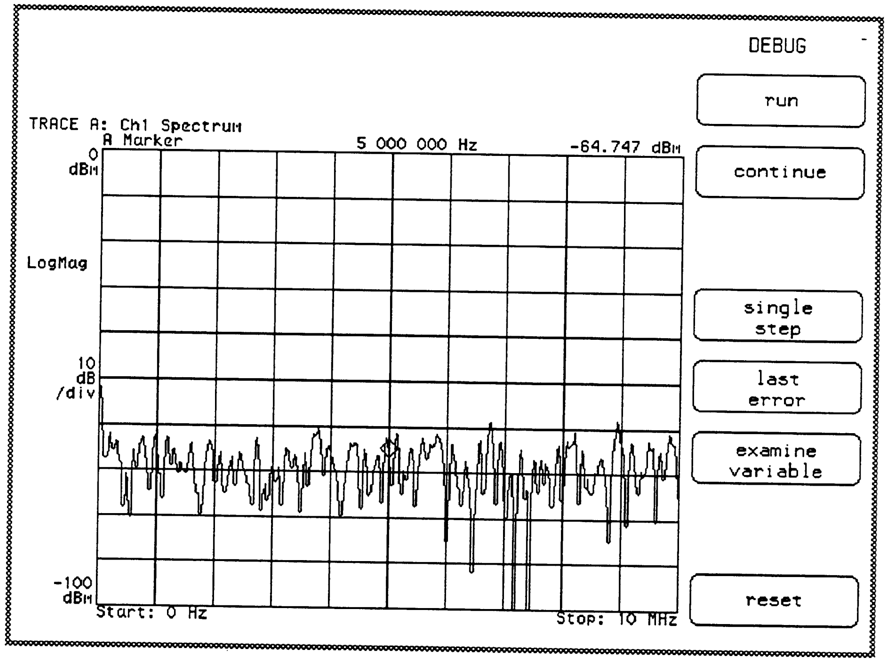

Figure 6-1. The menu

If using a keyboard, the [DEBUG] menu loads into the function keys as follows:

[run]	[**F1**]
[continue]	[**F2**]
[single step]	[**F4**]
[last error]	[**F5**]
[examine variable]	[**F6**]
[reset]	[**F8**]

The [debug] menu provides several debugging facilities. For example, using the [debug] menu you can:

- run or continue your program normally
- single step through your program one line at a time
- Display the last error encountered in your program
- Examine program variables
- Reset the program

By examining the values assigned to variables at various places in the program, you can get a much better idea of what is really happening in your program.

Use the [single step] softkey to execute the program one line at a time. You can study the program's operation and examine variable values.

By inserting a PAUSE statement in your program you can pause the program at any line and then examine the values of variables at that point in the program. Press [continue] to resume operation to the next PAUSE statement or to the end of the program. Press [single step] to walk through program lines following the PAUSE statement.

By combining these different features you can examine the program's operation and solve your particular problem.

Using [examine variable]

Pressing [examine variable] displays an entry window that allows you to enter the name of the variable you want to examine. The default is the name of the last examined variable. It also brings up the the alpha entry menu, so you can enter the variable name.

You must first perform a prerun operation to examine the value assigned to any variable in your program. A prerun is executed when you press either [run] or [single step]. After the prerun, press the [examine variable] key and enter the name of an existing variable in your program.

You can enter the variable as all uppercase letters. When you are finished entering a variable name, press [enter].

If you use [single step] and the program has not executed the line assigning that variable, the variable returns a value of zero.

Examining Strings

Enter string variables as you would any other variable. The entry window wraps to display a maximum of 10 lines of 42 characters each.

To select only a section of a string, use the HP Instrument BASIC substring syntax (see the "HP Instrument BASIC Programming Techniques" section in the *HP Instrument BASIC Users Handbook*). For example, to examine the 7 character substring starting at the second character of A$ enter:

```
A$[2;7]
```

Examining Arrays

You can examine an entire array or individual elements of the array. For example the entry:

```
I_array(1),I_array(2),I_array(3)
```

displays the elements 1 through 3 of the array **I_array**.

To select an entire array for examination enter the array variable name followed by an asterisk, (*); for example, I_array(*).

Example

I_array(20) is an integer array. The first and second elements are set to 100. After pressing [examine variables], enter "I_array(*)." The following is displayed:

```
I_array(*)  =  100  100  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0
```

An individual array element (for example, I_array(17)), is specified the same as any single variable.

Setting Breakpoints

A common method of debugging a program is the use of breakpoints. A breakpoint causes the program to stop at a defined point so that you can examine the program state at that point. In HP Instrument BASIC this is accomplished by inserting PAUSE statements in the program code. When the program runs, you can use [examine variable] to check or change variable values. Press [continue] to continue the program until the next PAUSE, STOP or END statement is encountered.

You can enter PAUSE statements and otherwise alter the contents of the active program by using the [**BASIC**] [edit] softkey. See chapter 5, "Developing Programs," for a description of the HP Instrument BASIC editing capabilities in the HP 89400-series analyzers.

Using [single step]

The [single step] softkey allows you to execute your program one line at a time. The line to be executed appears in the first line of the display. See figure 6-2.

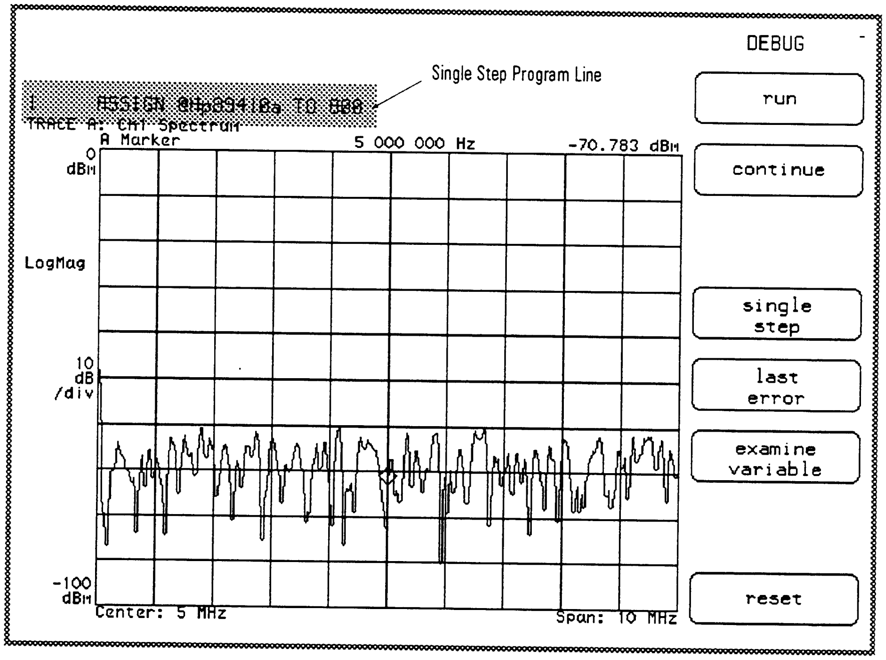

Figure 6-1. The menu

You can use [single step] from the beginning of the program or from any point where it has been paused. To resume regular execution of a program after using [single step], press [continue].

[single step] can be very helpful when used in conjunction with the [examine variables] key and the PAUSE statement. By placing a PAUSE statement at a point of interest in your program, you can run the program until it pauses, then single step through the critical program lines, checking variables values or program operation. To resume program execution at any time, press [continue].

Using [run], [continue], and [last error]

The [run] softkey operates the same as pressing [**BASIC**] [run]. It executes a prerun sequence and then begins executing the program at the first program line and continues until it reaches a PAUSE, STOP or END statement, or until the program is paused or stopped from the front panel.

The [continue] softkey allows you to resume regular program operation from a paused program or from a program in single step mode. This is identical to the [**BASIC**] [continue] softkey operation.

The [last error] softkey displays the error number and message of the last error encountered by the program. This is the front panel equivalent to the HP BASIC command, ERRM$.

Using [reset]

The [reset] softkey allows you to bring the program environment back to its default state. This is especially useful when you are using single step mode and you want to restart the program. Pressing [reset] closes all open files, sends an abort message to the HP-IB interface and resets the program counter to the first program line.

Chapter 7

Graphics and Display Techniques

7

Graphics and Display Techniques

HP Instrument BASIC programs have the ability to allocate portions of the instrument's display for text and graphics. This section provides a description of the various programming techniques used to do both.

Using the Partitions

There are several HP Instrument BASIC commands that require a display as an output device. These include commands such as PRINT, CLEAR SCREEN, MOVE, DRAW and GCLEAR. Since HP Instrument BASIC programs share all hardware resources with the instrument, the display must be shared for instrument and program use. All commands that output data to the screen write to a screen buffer and in order to view this output buffer, a portion of the display must be released from the instrument. You can do this manually when the program is not running by using the [**BASIC**] [BASIC display format] softkey menu. Performing equivalent actions from within a program that is running, requires sending an HP-IB command to the instrument; both to borrow a screen partition and again to give it back.

Allocating Partitions

The instrument's screen can be divided into two trace boxes (upper and lower). The upper and lower trace boxes can be combined into one large trace box for single trace displays. Any of these three trace boxes, called display partitions, can be used by an HP Instrument BASIC program.

There are two other non-partition areas of the screen that can be accessed by HP Instrument BASIC programs. The area on the right of the screen is reserved for softkey labels and can be accessed using the ON KEY statement. Also, a line at the top of the screen can be accessed via the DISP and INPUT statements.

To request one of the partitions from the analyzer, send the instrument the corresponding HP-IB command. "DISP:PROG UPP" allocates the upper partition, "DISP:PROG LOW" allocates the lower partition, and "DISP:PROG FULL" allocates the full screen partition. See figure 7-1.

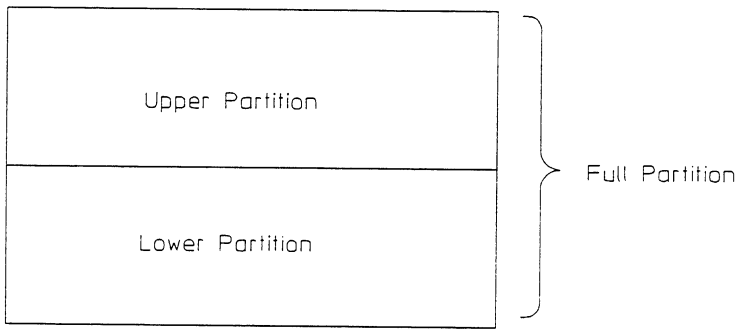

Figure 7-1. The Display Partitions

The following is an example of a program segment that prints a message to the upper trace box:

```
30  ASSIGN @Hp89410a TO 800
40  OUTPUT @Hp89410a;"DISP:PROG UPP"
50  CLEAR SCREEN
60  PRINT "This is the upper partition"
```

To be sure that you are writing to a partition that has not been assigned, include a WAIT statement. Or, add an HP-IB query command followed by an ENTER statement to synchronize the program with the instrument. The previous example would look like this:

```
30  ASSIGN @Hp89410a TO 800
40  OUTPUT @Hp89410a;"DISP:PROG UPP"
50  OUTPUT @Hp89410a;"DISP:PROG?"
60  ENTER  @Hp89410a;Part$
70  CLEAR SCREEN
80  PRINT "This is the upper partition"
```

The command DISP:PROG? (line 50 above) requests the instrument to send the current partition status. The ENTER statement on the next line reads that status and then continues.

De-Allocating Partitions

To return the display partition to the analyzer, use the "DISP:PROG OFF" command. This should be done before the termination of any program that has allocated a display partition. It may also be required within the program to allow someone to view instrument trace data. The following example demonstrates this command:

```
830  OUTPUT @Hp89410a;"DISP:PROG OFF"
```

Using Text

Most of the text capabilities of HP Instrument BASIC are covered in detail in the "HP Instrument BASIC Programming Techniques" section of the *HP Instrument BASIC Users Handbook*.

You can enable the display of text information by pressing [**BASIC**] [BASIC display format] [alpha on off]. This information is generated primarily by the HP Instrument BASIC PRINT statement.

The PRINT statement works the same in every partition. Information is printed starting at the top of the current partition and continues until the bottom of the partition is reached where the screen then scrolls up to allow additional lines to be printed. Causing the screen to scroll does *not* affect any graphics displayed on the screen, because text and graphics are written to different planes of the display.

All partitions have a width of 54 characters. The height varies according to partition. Both upper and lower partitions each contain 12 lines. The full partition contains 24 lines.

This information is useful if you are using the "PRINT TABXY" statement to position text. For example, the following program segment prints a message in the center of the full partition (assuming it has been allocated earlier in the program).

```
.
.
100  Maxlines=24
110  PRINT TABXY(23,Maxlines/2);"CENTER"
.
.
```

The following program segment demonstrates a technique to get text onto the screen quickly. Write your display message to a long string, using the OUTPUT statement, and then print the string to the screen. This speeds up screen display time considerably.

```
60   DIM Temp$[100],Big$[2000]
70   OUTPUT Temp$;"This is the first line of text"
80   Big$=Big$&Temp$
90   OUTPUT Temp$;"This is the second line of text"
100  Big$=Big$&Temp$
110  PRINTER IS CRT; WIDTH 2000
120  PRINT Big$
```

You can also print to the screen using the OUTPUT statement in conjunction with the display address (1). For example, the statement

```
OUTPUT CRT;" OUTPUT 1 WORKS WELL TOO"
```

writes the quoted text to the screen.

The display responds to several of the standard ASCII "control codes." These characters can be sent to the CRT by printing or sending the CHR$ function of the ASCII number. For example, CHR$(7) is the control code for the "bell" (CTRL-G) and has the effect of sounding the beeper. For more information on control codes recognized by the CRT, see the *HP Instrument BASIC Users Handbook*.

Note

It is sometimes a practice to embed these control codes in your PRINT statements when using external computers to develop programs. For example, the HP 9836 Series 200 Workstation allows you to enter control characters directly into the program using the "ANY CHAR" key. If you do this, do not attempt to use the HP Instrument BASIC editor on the program. This editor does not recognize embedded control codes and its actions may be unpredictable.

Using Graphics

You can enable the display of graphics information by pressing [**BASIC**] [BASIC display format] [graphics on off]. This information is generated primarily by the HP Instrument BASIC graphics statements.

Graphics and Display Partitions

You can position a graphics display area anywhere within the FULL display partition using the VIEWPORT statement. However, when you select the UPPER or LOWER partition, the analyzer shows you only those graphics that would be displayed in the lower half of the FULL partition. You must define a display area that falls within this lower half if you want to ensure that all graphics output is displayed within an UPPER or LOWER partition. For example, the following program line defines a display area that completely fills the lower half of the FULL partition:

```
VIEWPORT 0,RATIO*100,0,49
```

Graphics Line Buffering

When lines are drawn by a graphics statement, the endpoint coordinates and pen information for each line is normally saved in a graphics buffer in the analyzer's memory. This allows the lines to be redrawn automatically whenever you change the display partition. However, as the complexity of a graphic increases, the amount of memory required for the buffer also increases. You can prevent lines from being saved to the buffer, conserving the memory they would require, by sending the following HP-IB command: "DISP:PROG:VECT:BUFF OFF". When you want lines to be saved again, send "DISP:PROG:VECT:BUFF ON".

Graphics Pens

The PEN statement determines the color of the lines drawn by the graphics statements. The default pen is 1.

When you use the pen number "0," graphics statements erase lines—or more exactly, they erase those portions of any graphic elements that lie along the drawing path. The pen numbers used to draw a graphic on the analyzer's screen are also used to plot the same graphic on an external plotter. So if you use multiple pen numbers, the graphic will be drawn properly on the screen and plotted properly on a color plotter.

The PEN statement affects graphics displays only. Text is unaffected by PEN statements.

Table 7-1. Pens and Associated Graphics Colors

Pen	Color
0	Erase
1	White
2	Red
3	Yellow
4	Green
5	Cyan
6	Blue
7	Magenta
8	Black
9	Olive Green
10	Aqua
11	Royal Blue
12	Maroon
13	Brick Red
14	Orange
15	Brown

Example Program

The following program demonstrates many of the techniques discussed so far. Running this program produces the "HELP" screen displayed in figure 7-2. The graphics lines appear in red on your analyzer's display.

```
10   DIM A$[58],String$[2000]
20   GINIT
30   CLEAR SCREEN
40   GCLEAR
50   OUTPUT 800;"DISP:PROG FULL"
60   OUTPUT 800;"DISP:PROG?"
70   ENTER 800;P$
80   GRAPHICS ON
90   PEN 2
100  FRAME
110  MOVE 0,91
120  DRAW 100*RATIO,91
130  PRINT TABXY(28,2);"HELP"
140  OUTPUT A$;" This program demonstrates how to print"
150  String$=String$&A$
160  OUTPUT A$;" several lines of text at one time.  This"
170  String$=String$&A$
180  OUTPUT A$;" method offers the fastest possible print speed."
190  String$=String$&A$
200  PRINTER IS CRT;WIDTH 2000      !prevent auto cr/lf
210  PRINT TABXY(1,4);String$
220  END
```

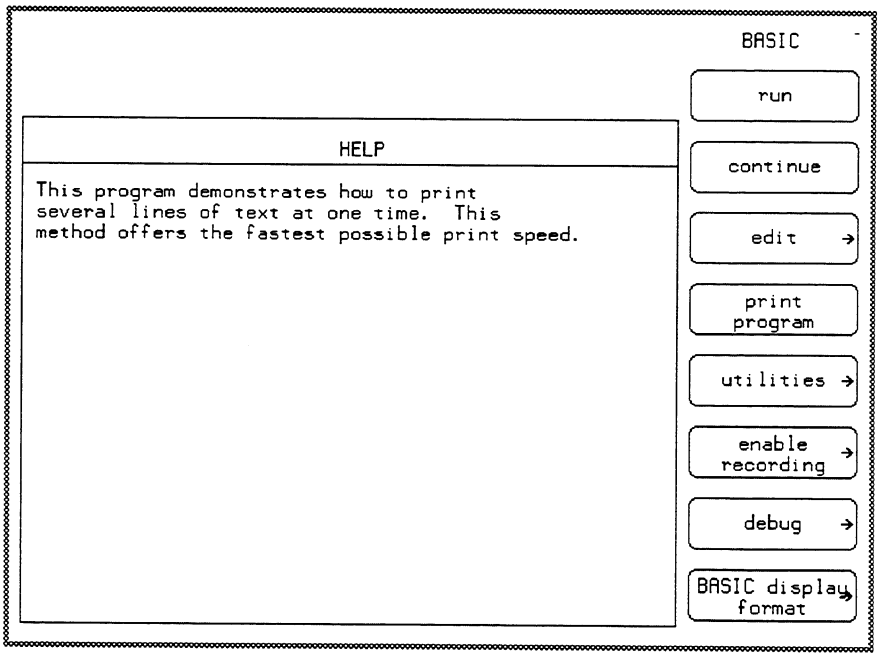

Figure 7-2. "HELP" Screen Output

Chapter 8

Interfacing with the HP-IB

8

Interfacing with the HP-IB

Introduction

This section describes the techniques necessary for programming the HP-IB interface. It also describes specific details of how this interface works and how to use it to control or interface with systems containing various HP-IB devices.

The HP-IB interface is Hewlett-Packard's implementation of the IEEE-488.1 Digital Interface for Programmable Instrumentation. The acronym HP-IB stands for "Hewlett-Packard Interface Bus," and is often referred to as the "bus."

The HP-IB Interface is both easy to use and allows great flexibility in communicating data and control information between the HP Instrument BASIC program and external devices.

HP Instrument BASIC is essentially an HP-IB instrument controller residing inside an instrument. It uses the host instrument's HP-IB interface for external communication and an internal HP-IB interface to communicate with the host instrument. This unique arrangement presents a few differences between HP Instrument BASIC's implementation of HP-IB control and the standard HP Series 200/300 BASIC Controller. A description of the interaction of HP Instrument BASIC with the host instrument and the external HP-IB interface is given in the section entitled "The HP Instrument BASIC HP-IB Model."

Communicating with Devices

This section describes programming techniques used to transfer data to and from HP-IB devices. General bus operation is described in a later section.

HP-IB Device Selectors

Since the HP-IB allows the interconnection of several devices, each device must have a means of being uniquely accessed. A device selector consists of two parts: the interface select code and the device's primary address. When a particular HP-IB device is to be accessed, it must be identified with both its interface select code and its bus address.

The interface select code is the first part of an HP-IB device selector. HP Instrument BASIC programs run inside a host instrument and communicate with it over the internal bus, which is addressed with select code 8. HP Instrument BASIC programs can also communicate with external devices via the host instrument's HP-IB interface. The external bus select code is 7.

The second part of an HP-IB device selector is the device's primary address. Each HP-IB device has a primary address which can be configured. The address can range from 0 to 30. For example, to specify the device on the interface at select code 7 (external bus) with a primary address of 22, use device selector = 722.

Each device's address must be unique. The procedure for setting the address of an HP-IB device is given in the installation manual for each device. Since the host instrument is the only device on the internal interface, its primary address on that interface is arbitrary and the instrument will respond to any primary address with a select code equal to 8XX (e.g., 800, 811, 822, etc.).

Secondary Addressing

Many devices have operating modes which are accessed through the extended addressing capabilities defined in the bus standard. Extended addressing provides for a second address parameter in addition to the primary address. Examples of statements that use extended addressing are as follows:

```
100  ASSIGN @Device TO 72205  !22=primary, 05=secondary
110  OUTPUT @Device;Message$

200  OUTPUT 72205;Message$

150  ASSIGN @Device TO 7220529       !Additional secondary
160                                  !address of 29
170  OUTPUT @Device;Message$

120  OUTPUT 7220529;Message$
```

The range of secondary addresses is 00-31. Up to six secondary addresses may be specified—a total of 15 digits including interface select code and primary address. Refer to the device's

operating manual for programming information associated with the extended addressing capability.

Moving Data Through the HP-IB

Data is sent from the program through the HP-IB with the OUTPUT statement. Data is entered into the program with the ENTER statement.

The following examples illustrate the use of HP-IB device selectors with OUTPUT and ENTER statements.

Examples

```
100   Hpib=7
110   Device_addr=22
120   Device_selector=Hpib * 100 + Device_addr
130   !
140   OUTPUT Device_selector;"SYST:ERR?"
150   ENTER Device_selector;Reading

320   ASSIGN @Hpib_device TO 702
330   OUTPUT @Hpib_device;"Data message"
340   ENTER @Hpib_device;Number

440   OUTPUT 800;"SOUR:FREQ 1 KHZ"

380   ENTER 724;Readings(*)
```

General Structure of the HP-IB

Communications through the HP-IB are made according to a precisely defined set of rules. These rules ensure that only orderly communication takes place on the bus.

For conceptual purposes, the organization of the HP-IB can be compared to that of a committee. A committee uses "rules of order" to govern the manner in which they conduct their business. For example, a committee may conduct their meetings using "Robert's Rules of Order." For the HP-IB, the rules of order are the IEEE 488.1 standard.

The HP-IB System Controller is analogous to the chairman of a committee. Only one device can be designated System Controller and it is designated *before* running a program. The System Controller cannot be changed while under the control of a HP Instrument BASIC program. However, as it is possible for a chairman to designate an "acting chairman" for the committee, so can control be passed to another device on the HP-IB. This device is called the Active Controller. It can be any device capable of directing HP-IB activities, such as an instrument (using printing and plotting functions) or a desktop computer.

When the System Controller is first turned on or reset, it assumes the role of Active Controller. These responsibilities may be subsequently passed to another device while the System Controller tends to other business. This ability to pass control allows more than one computer to be connected to the HP-IB at the same time.

In an effective committee, only one person may speak at a time. It is the responsibility of the chairman to "recognize" which member is to speak. Usually, all committee members present are expected to listen at all times; however, this is not always the case on the HP-IB. One of the most powerful features of the bus is the ability to selectively send data to individual (or groups of) devices. This allows fast talkers to communicate with fast listeners without having to wait for slower listeners on the bus.

During a committee meeting, the current chairman is responsible for telling the committee which member is to be the "talker" and which members are to be the "listeners." Before these assignments are given, she gets the attention of the members. The talker and listeners are designated and then the talker presents the data. The designation process may be repeated after the talker has completed his message.

On the HP-IB, the Active Controller takes similar action when a talker and the listener(s) are designated. The attention signal line (ATN) is asserted while the talker and listener(s) are being addressed. ATN is then cleared, signaling that those devices not addressed to listen may ignore all subsequent data messages. Thus, the ATN line separates data from commands. Commands are accompanied with the ATN line being true, while data messages are sent with the ATN line being false.

On the HP-IB, devices are addressed to talk and addressed to listen in an orderly manner. The Active Controller first sends a single command that causes all devices to stop listening. The talker's address is then sent, followed by the address(es) of the listener(s). After all listeners have been addressed, the data is sent from the talker to the listener(s). Only device(s) addressed to listen accept any data that is sent through the bus (until the bus is reconfigured by subsequent addressing commands).

The transfer of data, called a data message, exchanges information between devices on the HP-IB. A committee conducts business by exchanging ideas and information between the speaker and those listening to his presentation. On the HP-IB, data is transferred from the active talker to the active listener(s) *at a rate determined by the slowest active listener on the bus*. This restriction on the transfer rate is necessary to ensure that no data is lost by any device addressed to listen. The handshake used to transfer each data byte ensures that all data output by the talker is received by all active listeners.

Examples of Bus Sequences

With HP Instrument BASIC, all data transfers through the HP-IB involve a talker and only one listener.

The following example illustrates the sequence of commands which are generated by the Active Controller to send data to an HP-IB device through the bus with a simple OUTPUT statement.

```
OUTPUT 701;"DATA"
```

1. The unlisten command is sent.
2. The talker's address, which is also a command, is sent. In this case, the address of the active controller.
3. The listener's address (01), which is also a command, is sent.
4. The data bytes "D", "A", "T", "A", CR, and LF are sent; all bytes are sent using the HP-IB's interlocking handshake to ensure that the listener has received each byte.

Similarly, all ENTER statements involve transferring data from a talker to only one listener. For instance, the following ENTER statement invokes the following sequence of commands and data-transfer operations.

```
ENTER 722;Voltage
```

1. The unlisten command is sent.
2. The talker's address (22), which is a command, is sent.
3. The listener's address, also a command, is sent. In this case, the listener's address is the active controller's address.
4. The data is sent by device 22 to the controller using the HP-IB handshake.

General Bus Management

The HP-IB standard provides several mechanisms that allow managing the bus and the devices on the bus. The following is a summary of the statements that invoke these control mechanisms.

ABORT is used to abruptly terminate all bus activity and reset all devices to their power-on states.

CLEAR is used to set all (or only selected) devices to a pre-defined, device-dependent state.

LOCAL is used to return all (or selected) devices to local (front panel) control.

LOCAL LOCKOUT is used to disable all devices' front panel controls.

REMOTE is used to put all (or selected) devices into their device-dependent, remote modes.

SPOLL is used to perform a serial poll of the specified device which must be capable of responding.

TRIGGER is used to send the trigger message to a device (or selected group of devices).

These statements (and functions) are described in the following sections. However, the actions that a device takes upon receiving each of the above statements are generally different for each device. *For external devices, refer to the particular device's documentation to determine how it responds.*

All of the bus management statements, with the exception of ABORT, require that the HP Instrument BASIC program be the Active Controller on the interface. A running program is always the Active Controller on the internal interface (select code 8). For the program to be the active controller on the external interface (select code 7), the host instrument must either be set as the System Controller or have control passed to it from the external controller. The program automatically assumes the controller status of the host instrument. For additional information refer to "The HP Instrument BASIC HP-IB Model" section later in this chapter.

REMOTE

External Devices

Most HP-IB devices can be controlled either from the front panel or from the bus. The device is in the "Local" state if the front panel controls are currently functional. If the device is controlled through the HP-IB, it is in the Remote state. Pressing the [**Local/Setup**] key returns the device to Local (front panel) control; unless the device is in the "Local Lockout" state, or the device is the host instrument.

The Remote message is automatically sent to all devices whenever the System Controller is powered on, is reset, or sends the Abort message. A device enters the Remote state automatically whenever it is addressed. The REMOTE statement also sends the Remote message. This causes all (or specified) devices on the bus to change from local control to remote control. The host instrument must be set to System Controller before an HP Instrument BASIC program can execute the REMOTE statement on select code 7 (the external bus).

Examples

```
REMOTE 7

ASSIGN @Device TO 700
REMOTE @Device

REMOTE 700
```

Host Instrument

The REMOTE statement has no effect on the host instrument because it is *always* in remote control whenever an HP Instrument BASIC program is running. Specifying the internal interface in a REMOTE statement has no effect and does not generate an error.

LOCAL LOCKOUT

External Devices

The Local Lockout message effectively locks out the "local" switch present on most HP-IB device front panels. This prevents anyone from interfering with the device's system operations by pressing buttons. Local lockout maintains system integrity. As long as Local Lockout is in effect, no bus device can be returned to local control from its front panel.

The Local Lockout message is sent by executing the LOCAL LOCKOUT statement. This message is sent to all devices on the external interface.

Examples

```
ASSIGN @Hpib TO 7
LOCAL LOCKOUT @Hpib

LOCAL LOCKOUT 7
```

The Local Lockout message is cleared when the Local message is sent by executing the LOCAL statement. Executing the ABORT statement does *not* cancel the Local Lockout message.

Host Instrument

The Local Lockout message is not supported for the host instrument because some front panel functionality is always necessary in order to pause or to abort the program. Specifying the internal interface in a LOCAL LOCKOUT statement does not generate an error and has no effect.

LOCAL

External Devices

It is good systems practice to return all devices to local control upon conclusion of remote-control operations. For example, an operator might need to troubleshoot or to work from the front panel to make special tests. Executing the LOCAL statement returns the specified devices to local (front panel) control.

If primary addressing is specified, the Go-to-Local message is sent only to the specified device. However, if the interface select code alone is specified (LOCAL 7), the Local message is sent to *all* devices on the external interface. Any previous Local Lockout message which is still in effect is automatically cleared.

Examples

```
ASSIGN @Hpib TO 7
LOCAL @Hpib

ASSIGN @Device TO 700
LOCAL @Device
```

Host Instrument

The LOCAL statement has no effect on the host instrument because it is always in remote control whenever an HP Instrument BASIC program is running. Specifying the internal interface in a LOCAL statement does not generate an error.

TRIGGER

External HP-IB Devices

The TRIGGER statement sends a Trigger message to a selected device or group of devices. The purpose of the Trigger message is to initiate some device-dependent action; for example, it can be used to trigger a digital voltmeter to perform its measurement cycle. The response of a device to a Trigger Message is strictly device-dependent. Neither the Trigger message nor the interface indicates what action is initiated by the device.

Examples

```
ASSIGN @Hpib TO 7
TRIGGER @Hpib

ASSIGN @Device TO 707
TRIGGER @Device
```

Specifying only the interface select code sends a Trigger message to all devices currently addressed to listen on the bus. Specifying a device's primary address in the statement triggers only the device addressed by the statement.

Host Instrument

The TRIGGER statement is fully compatible on the internal HP-IB interface. HP 89400-series analyzers must be set to trigger on the HP-IB for this statement to be effective.

```
OUTPUT @HP89410a;"TRIG:SOUR BUS"
TRIGGER @Hp89410A
```

CLEAR

External HP-IB Devices

The CLEAR statement provides a means of "initializing" a device to its predefined, device-dependent state. When the CLEAR statement is executed, the Clear message is sent either to all devices or to the specified device, depending on the information contained within the device selector. If only the interface select code is specified, all devices on the specified HP-IB interface are cleared. If primary-address information is specified, the Clear message is sent only to the specified device. Only the Active Controller can send the Clear message.

Examples

```
ASSIGN @Hpib TO 7
CLEAR @Hpib

ASSIGN @Device TO 700
CLEAR @Device
```

Host Instrument

The CLEAR statement is fully compatible on the internal interface.

ABORT

External Devices

This statement terminates all activity on the external bus and returns all of the devices on the HP-IB to a reset (or power-on) condition. Whether this affects other modes of the device depends on the device itself. The HP Instrument BASIC program must be the Active Controller or the System Controller to perform this function. If it is the System Controller and has passed active control to another device, executing this statement returns active control to the program. *Only the interface select code is specified*; primary-addressing information (such as 724) is not included.

Examples

```
ASSIGN @Hpib TO 7
ABORT @Hpib

ABORT 7
```

Aborting the Internal Bus

ABORT is not supported for the internal bus, select code 8. Executing ABORT 8 does not generate an error.

HP-IB Service Requests

Most HP-IB devices, such as voltmeters, frequency counters, and spectrum analyzers, are capable of generating a "service request" when they require the Active Controller to take action. Service requests are generally made after the device has completed a task (such as making a measurement) or when an error condition exists (such as a printer being out of paper). The documentation, operating or programming manuals, for each device describes the device's capability to request service and the conditions in which the device requests service.

To request service, the device sends a Service Request message (SRQ) to the Active Controller. The mechanism by which the Active Controller detects these requests is the SRQ interrupt. Interrupts allow an efficient use of system resources, because the system executes a program until interrupted by an event's occurrence. If enabled, the external event initiates a program branch to a routine which "services" the event and executes remedial action.

Setting Up and Enabling SRQ Interrupts

In order for an HP-IB device to initiate a service routine in the Active Controller, two prerequisites must be met:

1. The SRQ interrupt event must have a defined service routine.
2. The SRQ interrupt must be enabled to initiate the branch to the service routine.

The following program segment shows an example of setting up and enabling an SRQ interrupt.

```
100  Hpib=7
110  ON INTR Hpib GOSUB Service_routine
120  !
130  Mask=2
140  ENABLE INTR Hpib;Mask
```

Since HP Instrument BASIC recognizes only SRQ interrupts, the value assigned to the mask is meaningless. However, a mask value may be present as a placeholder for compatibility with HP Series 200/300 BASIC programs.

When an SRQ interrupt is generated by any device on the bus, the program branches to the service routine when it exits the current line — either when the execution of the line is completed or when the line calls a user-defined function. The service routine must perform the following operations:

1. Determine which device is requesting service (parallel poll).
2. Determine what action is requested (serial poll).
3. Clear the SRQ line.
4. Perform the requested action.
5. Re-enable interrupts.
6. Return to the former task (if applicable).

Note

The ON INTR statement must always precede the ENABLE INTR statement when the two are used in the same program.

Servicing SRQ Interrupts

The SRQ is a level-sensitive interrupt; in other words, the interrupt may not be immediately detected when the SRQ line goes low. This implies that an interrupt may not be generated if the SRQ is present momentarily but does not remain long enough to be sensed by the controller. The level-sensitive nature of the SRQ line also has implications, which are described in the following example.

Example of a SRQ Interrupt

Assume only one device is currently on the bus. The following service routine first serially polls the device requesting service, thereby clearing the interrupt request. In this case, the controller did not have to determine which device was requesting service because only one device is on the bus. Also, the type of interrupt is not determined because only service request interrupts are enabled in HP Instrument BASIC. The service is then performed, and the SRQ event is re-enabled to generate subsequent interrupts.

```
500 Serv_rtn:  Ser_poll=SPOLL(@Device)
510    ENTER @Device;Value
520    PRINT Value
530    ENABLE INTR 7
540  RETURN
```

The IEEE standard specifies that when an interrupting device is serially polled, it is to stop interrupting until a new condition arises (or the same condition arises again). In order to "clear" the SRQ line, *it is necessary to perform a serial poll on the device*. This poll is an acknowledgement from the controller to the device that it has seen the request for service and is responding. The device then removes its request for service by releasing the SRQ line. When the SRQ line is released, the line goes high.

If the SRQ line had not been released, the controller would have immediately branched to the service routine after enabling interrupts on the external interface (line 530). This is another implication of the level-sensitive nature of the SRQ interrupt.

Once an interrupt is sensed and logged, the interface cannot generate another interrupt until after the initial interrupt is serviced. The controller disables all subsequent interrupts from an interface until a pending interrupt is serviced. For this reason, it is necessary to allow for subsequent branching.

Conducting a Serial Poll

A sequential poll of individual devices on the bus in known as a Serial Poll. The status of a specific device is returned in response to a Serial Poll. One entire byte is used. This is called the "Status Byte" message. Depending on the device, the Status Byte may indicate an overload condition, a request for service, or a printer which is out of paper. The particular response of each device depends on the device.

The SPOLL function performs a Serial Poll of the specified device. The HP Instrument BASIC program *must be* the Active Controller in order to execute it.

Examples

```
ASSIGN @Device TO 700
Status_byte=SPOLL(@Device)

Spoll_724=SPOLL(724)
```

The Serial Poll is meaningless for the external bus since it must poll the individual devices on the bus. Therefore, primary addressing must be used with the SPOLL function.

Passing and Regaining Control

Passing control can be accomplished in one of two ways: it can be handled by the system, or it can be handled by the program. To handle it programmatically, use the PASS CONTROL statement. HP Instrument BASIC or the analyzer can control the external bus (select code 7). The following statements first define the HP-IB's select code, specify the new Active Controller's primary address and then pass control to that controller.

```
100  Hp_ib=7
110  New_ac_addr=20
120  PASS CONTROL 100*Hp_ib+New_ac_addr
```

Once the new Active Controller has accepted active control, the controller which passed control assumes the role of a non-Active Controller on the HP-IB. *HP Instrument BASIC programs cannot act as a device when in a role of non-Active controller.*

Active control of the internal HP-IB bus (select code 8) cannot be passed. The statement "PASS CONTROL 800" passes control of the external bus to the instrument. This is required whenever the analyzer performs a plot operation to a peripheral on the bus or the analyzer accesses an external disk drive. These concepts are discussed next in "The HP Instrument BASIC HP-IB Model."

The HP Instrument BASIC HP-IB Model

The fact that HP Instrument BASIC resides in, and co-exists with an instrument creates a large set of possible interactions, both internally within the instrument as well as externally with other controllers and instruments. This section defines the principal players and rules of order when HP Instrument BASIC executes within the host instrument.

External and Internal Busses

There is physically only one HP-IB port and one HP-IB address for the HP 89400-series analyzers. HP Instrument BASIC has access to two HP-IB ports: the "real" external port (select code 7) and a "virtual" internal port (select code 8), through which it communicates with the HP 89400-series analyzers. See figure 8-1.

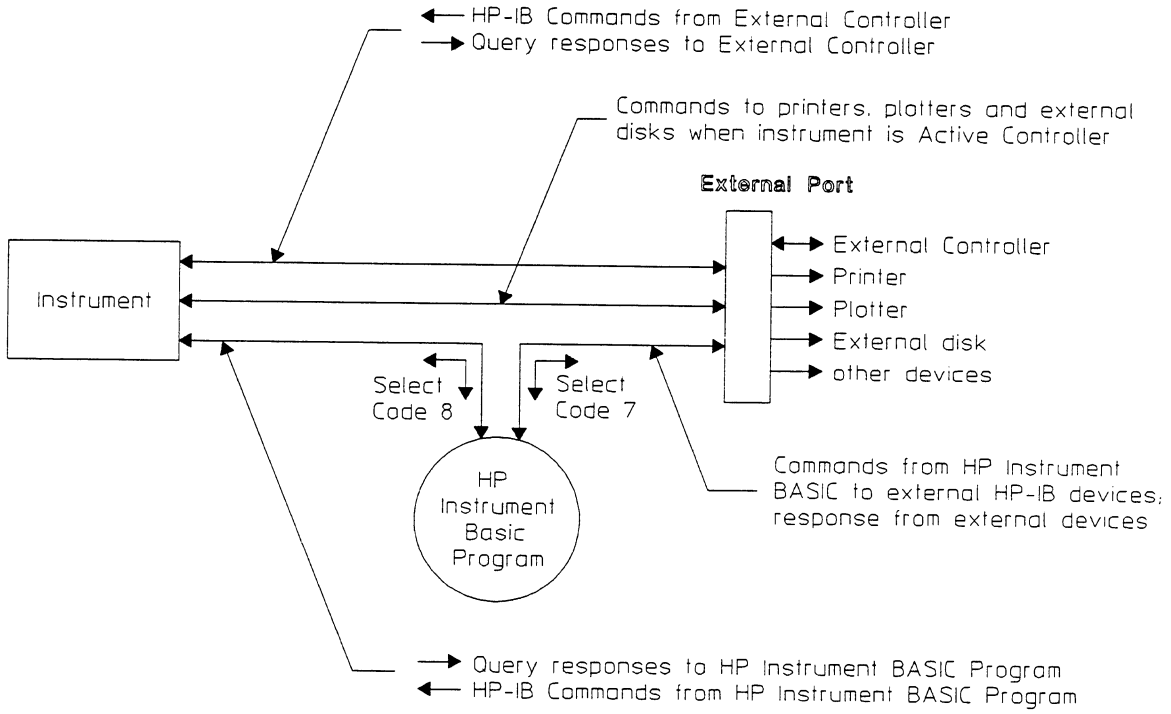

Figure 8-1. HP 89400 Series External and Internal Port

Service Request Indicators

An external controller may perform a serial poll (SPOLL) at any time without affecting a running HP Instrument BASIC program. There are two Service Request Indicators (SRI) – one for the external port and one for the internal port. The internal SRI can only be cleared by an HP Instrument BASIC program performing an SPOLL on device 800. The external SRI can only be cleared by an SPOLL from an external controller and can only be set when there is no active HP Instrument BASIC program.

The two SRI's are set to their OR'd value when a program starts, and again when it finishes. This assures that any pending SRQ's can be serviced by the instrument's new controller.

The pausing or termination of a program causes the PROGRAM_RUNNING bit in the Operation Status register to go low. This can be used to generate an external SRQ. (For an example, see the example program, TWO_CTLR, in chapter 9.)

Status Registers

The HP 89400-series analyzers' status registers contain information about various analyzer conditions. There are eight register sets. Their reporting structure is summarized in figure 8-2.

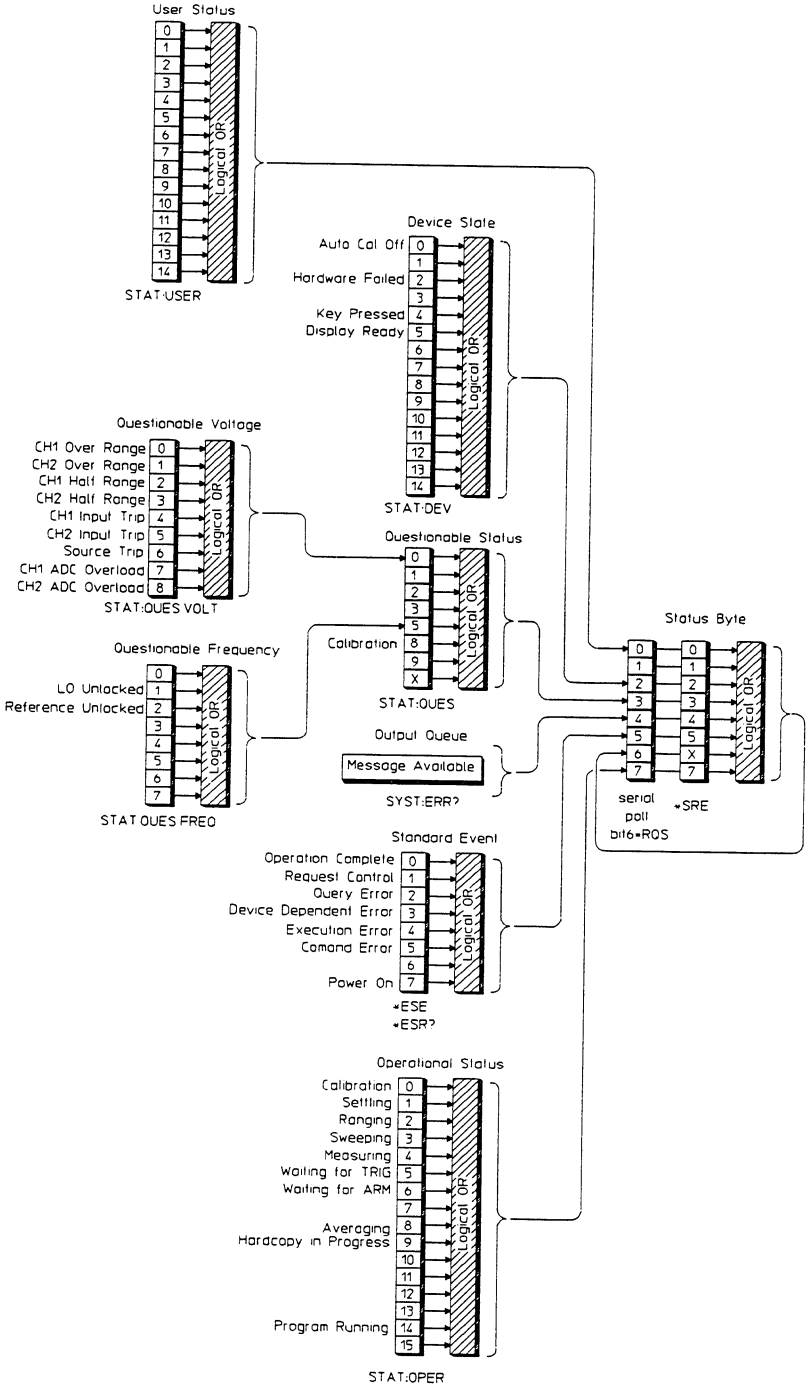

Figure 8-2. HP 89400 Series Status Registers

For more detailed information about the analyzer's register sets, refer to the *HP-IB Command Reference* shipped with your HP 89400-series analyzer.

HP Instrument BASIC as the Active Controller

The HP Instrument BASIC program is *always* the Active Controller on the internal bus (select code 8). When a program starts running, the HP-IB controller status of the instrument is automatically passed to the program. See figure 8-3. For example, if the instrument is set as System Controller, a program running in the instrument automatically becomes the System Controller and the Active Controller on the external bus and the instrument relinquishes active control. When the program stops, the instrument regains active control.

Similarly, if an instrument set as Addressable Only is passed control from an external controller, any HP Instrument BASIC program running in the instrument becomes active controller on the external interface.

There are two cases when a program running in an instrument can become the Active Controller on the external interface:

- When the host instrument is set as System Controller and the program has *not* passed control.
- When the host instrument is set as Addressable Only and the instrument has been passed control from an external controller.

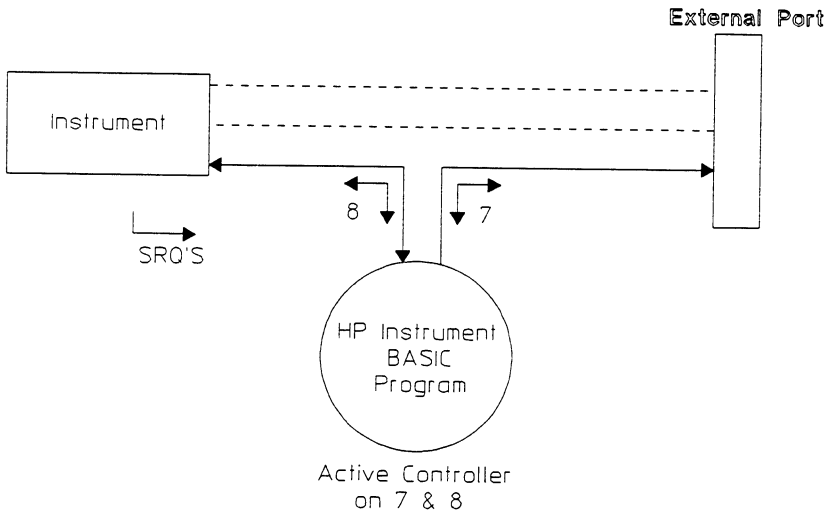

Figure 8-3. The Program as Active Controller on the External Interface

Passing Active Control to the Instrument

The only way that the HP 89400-series analyzers can gain active control of the external interface while an
HP Instrument BASIC program is running is if the program is currently the Active Controller on select code 7 and passes control to the instrument. Normally, the active controller on the external bus can pass control to any device on the interface by using the statement

```
PASS CONTROL 7xx
```

where "xx" represents the primary address of the device on the bus. However, since an HP Instrument BASIC program does not interface with the host instrument via select code 7, a different method must be used to pass control. To pass active control of the external interface from an HP Instrument BASIC program to the host instrument, use the statement:

```
PASS CONTROL 8xx
```

where "xx" represents any two digit number from 00 to 99. This allows the instrument to control external plotters, printers and disk drives. See figure 8-4. When the instrument is finished with its HP-IB control activity, it automatically passes control back to the program. If the instrument is waiting for control and the HP Instrument BASIC program terminates, control is implicitly passed back to the instrument. See figure 8-5.

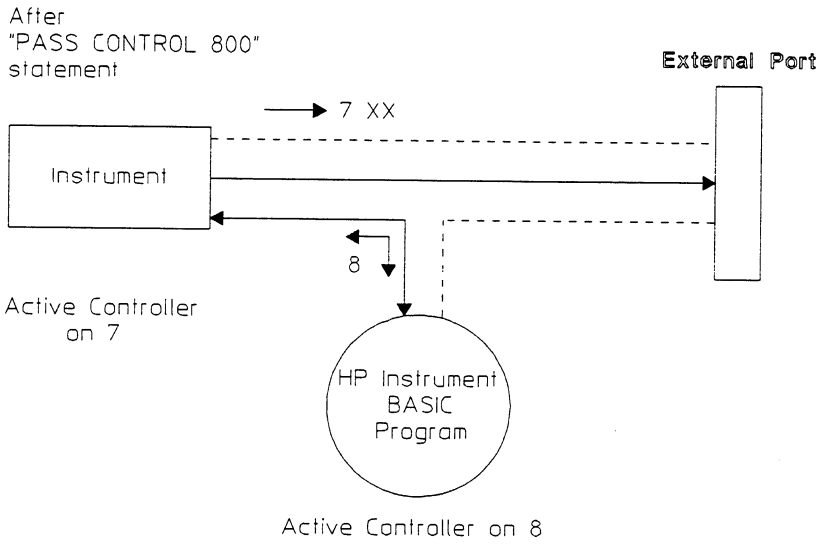

Figure 8-4. Passing Control of the External Interface to the HP 89400-series Analyzer

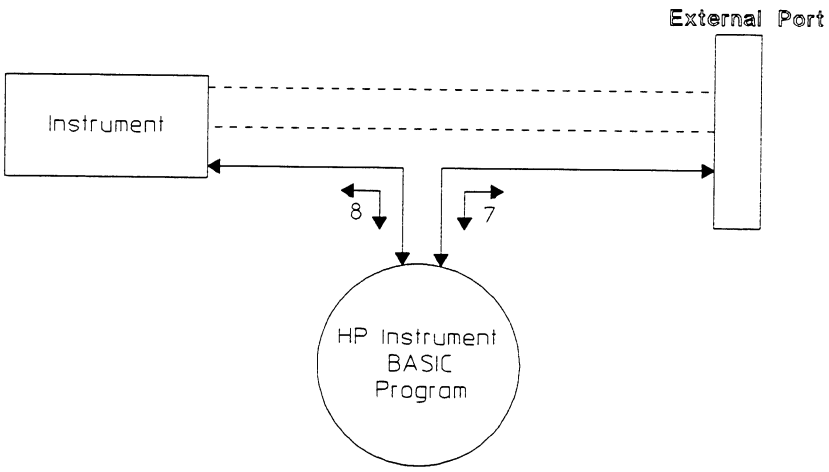

Figure 8-5. Control Passed Back to Program When Instrument Is Done

Note

Control of the internal bus is used to govern access to the external bus. When the instrument is given control of the internal bus, it actually gains access to the external HP-IB hardware.

HP Instrument BASIC as a Non-Active Controller

HP Instrument BASIC programs are always the Active Controller on the internal interface. There are two cases when an HP Instrument BASIC program does not have control of the external HP-IB interface:

- When the host instrument is set as Addressable Only and active control has *not* been passed from an external device.
- When the host instrument is set as System Controller and the program has passed control to either the host instrument or to another device on the external interface.

In both of these cases, the HP Instrument BASIC program cannot perform activities of any kind on the external bus. See figure 8-6.

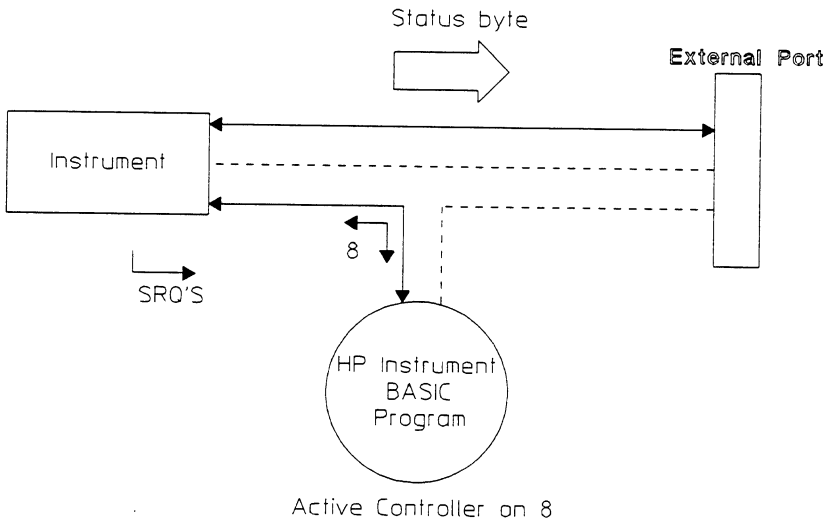

Figure 8-6. The Program as Non-Active Controller

Note

An HP Instrument BASIC program cannot act as a device on the external bus. To communicate with an external controller, the HP Instrument BASIC program must be Active Controller and the external controller must act as the device (see "Interfacing with an External Controller").

Interfacing with an External Controller

So far, we have limited our discussion to the ability to interface HP Instrument BASIC programs via HP-IB with a network of external devices. It is possible to include a computer in the network, and to interface an HP Instrument BASIC program with another program running in that computer.

External controller programs can interface with HP Instrument BASIC programs (hereafter referred to as "internal programs") over HP-IB in two ways:

The two programs can pass data back and forth using simple OUTPUT and ENTER statements. This requires coordination of both the internal and external programs and also requires that the internal program be the Active Controller during the interaction. To get an internal program and an external program to work together successfully, you should have a good understanding of the HP-IB model, as presented earlier in this chapter.

The external program can make use of the extensive set of HP 89400-series analyzers HP-IB commands that interface with HP Instrument BASIC programs. These commands fall under the subsystems PROGram and MMEMory, and allow the external controller to remotely perform many of the HP Instrument BASIC front panel activities. This includes the ability to run, stop, pause, continue, get, save or delete an internal program. You can also remotely set a program's memory size and query or set the values of numeric and string variables.

Commands that allow you to transfer programs and program data to and from the instrument are included in the HP 89400-series analyzers' HP-IB command set. Programs can be transferred (uploaded and downloaded) between an external controller and the program buffer in the instrument. Data can be transferred between an external program and internal program by setting and querying internal program variables. These commands are described in detail in *HP-IB Command Reference* shipped with your HP 89400-series analyzer.

Transferring Data Between Programs

Using OUTPUT and ENTER Statements

All data sent from an external controller to the instrument's external port is received by the instrument — not by any program running in it. Therefore, an HP Instrument BASIC program that is not the Active Controller cannot enter or output data via the external interface bus. In order to pass data between an external controller and an internal program using OUTPUT and ENTER statements, *the internal program must be given active control* and the external controller must become the non-Active Controller. All HP Series 200/300 BASIC controllers have the ability to enter and output data via HP-IB while acting as a non-Active Controller.

Note

Moving data through the HP-IB and running a measurement in the host instrument at the same time can slow both operations significantly. It is recommended that you do not perform these operations concurrently.

One method of passing data between the two controllers is to first set the instrument as Addressable Only. Next, run an HP Series

200/300 BASIC program that starts the HP Instrument BASIC program and then passes control to it. Thereafter, the HP Instrument BASIC program can output data to, and enter data from, the external controller. The following two programs, found on the
Example Programs disk, demonstrate how to transfer data between an internal program and an external controller program.

The first program, DTXFRB, runs on an HP Series 200/300 workstation. It assumes that a disk containing the corresponding HP Instrument BASIC program DTXFRA is in the analyzer's disk drive. It remotely loads the HP Instrument BASIC program, starts it and then transfers active control to it. The HP Instrument BASIC program DTXFRA, with active control of the interface, queries the external program for the name of the drive to catalog, and then sends the cataloged string to the external program and passes back active control. After receiving the catalog data, the external program goes into a loop (line 460). This command continues to generate an error until control is passed back to the host computer, which again becomes the active controller.

```
10  !BASIC program:  DTXFRB -- Data transfer BASIC to BASIC
20  !-----------------------------------------------------------------
30  ! This program demonstrates how to transfer data from an HP Instrument
40  ! BASIC program.  This program, which runs on the computer, loads a
50  ! a program into the HP89410A, runs it, and then gives it control of
60  ! the bus. This program then acts as a device on the bus; sending and
70  ! receiving data. Before running this program, a disk with the program
80  ! 'DTXFRA' should be in the HP89410A's internal drive.
90  !-----------------------------------------------------------------
100 Scode=7                         !Select code for interface
110 Address=11                      !Address for HP89410A
120 Hp89410a=Scode*100+Address
130 CLEAR Hp89410a
140 OUTPUT Hp89410a;"*CLS"          !Clear the EVENT registers
150 CLEAR SCREEN                    !Clear the display
160 !
170 DIM Directory$(1:100)[85]       !Array to hold catalog listing
180 !
190 INPUT "Put disk with program 'DTXFRA' into the HP89410A. Press ",A$
200 DISP "Loading program on HP89410A..."
210 OUTPUT Hp89410a;"MMEM:LOAD:PROG 'INT:DTXFRA'" !Load BASIC program from disk
220 OUTPUT Hp89410a;"*OPC?"
230 ENTER Hp89410a;Opc              !Wait here until program loaded
240 OUTPUT Hp89410a;"*ESR?"         !Read the EVENT STATUS reg
250 ENTER Hp89410a;Esr
260 IF Esr0 THEN                    !Have any errors occurred
270   BEEP
280   DISP "Error occurred while loading 'DTXFRA'...Cannot continue program."
290   STOP
300 END IF
310 !
320 OUTPUT Hp89410a;"*PCB 21"       !Set pass control back address
330                                 !to HP-IB address for controller
340 DISP "Running the program..."
350 OUTPUT Hp89410a;"PROG:STAT RUN"!Start the program
360 PASS CONTROL Hp89410a           !Give program control of bus
370 !
380 OUTPUT Scode;":INTERNAL"       !Wait until addressed to talk
390 DISP "Reading data..."
400 ENTER Scode;Directory$(*)       !Wait until addressed to listen
410 !
420 FOR I=1 TO 100                  !Print the catalog
430   IF LEN(Directory$(I))0 THEN PRINT Directory$(I)
440 NEXT I
450 !
460 ON ERROR GOTO 470               Loop until control passed back
470 LOCAL Hp89410a
480 DISP ""
490 END
```

```
10  !BASIC program:  DTXFRA -- Data transfer BASIC to BASIC
15 !-------------------------------------------------------------------
20 ! This program demonstrates how to transfer data to and from an
30 ! external controller.  In this example a catalog listing is transferred
40 ! from the HP89410A to the external controller.  For more information
50 ! look at the program listing for 'DTXFRB'
60 !
70 ! This program is intended to be executed with HP Instrument BASIC.
80 !-------------------------------------------------------------------
90  DIM Directory$(1:100)[85]          !Create string array for catalog
100 !
110  Host=721                          !Address for external controller
120 !
130  ON ERROR GOTO 140                 !Loop until control is passed to the HP89410A
140  ENTER Host;Stor_dev$              !Address Host to talk, read device to catalog
150  OFF ERROR
160 !
170  DISP "Reading catalog..."
180  CAT Stor_dev$ TO Directory$(*) !Catalog into the string array
190 !
200  DISP "Transferring data..."
210  OUTPUT Host;Directory$(*)         !Address Host to listen, write array
220 !
230  PASS CONTROL Host                 !Pass control back to host
240  DISP "DONE"
250  END
```

Setting and Querying Variables

Another means of transferring data between an internal and an external program involves the ability to set and query internal program variables from an external program. The "PROGram:NUMBer" and "PROGram:STRing" statements (and their query counterparts) are part of the HP 89400-series HP-IB commands.

The command

```
PROG:NUMBer <"label">,<numeric value>
```

sets the value of a numeric variable in the program. The command

```
PROG:STRing <"label$">,<"string value">
```

sets the value of a string variable in the program. In both the PROG:NUMB and PROG:STR commands and queries, the *label* must be a string in quotes. In the PROG:STRing command, the *string variable data* must also be in quotes.

Numeric and string parameters can also be queried. The query

```
PROG:NUMBer? <"label">
```

returns the value of the specified INTEGER or REAL variable. If you precede this HP-IB command with the FORMat ASCII command (for example, OUTPUT 719;"FORM ASCii,5") the number returns as a readable ASCII number.

The query

```
PROG:STRing? <"label$">
```

returns the value of the specified string variable.

Arrays of REAL or INTEGER types may be sent or queried, but arrays of strings are not allowed. Array elements are separated by commas.

Examples

```
OUTPUT 711;"PROG:NUMB 'Test',99"

OUTPUT @Ibasic;"PROG:STRING 'A$','String Data'"

OUTPUT 711;"PROG:NUMB? 'Iarray(*)'"
```

The following program segment sends both numeric and a string variable queries and enters the resulting data:

```
10  ASSIGN @Prog TO 711
20  OUTPUT @Prog;"FORM ASCII,3"
30  OUTPUT @Prog;"PROG:NUMB? 'Test'"
40  ENTER @Prog; Testval
50  PRINT "The value of the variable Test =  ";Testval
60  OUTPUT @Prog;"PROG:STR? 'A$'"
70  ENTER @Prog; Str$
80  PRINT "A$ = ";Str$
90  END
```

Downloading and Uploading Programs

Programs can be transferred between an external controller and program memory using the HP-IB download command "PROG:DEFine" and its converse upload query "PROG:DEFine?". Programs that use these commands are executed in the external controller.

Downloading

Program data transferred (downloaded) from the external controller to the instrument is always transferred as an "arbitrary block." The arbitrary block may be a definite length block or an indefinite length block. The indefinite length block is by far the easiest to transfer. It is simply a block of data that begins with the characters "#0" preceding the first line and ends with a line-feed character accompanied by an EOI signal on the HP-IB interface.

When using the HP-IB command "PROG:DEF" to download program lines, the "#0" *should not be followed by a line-feed.* Each program line then requires a line number at its beginning and a line-feed at its end. To end the arbitrary block of program lines, a single line-feed must be output with the OUTPUT END parameter, which sends the EOI (End or Identify) signal on the HP-IB control lines.

The following program runs on an external HP Series 200/300 workstation. It demonstrates downloading a short program into the program buffer of the instrument. It is included in the HP 89410A Example Programs disk. The file must be an ASCII file.

```
10 ! ----------------------------------------------------------
20 ! HP BASIC Program: DOWNLOAD
30 ! This program downloads a file into an HP 89410A Instrument
40 ! BASIC program from an external controller.
50 ! The downloaded program must be an HP BASIC ASCII type file.
60 ! It will NOT work with DOS or HP-UX (untyped) files.
70 ! ----------------------------------------------------------
80 !
90  DIM Load_file$[20],Prog_line$[256],Command$[80],Name$[10]
100 DIM Diskname$[20],Answer$[2]
110 ASSIGN @Hp89410a TO 711
120 !
130 ! The file of program to download must be an ASCII file.
140 !
150 INPUT "ENTER NAME OF FILE TO DOWNLOAD ",Load_file$
160 ASSIGN @File TO Load_file$
170 OUTPUT @Hp89410a;"PROG:DEL"
180 ON ERROR GOTO End_load
190 OUTPUT @Hp89410a;"PROG:DEF #0";
200 LOOP
210  ENTER @File;Prog_line$
220  PRINT Prog_line$
230  OUTPUT @Hp89410a;Prog_line$
240 END LOOP
250 !
260 End_load: !
270 OUTPUT @Hp89410a;CHR$(10) END
280 INPUT "SAVE PROGRAM TO INSTRUMENT'S DEFAULT DRIVE? [Y/N]",Answer$
290 IF UPC$(Answer$)="Y" THEN
300  INPUT "ENTER NAME FOR DISK FILE",Diskname$
310  OUTPUT @Hp89410a;"MMEM:STORE:PROGRAM '"&Diskname$&"'"
320 END IF
330 END
```

The OUTPUT statement on line 210 is terminated with a semicolon to suppress the line-feed that would otherwise occur.

As each line of the program is downloaded it is checked for syntax. If an error is found, the error message is displayed in a pop-up message window and the line is commented and checked for syntax again. If it still causes an error (for example the line may be too long) the line is discarded.

Any lines that currently exist in the memory buffer remain unless they are overwritten by downloaded program lines. This makes it easy to edit lines in an external controller and then download only the edited lines into an existing program. If you want to completely overwrite the current program in memory, you must delete the program first. This can be done remotely using the "PROG:DEL" command.

Uploading

The command "PROG:DEF?" is used to upload a program from the program buffer. The entire program is then returned as an definite length arbitrary block. A definite length block starts with the " #" character followed by a single digit defining the number of following digits to read as the block length. The following program demonstrates an uploading routine executed on an external controller. It is included in the Example Programs disk.

```
10 ! HP BASIC example program : UPLOAD
20 ! ----------
30 ! This program runs on an HP BASIC workstation connected to
40 ! the HP 89410A with HP Instrument BASIC installed. The 89410A
50 ! must have its address set to 711 and must be set up as
60 ! ADDRESSABLE ONLY on the HP-IB. This program uploads the
70 ! current program in the HP 89410A's memory to an ASCII file
80 ! on the workstation's current MSI disk.
90 ! ----------
100  ASSIGN @Hp89410a TO 711
110  DIM Prog_line$[256]
120  INPUT "ENTER NAME OF FILE INTO WHICH TO UPLOAD PROGRAM ",Filename$
130  PRINT Filename$
140  CLEAR @Hp89410a
150  OUTPUT @Hp89410a;"PROG:DEF?"
160  ENTER @Hp89410a USING "#,A,D";Prog_line$,Ndigits
170  ENTER @Hp89410a USING "#,"&VAL$(Ndigits)&"D";Nbytes
180  PRINT Nbytes
190  Openfile(@File,Filename$,Nbytes)
200  ASSIGN @File TO Filename$
210  LOOP
220   ENTER @Hp89410a;Prog_line$
230  EXIT IF LEN(Prog_line$)=0
240   PRINT Prog_line$
250   OUTPUT @File;Prog_line$
260  END LOOP
270  ASSIGN @File TO *
290  END
300  SUB Openfile(@File,Filename$,Fisize)
310   ON ERROR GOTO Openerr
320   IF Fisize MOD 256 > 0 THEN Fisize=Fisize+256
330   CREATE ASCII Filename$,Fisize DIV 256
340 Openerr: !
350   IF ERRN  < > 54 THEN
360    PRINT ERRM$
370   END IF
380  SUBEND
```

The subroutine, Openfile, (lines 300 through 330) creates a LIF file in which to save the uploaded program. The number of 256 byte records declared in the CREATE ASCII statement (line 330) is simply the file size (declared in the definite block header) divided by 256. Line 320 accommodates any remainder in this calculation by increasing the file size number by one record if any remainder exists.

Although this simple method works for many uploaded programs, there may still be a problem with the file size caused by the OUTPUT statement in line 250. This is because every ASCII line in a LIF file contains a two byte length header and possibly one additional pad byte to make the length an even number of bytes. These extra bytes are not included in the definite length block header information. You can account for this extra overhead by allocating an extra 10 to 15 percent of space when you create the ASCII file. For example, the Openfile subroutine could be rewritten as:

```
300  SUB Openfile(@File,Filename$,Fisize)
310      ON ERROR GOTO Openerr
315      Fisize = Fisize + (Fisize * .15)
320      IF Fisize MOD 256 > 0 THEN Fisize=Fisize+256
330      CREATE ASCII Filename$,Fisize DIV 256
```

Chapter 9

Example Programs

9

Example Programs

The *IBASIC Example Programs* disk, shipped with this manual, includes sample programs to help you learn and use IBASIC. This chapter contains listings of some of the programs on the *IBASIC Example Programs* disk. The programs included in this chapter are described below. You can load and run the program "READ_ME" for descriptions of other programs on the disk.

ARBSOURC Demonstrates programming the arbitrary source and transferring trace data to the data registers. The arbitrary source uses one of three waveforms, which have been downloaded to the data register. An existing trace is copied to the data register, uploaded, shifted left and right and then reloaded into the arbitrary source data register.

OPC_SYNC Demonstrates synchronizing the program and the analyzer using the *OPC statement to set the OPERATION_COMPLETE bit in the Standard Event Status Register. The Status Register is masked to generate an SRQ when all pending operations have completed. The program handles the service request interrupt.

OPCQSYNC Demonstrates synchronizing the program and the analyzer using the *OPC? query. The programs pauses on an ENTER statement while it waits for the pending HP-IB operations to complete and for a "1" to be returned in response to the *OPC? query.

SNGL_SWP Demonstrates using the Standard Event Register to detect a SWEEPING event and generates an SRQ. The Program handles the SRQ interrupt and allows you initiate the next single sweep.

TRC_LOAD Demonstrates downloading and uploading trace data between a program and the analyzer.

TWO_CTLR Demonstrates using an external controller to download an HP Instrument BASIC program, run it, and query variables.

WAI_SYNC Demonstrates synchronizing the program and the analyzer using the *WAI statement.

ARBSOURC

```
10   ! HP Instrument BASIC example program: ARBSOURC
20   !-----------------------------------------------
30   ! This program is used with the HP 89410a
40   ! Vector Signal Analyzer.  It allows a user
50   ! to create ramp, triangle, and square waves
60   ! and download them to an analyzer data register
70   ! to be used by the arbitrary source.
80   ! It also allows captured traces to be uploaded
90   ! to the program, shifted left or right, and then
100   ! downloaded into a data register.
110   !-----------------------------------------------
120  COM /Traces/ Trace_in(1:1024),Trace_out(1:1024)
130  COM /Assigns/ @Format_off,@Format_on,@Hp89410a
140  INTEGER Byte_count,Block_count
150   !
160  Sc=8
170  Addr=0
180  Device=(Sc*100)+Addr
190   !
200  ASSIGN @Format_on TO Device;FORMAT ON
210  ASSIGN @Format_off TO Device;FORMAT OFF
220  ASSIGN @Hp89410a TO Device
230   !
240   ! Set data xfer to binary
250  OUTPUT @Hp89410a;"FORM:DATA REAL,64"
260  Frequency=1024
270  High=2.5        ! high limit for arbitrary source
280  Low=-2.5        ! low limit for arbitrary source
290   !
300   ! Check if measurement memory configuration is correct.
310   !
320  OUTPUT @Hp89410a;"MEM:MALL:MEAS:FPO?"
330  ENTER @Hp89410a;Block_count
340  IF Block_count<401 THEN
350   ! Try to upsize maximum # of frequency points
360    OUTPUT @Hp89410a;"MEM:MALL:MEAS:FPO 401"
370    OUTPUT @Hp89410a;"MEM:MALL:MEAS:FPO?"
380    ENTER @Hp89410a;Block_count
390    IF Block_count<>401 THEN
400      DISP "Need <max freq pts> >= 401 to RUN"
410      BEEP
420      STOP
430    END IF
440  END IF
450   !
460   !-----------------------------------------------
470   !  Setup display, and arbitrary source
480   !-----------------------------------------------
490  OUTPUT @Hp89410a;"ROUT:REC IF"
500  OUTPUT @Hp89410a;"SYST:PRES"
```

```
510  OUTPUT @Hp89410a;"BAND:MODE:ARB ON"
520  OUTPUT @Hp89410a;"DISP:FORM TWO"
530  OUTPUT @Hp89410a;"CALC1:FEED 'XTIM:VOLT 1'; *WAI"
540  OUTPUT @Hp89410a;"CALC2:FEED 'D1'; *WAI"
550  OUTPUT @Hp89410a;"ABOR;:INIT; *WAI"
560  OUTPUT @Hp89410a;"INIT:CONT OFF"
570  Running=False
580  OUTPUT @Hp89410a;"TRAC:COPY D1,TRAC1; *WAI"
590  OUTPUT @Hp89410a;"SOUR:FUNC:USER:FEED 'D1'"
600  OUTPUT @Hp89410a;"SOUR:FUNC USER"
610  OUTPUT @Hp89410a;"SOUR:VOLT 1.0 Vpk"
620  OUTPUT @Hp89410a;"OUTP ON"
630  OUTPUT @Hp89410a;"CAL:AUTO OFF"
640  OUTPUT @Hp89410a;"VOLT1:RANG:AUTO:DIR EITH;:VOLT1:RANG:AUTO ON"
650  !
660  DISP "Connect SOURCE to CHANNEL 1 and choose waveform"
670  !
680  !-----------------------------------------------
690  ! Setup menu
700  !-----------------------------------------------
710  ON KEY 0 LABEL "Ramp" GOSUB Ramp
720  ON KEY 1 LABEL "Triangle" GOSUB Triangle
730  ON KEY 2 LABEL "Square" GOSUB Square
740  ON KEY 3 LABEL "Copy A->D1" GOSUB Storetrace
750  ON KEY 4 LABEL "Load D1 " GOSUB Data_in
760  ON KEY 5 LABEL "Shift trace" CALL Shift_trace
770  ON KEY 6 LABEL "Pause/continue measurement" GOSUB Pause_cont
780  ON KEY 7 LABEL "Exit" GOTO Endit
790  !
800 Waiting:  !
810  WAIT .1
820  GOTO Waiting
830  !
840  !-----------------------------------------------
850 Ramp:  ! create ramp wave in D1
860  GOSUB Enterfreq
870  DISP "Calculating ramp wave"
880  N=1
890  FOR I=1 TO 1024/Period
900   Current=Low
910   FOR J=1 TO Period
920    Trace_out(N)=Current
930    N=N+1
940    Current=Current+(High-Low)/Period
950   NEXT J
960  NEXT I
970  !
980  CALL Data_out  ! send Trace_out to D1
990  DISP ""
1000 RETURN
1010 !
1020 !-----------------------------------------------
1030 Triangle:  ! create triangle wave in D1
```

```
1040  GOSUB Enterfreq
1050  DISP "Calculating triangle wave"
1060  N=1
1070  FOR I=1 TO 1024/Period
1080   Current=Low
1090   FOR J=1 TO Period/2
1100    Trace_out(N)=Current
1110    N=N+1
1120    Current=Current+(High-Low)/Period
1130   NEXT J
1140   Current=Current-(High-Low)/Period
1150   FOR J=1 TO Period/2
1160    Current=Current-(High-Low)/Period
1170    Trace_out(N)=Current
1180    N=N+1
1190   NEXT J
1200  NEXT I
1210  CALL Data_out  ! send Trace_out to D1
1220  DISP ""
1230  RETURN
1240  !
1250  !-------------------------------------------------
1260 Square:  ! create square wave in Trace_out array
1270  GOSUB Enterfreq
1280  DISP "Calculating square wave"
1290  Toggle=1
1300  N=1
1310  Current=Low
1320  FOR I=1 TO 1024/Period
1330   FOR J=1 TO Period
1340    Trace_out(N)=Current
1350    N=N+1
1360   NEXT J
1370   IF Toggle THEN Current=High
1380   IF NOT Toggle THEN Current=Low
1390   Toggle=NOT Toggle
1400  NEXT I
1410  CALL Data_out  ! send Trace_out to D1
1420  DISP ""
1430  RETURN
1440  !
1450  !-------------------------------------------------
1460 Storetrace:  ! Copy Trace 1 to D1
1470  OUTPUT @Hp89410a;"TRAC:COPY D1,TRAC1"
1480  RETURN
1490  !
1500  ! Enter frequency, which must be power of two so
1510  ! that waveform is symmetrical in 1024 point block
1520  !-------------------------------------------------
1530 Enterfreq:  !
1540  INPUT "Enter frequency (as power of 2 >= 128): ",Frequency
1550  IF Frequency>65536 THEN Frequency=65536
1560  Period=128/Frequency*1024
```

```
1570 RETURN
1580 !
1590 !-----------------------------------------------
1600 Data_in: ! Read D1 trace data into Trace_in array
1610 CLEAR @Hp89410a
1620 OUTPUT @Hp89410a;"FORM:DATA REAL,64"
1630 OUTPUT @Format_on;"TRAC:DATA? D1"
1640 ENTER @Format_on USING "%,A,D";A$,Byte_count
1650 ENTER @Format_on USING "%,4D";Block_count
1660 ENTER @Format_off;Trace_in(*)
1670 ENTER @Format_on;A$
1680 RETURN
1690 !
1700 !-----------------------------------------------
1710 Pause_cont: ! Pause or continues measurement
1720 IF Running THEN
1730  OUTPUT @Hp89410a;"INIT:CONT OFF"
1740 ELSE
1750  OUTPUT @Hp89410a;"INIT:CONT ON"
1760 END IF
1770 Running=NOT Running
1780 RETURN
1790 !
1800 Endit:END
1810 !
1820 SUB Data_out
1830 !-----------------------------------------------
1840 ! Sends trace data from Trace_out array to D1
1850 ! data register in HP 89410a and sets source
1860 ! output as arbitrary source
1870 !-----------------------------------------------
1880  COM /Traces/ Trace_in(1:1024),Trace_out(1:1024)
1890  COM /Assigns/ @Format_off,@Format_on,@Hp89410a
1900  OUTPUT @Format_on;"TRAC:DATA D1,";
1910  OUTPUT @Format_on;"#48192";
1920  OUTPUT @Format_off;Trace_out(*);
1930  OUTPUT @Format_on;CHR$(10)
1940  FOR I=1 TO 1024
1950   Trace_in(I)=Trace_out(I)
1960  NEXT I
1970  Running=True
1980  OUTPUT @Hp89410a;"INIT:CONT ON"
1990  OUTPUT @Hp89410a;"ABOR;:INIT; *WAI"
2000  OUTPUT @Hp89410a;"DISP:WIND1:TRAC:Y:AUTO ONCE"
2010  OUTPUT @Hp89410a;"DISP:WIND2:TRAC:Y:AUTO ONCE"
2020 SUBEND
2030 !
2040 SUB Shift_trace
2050 !-----------------------------------------------
2060 ! Shifts trace left or right by defined stepsize
2070 ! and loads shifted trace into D1
2080 !-----------------------------------------------
2090  COM /Traces/ Trace_in(1:1024),Trace_out(1:1024)
```

```
2100   Stepsize=10
2110   ON KEY 0 LABEL "",2 GOTO Waiting
2120   ON KEY 1 LABEL "Shift left",2 GOSUB Shift_left
2130   ON KEY 2 LABEL "Shift right",2 GOSUB Shift_right
2140   ON KEY 3 LABEL "",2 GOTO Waiting
2150   ON KEY 4 LABEL "",2 GOTO Waiting
2160   ON KEY 5 LABEL "Step size",2 GOSUB Step_size
2170   ON KEY 6 LABEL "",2 GOTO Waiting
2180   ON KEY 7 LABEL "",2 GOTO Waiting
2190   ON KEY 8 LABEL "Return",2 GOTO Sub_end
2200   !
2210 Waiting:  !
2220   WAIT .1
2230   GOTO Waiting
2240   !
2250 Shift_left:  ! shift the trace left "Stepsize" bins
2260   DISP "Shifting D1 data"
2270   I=1
2280   FOR J=Stepsize TO 1024
2290     Trace_out(I)=Trace_in(J)
2300     I=I+1
2310   NEXT J
2320   FOR J=1 TO Stepsize-1
2330     Trace_out(I)=Trace_in(J)
2340     I=I+1
2350   NEXT J
2360   CALL Data_out
2370   DISP ""
2380   RETURN
2390   !
2400   !----------------------------------------------------
2410 Shift_right:  ! shift the trace right "Stepsize" bins
2420   DISP "Shifting D1 data"
2430   I=1
2440   FOR J=Stepsize TO 1024
2450     Trace_out(J)=Trace_in(I)
2460     I=I+1
2470   NEXT J
2480   FOR J=1 TO Stepsize-1
2490     Trace_out(J)=Trace_in(I)
2500     I=I+1
2510   NEXT J
2520   CALL Data_out! send trace to D1
2530   DISP ""
2540   RETURN
2550   !----------------------------------------------------
2560 Step_size:  !
2570   INPUT "Enter shift stepsize (1..1024)",Stepsize
2580   RETURN
2590   !----------------------------------------------------
2600 Sub_end:SUBEND
```

OPC_SYNC

```
10  ! HP Instrument BASIC program: OPC_SYNC - Measurement synchronization
20  ! ----------------------------------------------------------------
30  ! This program demonstrates how to use the *OPC command to
40  ! allow an SRQ to interrupt program execution.  *OPC will set
50  ! the OPERATION_COMPLETE bit in the EVENT STATUS register
60  ! when all pending HP-IB commands have finished.  With the proper
70  ! register masks, this will generate a service request.
80  ! ----------------------------------------------------------------
90  !
100  Scode=8                    ! Interface select code
110  Address=0
120  Hp89410a=Scode*100+Address
125  OUTPUT Hp89410a;"SYST:PRES;*WAI"
130  !
140  OUTPUT Hp89410a;"FREQ:SPAN 50 HZ" !Measurement will take ~8 seconds
150  OUTPUT Hp89410a;"*CLS"          !Clear the STATUS BYTE register
160  OUTPUT Hp89410a;"*ESE 1"        !Program the EVENT STATUS ENABLE reg.
170  OUTPUT Hp89410a;"*SRE 32"       !Program the STATUS BYTE ENABLE reg.
180  !
190  ON INTR Scode,2 GOTO Srq_handler  !Set up interrupt branching
200  ENABLE INTR Scode;2             !Allow SRQ to generate an interrupt
210  !
220  OUTPUT Hp89410a;"ABORT; INIT"    !Start the measurement
230  OUTPUT Hp89410a;"*OPC"          !Generate SRQ when all commands have
240                                  !finished.
250  Start_time=TIMEDATE
260  LOOP                           !Do something useful while waiting
270   DISP USING "14A, 2D.D";"Elapsed time :",TIMEDATE-Start_time
280   WAIT .1
290  END LOOP
300  !
310 Srq_handler:                    !Got an SRQ
320   Stb=SPOLL(Hp89410a)           !Read STATUS BYTE and clear SRQ
330   BEEP
340   OUTPUT Hp89410a;"*ESR?"       !Read and clear EVENT STATUS reg.
350   ENTER Hp89410a;Esr
360   DISP "Got the SRQ! SPOLL returns:";Stb;"  ESR returns:";Esr
370  END
```

OPCQSYNC

```
10   ! HP Instrument BASIC program: OPCQSYNC - Measurement synchronization
20   ! ----------------------------------------------------------------
30   ! This program demonstrates how to use the *OPC? HP-IB command
40   ! to hang the bus on a query before continuing on with the
50   ! program. After all pending HP-IB commands have finished,
60   ! the HP 89410a will return a '1' in response to *OPC?.
70   ! ----------------------------------------------------------------
80   !
90   Scode=8
100  Hp89410a=Scode*100
110  !
120  OUTPUT Hp89410a;"SYST:PRES"       !Preset the HP89410a
130  OUTPUT Hp89410a;"*OPC?"           !Pause on ENTER statement until
140  ENTER Hp89410a;Opc                !PRESET command has finished
150  !
160  OUTPUT Hp89410a;"FREQ:SPAN 50 Hz" !Measurement will take ~8 seconds
170  DISP "Measurement started ..."
180  OUTPUT Hp89410a;"ABOR; INIT"      !Start the measurement
190  OUTPUT Hp89410a;"*OPC?"           !Pause until all pending HP-IB
200  ENTER Hp89410a;Opc                !commands have finished.
210  BEEP
220  DISP "Measurement done"
230  OUTPUT Hp89410a;"DISP:FORM TWO"
240  OUTPUT Hp89410a;"INIT:CONT OFF"
250  END
```

SNGL_SWP

```
10   ! HP Instrument BASIC example program: SNGL_SWP
20   !-------------------------------------------------------
30   ! This program demonstrates using the instrument's status
40   ! registers to enable SRQs for event initiated program
50   ! interrupts.  In this case the sweeping bit is used
60   ! to detect the end of sweep event.
70   !-------------------------------------------------------
80   !
90   Sc=8
100  Addr=0
110  Device=(Sc*100)+Addr
120  ASSIGN @Hp89410a TO Device
130  !
140  CLEAR SCREEN
150  GCLEAR
160  OUTPUT @Hp89410a;"ROUT:REC IF"
170  OUTPUT @Hp89410a;"SYST:PRES"
180  OUTPUT @Hp89410a;"INST SCAL"
190  OUTPUT @Hp89410a;"BAND 1000 HZ"
200  OUTPUT @Hp89410a;"INIT:CONT OFF"
210  IF Sc=8 THEN
220    OUTPUT @Hp89410a;"DISP:FORM TWO"
230    OUTPUT @Hp89410a;"DISP:PROG LOW"
240    CLEAR SCREEN
250  END IF
260  !
270  ! Setup registers to detect end of sweep
280  !
290  OUTPUT @Hp89410a;"*CLS"            ! clear any pending events
300  OUTPUT @Hp89410a;"*SRE 128"        ! allow SRQ from operation register
310  OUTPUT @Hp89410a;"STAT:OPER:ENAB 8"  ! allow SRQ from sweeping bit
320  OUTPUT @Hp89410a;"STAT:OPER:NTR 8"   ! latch sweeping FALSE
330  OUTPUT @Hp89410a;"STAT:OPER:PTR 0"   ! do not latch sweeping TRUE
340  !
350  ! set up interrupts
360  ON INTR Sc GOSUB Check_srq
370  ENABLE INTR Sc;2
380  !
390  OUTPUT @Hp89410a;"ABOR;:INIT"
400  !
410  ! Wait for SRQ
420  !
430 Hang_out:WAIT .1
440  GOTO Hang_out
450  !
460 Check_srq:    !
470  !
480  PRINT "SRQ Received"
490  Sb=SPOLL(Device)
500  PRINT "SPOLL(";Device;") = ";Sb
```

```
510  Queryswp(@Hp89410a)
520  ENABLE INTR Sc
530  RETURN
540  !
550  END
560  !************************************************
570  ! Query operational event status register and check
580  ! if sweeping event detected.
590  !************************************************
600  SUB Queryswp(@Device)
610   OUTPUT @Device;"STAT:OPER:EVEN?"
620   ENTER @Device;Resp
630   PRINT "STAT:OPER:EVEN?: ";Resp
640   IF Resp=8 THEN
650    INPUT "PRESS ENTER FOR NEXT SWEEP (ENTER 'Q' TO QUIT)",A$
660    IF UPC$(A$)="Q" THEN STOP
670    OUTPUT @Device;"INIT"
680    PRINT "Next sweep!"
690    PRINT
700   END IF
710  SUBEND
```

TRC_LOAD

```
10   ! HP Instrument BASIC example program : TRC_LOAD
20   !-------------------------------------------------------------
30   ! This program demonstrates downloading and uploading a trace
40   ! on the HP 89410A with IBASIC.
50   !-------------------------------------------------------------
60   !
70   DIM Dump_data(1:3201),Load_data(1:3201)
80   INTEGER Num_pts
90   !
100   Sc=8
110   Addr=0
120   Device=(Sc*100)+Addr
130   ASSIGN @Hp89410a TO Device
140   !
150   CLEAR SCREEN
160   !
170   ! Split screen
180   !
190   OUTPUT @Hp89410a;"DISP:FORM TWO"
200   !
210   ! Save contents of trace A into data register
220   !
230   OUTPUT @Hp89410a;"TRAC:COPY D2,TRAC1"
240   !
250   ! Dump trace contents of data register into array
260   !
270   DISP "Dumping trace..."
280   !
290   Dump_trace(@Hp89410a,Dump_data(*),Num_pts,"TRAC:DATA? D2")
300   !
310   ! Redimension Load_data array to match trace length
320   !
330   REDIM Load_data(1:Num_pts)
340   !
350   ! Flip-flop data points before uploading
360   !
370   DISP "Reversing data points..."
380   FOR I=1 TO Num_pts
390   Load_data(Num_pts-I+1)=Dump_data(I)
400   NEXT I
410   !
420   ! Upload reversed trace into data register
430   !
440   DISP "Uploading trace..."
450   Upload_trace(@Hp89410a,Load_data(*),Num_pts,"TRAC:DATA D1,")
460   DISP
470   !
480   ! Display reversed data register
490   !
500   OUTPUT @Hp89410a;"CALC2:FEED 'D1'"
```

```
510  !
520  !
530  END
540  !##############################################################################
550  !
560  SUB Dump_trace(@Hp89430a,REAL Trace_data(*),INTEGER Num_pts,Command$)
570  !------------------------------------------------------------------------
580  ! MODULE DESCRIPTION:
590  !  This module dumps a trace of data from the instrument.
600  !
610  ! INPUTS:  Command$  : HP-IB mnemonic used to prompt the instrument
620  !                  for the trace of data.
630  ! OUTPUTS:  Trace_data : Array of data received from instrument.
640  !         Num_pts   : Number of points dumped into Trace_data.
650  !
660  !------------------------------------------------------------------------
670   DIM A$[10]
680   INTEGER Dig_cnt
690   CLEAR @Hp89430a
700   OUTPUT @Hp89430a;"FORM:DATA REAL,64"
710   OUTPUT @Hp89430a;Command$
720   ASSIGN @Hp89430a;FORMAT ON
730   Dig_cnt=-1
740   ENTER @Hp89430a USING "%,A,D";A$,Dig_cnt
750   IF (A$ < > "#") OR (Dig_cnt < =0) THEN
760    PRINT "NOT CORRECT BLOCK MODE"
770    CLEAR @Hp89430a
780   ELSE
790    ENTER @Hp89430a USING "%,"&VAL$(Dig_cnt)&"D";Num_pts
800    IF (Num_pts MOD 8=0) THEN
810     Num_pts=Num_pts DIV 8
820     !
830     ! Redimension Trace_data array to match trace length
840     !
850     REDIM Trace_data(1:Num_pts)
860     !
870     ASSIGN @Hp89430a;FORMAT OFF
880     ENTER @Hp89430a;Trace_data(*)
890     ASSIGN @Hp89430a;FORMAT ON
900     ENTER @Hp89430a;A$          ! Read CR/LF
910    ELSE
920     PRINT Data_read;" not REAL (double) size (divisible by 8)"
930     CLEAR @Hp89430a
940    END IF
950   END IF
960  SUBEND
970  !##############################################################################
980  !
990  SUB Upload_trace(@Hp89410a,REAL Trace_data(*),INTEGER Num_pts,Command$)
1000 !------------------------------------------------------------------------
1010 ! MODULE DESCRIPTION:
1020 !  This module uploads a trace of data.
1030 !
```

```
1040  ! INPUTS:  @Hp89410a  : Device to dump data to.
1050  !          Command$  : HP-IB mnemonic used to load the trace of
1060  !                      data into the instrument
1070  ! OUTPUTS:  Trace_data : Array of data to load into instrument.
1080  !
1090  !-------------------------------------------------------------------
1100   DIM Header$[20]
1110   !
1120   ! Format block header
1130   !
1140   Dig_cnt=INT(LGT(Num_pts*8)+.9999)
1150   OUTPUT Header$ USING "#,D,"&VAL$(Dig_cnt)&"D";Dig_cnt,Num_pts*8
1160   Header$=" #"&Header$
1170   !
1180   CLEAR @Hp89410a
1190   OUTPUT @Hp89410a;"FORM:DATA REAL,64"
1200   OUTPUT @Hp89410a;Command$;Header$;
1210   ASSIGN @Hp89410a;FORMAT OFF
1220   OUTPUT @Hp89410a;Trace_data(*),END
1230   ASSIGN @Hp89410a;FORMAT ON
1240   OUTPUT @Hp89410a;CHR$(10)
1250  SUBEND
1260  !
```

TWO_CTLR

```
10  !HP BASIC program:  TWO_CTLR - Two controller operation
20  !-------------------------------------------------------
30  !This program demonstrates how an external controller
40  !and HP Instrument BASIC can work together.  This program
50  !will download a BASIC program to the HP 89410A and run it two
60  !times.  After each run, two BASIC program variables will
70  !will be read from the HP 89410A and displayed.
80  !-------------------------------------------------------
90  !
100  Scode=7                    !Select code for interface
110  Address=11                 !Address for HP 89410A
120  Hp89410a=Scode*100+Address
130  !
140  CLEAR Hp89410a
150  OUTPUT Hp89410a;"PROG:DEL:ALL"   !Scratch the program space
160  !
170  DISP "Downloading the program..."
180  ASSIGN @Prog TO Hp89410a;EOL CHR$(10) !Change EOL character
190  OUTPUT @Prog;"PROG:DEF #0";        !Send program
200  OUTPUT @Prog;"10 COM INTEGER Times_run,Test$[10]"
210  OUTPUT @Prog;"20 Times_run=Times_run +1"
220  OUTPUT @Prog;"30 IF Times_run=1 THEN Test$=""PASS"""
230  OUTPUT @Prog;"40 IF Times_run=2 THEN Test$=""FAIL"""
240  OUTPUT @Prog;"50 BEEP"
250  OUTPUT @Prog;"60 END"
260  OUTPUT @Prog;CHR$(10) END       !Terminate the data block
270  !
280  !Set up registers for interrupt on PROGRAM_RUNNING going false
290  OUTPUT Hp89410a;"*CLS"          !Clear the STATUS register
300  !Program NTR reg and OPERATION ENABLE reg for PROGRAM_RUNNING bit
310  OUTPUT Hp89410a;"STAT:OPER:NTR 16384"
320  OUTPUT Hp89410a;"STAT:OPER:ENAB 16384"
330  OUTPUT Hp89410a;"*SRE 128"   !Allow SRQ on bit 7 of STATUS reg
340  !
350  DISP "Running the program..."
360  OUTPUT Hp89410a;"PROG:STAT RUN"  !Run Program
370  Display_res(Hp89410a,Scode)     !Read and display variables
380  OUTPUT Hp89410a;"PROG:STAT RUN"  !Run Program  again
390  Display_res(Hp89410a,Scode)     !Read and display variables
400  DISP "Done"
410  !
420  END                        !End of this program
430  !
440  SUB Display_res(Hp89410a,Scode)
450  ! This subprogram waits for an SRQ interrupt to signal that a
460  ! BASIC program has finished.  It then clears the HP-IB registers
470  ! by reading them.  Once that is done, the values of two IBASIC
480  ! variables are read and displayed.
490  !
500  ON INTR Scode GOTO Read_results    !Set up interrupt branching
```

```
510   ENABLE INTR Scode;2              !Allow interrupt on SRQ
520 Idle:   GOTO Idle
530   !
540 Read_results:                      !Program has finished
550   A=SPOLL(Hp89410a)                !Read and clear the SRQ
560   OUTPUT Hp89410a;"STAT:OPER?" !Read and clear OPERATION STATUS reg.
570   ENTER Hp89410a;Event
580   WAIT .5
590   !
600   OUTPUT Hp89410a;"FORM:DATA ASCII,3"
610   OUTPUT Hp89410a;"PROG:NUMB? 'Times_run'"!Read the first variable
620   ENTER Hp89410a;Times_run
630   !
640   OUTPUT Hp89410a;"PROG:STR? 'Test$'"    !Read the second variable
650   ENTER Hp89410a;Test$
660   !
670   PRINT "Times_run: ";Times_run,"Test$: ";Test$
680   SUBEND
```

WAI_SYNC

```
10  ! HP Instrument BASIC program: WAI_SYNC - Measurement synchronization
20  ! ----------------------------------------------------------------
30  ! This program demonstrates how to use the *WAI command to
40  ! prevent execution of an HP-IB command until all previous
50  ! commands have finished. In this example, the trace display
60  ! measurement has finished.
70  ! The *WAI command does not affect program operation. The
80  ! program will run to completion, sending all of the commands to
90  ! to the HP89410A without waiting for them to be executed.
100 ! ----------------------------------------------------------------
110 Scode=8                    !Interface select code
120 Address=0
130 Hp89410a=Scode*100+Address
140 !
150 DISP "Sending HP-IB commands..."
160 OUTPUT Hp89410a;"SYST:PRES"
170 OUTPUT Hp89410a;"AVER:COUN 1"
180 OUTPUT Hp89410a;"AVER ON"
190 OUTPUT Hp89410a;"FREQ:SPAN 50 HZ"!Set narrow span
200 OUTPUT Hp89410a;"ABORT; INIT"   !Start the measurement
210 OUTPUT Hp89410a;"*WAI"         !Tell analyzer to wait here until
220                          !all HP-IB commands have finished
230 OUTPUT Hp89410a;"DISP:FORM TWO"  !Go to upper/lower after waiting
240 DISP "Display will go to 2 grids when measurement complete."
250 END
```

Chapter 10

Instrument-Specific HP Instrument BASIC Features

10

Instrument-Specific HP Instrument BASIC Features

Introduction

The *HP Instrument BASIC Users Handbook* that accompanies this manual is divided into the following sections:

- "HP Instrument BASIC Programming Techniques"
- "HP Instrument BASIC Interfacing Techniques"
- "HP Instrument BASIC Language Reference"

The *HP Instrument BASIC Users Handbook* is included with all Hewlett-Packard instruments that use HP Instrument BASIC. Since each instrument is different, the way that HP Instrument BASIC interfaces and interacts with its host often changes from one instrument to another.

For example, some instruments employ editors, while others do not, and front panel interfaces often vary a great deal from one instrument to another. For this reason, many parts of the *HP Instrument BASIC Users Handbook* are either generic in nature, or apply to only one of many possible instrument interfaces.

This chapter describes how to use the *HP Instrument BASIC Users Handbook* for the HP 89400-series analyzers. Global exceptions apply throughout the handbook. These differences are discussed by category. Specific differences for each command are listed in table 10-4.

Global Exceptions

Global Exceptions can be categorized as follows:

- HP-IB, GPIO and RS-232 interfaces
- Display and keyboard interfaces
- Disk I/O
- Miscellaneous command differences

Each of these categories is explained in detail in the following sections.

HP-IB, GPIO and RS-232 Interfaces

The *HP Instrument BASIC Users Handbook* refers to various interface types, particularly in chapter 2 of "HP Instrument BASIC Interfacing Techniques." HP Instrument BASIC in the HP 89400-series analyzers supports only the HP-IB interface, and not the GPIO, serial, or parallel interfaces.

In addition, HP Instrument BASIC in the HP 89400-series analyzers supports only two interface select codes: 7 for the external bus, and 8 for the internal bus. This affects the following commands:

- ABORT
- ENABLE INTR
- DISABLE INTR
- ON INTR
- OFF INTR
- ON TIMEOUT
- OFF TIMEOUT

Select code 8, the internal HP-IB interface, is not valid with the LOCAL LOCKOUT and REMOTE commands. Only select code 7, the external HP-IB interface may be used with these commands.

When using the statements ENABLE INTR and ON INTR, the line containing the ON INTR statement must always precede the line containing ENABLE INTR.

The select codes for GPIO (select code 12), serial interface (select code 9), and parallel interface (select code 26) are not valid in the HP 89400-series analyzers. The internal default select code for the display (select code 1) and for the keyboard (select code 2) are valid. The display function returns 1. "ENTER CRT" is not supported. The KBD function returns 2. Select code 2 may be used with the ENTER statement, as in ENTER 2, which allows text to be entered from the keyboard.

Display and Keyboard Interfaces

The following section describes the differences between the standard interface assumed by the *HP Instrument BASIC Users Handbook* and the HP 89400-series analyzers' display and keyboard interface. In addition, the HP 89400-series analyzers' keys that emulate command line execution of HP Instrument BASIC keywords are listed.

Display Differences

Most references to the display (CRT) in the *HP Instrument BASIC Users Handbook* assume a standard 80 column terminal. The HP 89400-series analyzers have a 54 column display for text. This affects references to the width of the default PRINTER IS device (the display) in the LIST, PRINT and PRINTER IS commands.

You must allocate a display partition to view output to the display because the instrument shares the display with HP Instrument BASIC. This affects both the text commands listed above, as well as the graphics commands, MOVE and DRAW. Three different display partitions, UPPER, LOWER or FULL, may be allocated. The text width for all three is the same. The only change for the text commands is how much text is displayed at one time.

The PEN command is implemented with 16 choices, 0 through 15. Pens 1 through 15 draw lines in the colors shown in Table 10-1. PEN 0 erases all line segments that any following DRAW commands encounter. PEN 0 does *not* perform a pixel complement function as it does in HP BASIC.

Table 10-1. Pens and Associated Graphics Colors

Pen	Color
0	Erase
1	White
2	Red
3	Yellow
4	Green
5	Cyan
6	Blue
7	Magenta
8	Black
9	Olive Green
10	Aqua
11	Royal Blue
12	Maroon
13	Brick Red
14	Orange
15	Brown

Keyboard Differences

The *HP Instrument BASIC Users Handbook* assumes the use of a standard HP BASIC Series 200/300 workstation keyboard. It also assumes that HP Instrument BASIC works in "command line execution mode," where individual commands may be entered and executed from the keyboard.

HP Instrument BASIC in the HP 89400-series analyzers works with an HP approved PC keyboard and does *not* use command line execution mode. When using the HP Instrument BASIC editor, the HP 89400-series analyzers' front-panel hardkeys become alpha keys. Softkey menus emulate many of the keywords that are executable from the command line on an HP BASIC workstation (such as RUN, CONTINUE, and SCRATCH).

The following hardkeys and softkeys correspond to HP Instrument BASIC keywords:

CONTINUE

[BASIC] [continue]

[BASIC] [debug] [continue]

DEL

[BASIC] [edit] [delete line]

EDIT

[BASIC] [edit]

LIST

[BASIC] [print program]

PAUSE

[BASIC]

RUN

[BASIC] [run]

[BASIC] [debug] [run]

SCRATCH

[BASIC] [utilities] [scratch]

SECURE

[BASIC] [utilities] [secure]

STOP

[Local/Setup]

REN

[BASIC] [utilities] [renumber]

The following HP Instrument BASIC keywords are described in the "Language Reference" section of the *HP Instrument BASIC Users Handbook* in terms of a standard workstation keyboard. Their use with the HP 89400-series analyzers is described below:

EDIT

Ignore all documentation in the "HP Instrument BASIC Language Reference" for the EDIT command. See chapter 5, "Developing Programs," for information on using the HP Instrument BASIC editor in the HP 89400-series analyzers.

ON KEY and OFF KEY

Eight softkeys are available for use in the HP 89400-series analyzers. They appear on the right side of the display where instrument softkeys normally appear. Key selector values range from 0 through 7. In addition, the softkeys are loaded into the function keys ([**F1**] - [**F8**]) with the keyboard.

INPUT

When an INPUT statement is encountered in an HP Instrument BASIC program, the alpha entry menu appears on the display. You can use your keyboard, or the front panel alpha keys, the numeric keypad and the symbol softkeys to enter a response.

To enter an input response press the [enter] softkey in the alpha entry menu or the [**Enter**] key on the keyboard. Disregard all keys mentioned in the "HP Instrument BASIC Language Reference" description of this key.

You have two options for terminating an INPUT command:

> Press [enter] (either the softkey or the [**Enter**] key on the keyboard) to complete entry.
> Press the **[BASIC]** hardkey to pause the program.

To continue the program after pressing the **[BASIC]** hardkey, press the [continue] softkey in the **[BASIC]** menu or the [single step] softkey in the [debug] menu. The INPUT statement is re-executed.

You *cannot* press the **[Local/Setup]** hardkey to stop the program because it is redefined as an alpha key whenever the alpha entry menu appears.

To enter an input response while a program is under remote control—that is, an external controller is executing the program—the program *must be returned* to local (front panel) control. Press the **[Local/Setup]** hardkey to return the program to local control. Enter the input response as described above. The instrument remains in local control after terminating the input command. Pressing the **[Local/Setup]** hardkey again, resets the program. It is recommended that you specify the exact key sequence expected in your input prompt.

ENTER

For a description of using ENTER with a keyboard (ENTER KBD) see the previous description for the INPUT statement.

Note

In the "HP Instrument BASIC Language Reference" every command contains a line stating whether or not the command is "keyboard executable." Disregard this information for HP Instrument BASIC in the HP 89400-series analyzers. The keyboard for the HP 89400-series analyzers does not operate in this manner.

Disk I/O

The following section specifies the implementation of disk I/O functions.

Disk I/O Commands

Many HP Instrument BASIC commands that pertain to the disk I/O (SAVE, RE-SAVE, COPY, MSI, etc.) have similar functions executed by normal front-panel operations. *These front-panel operations are not considered to be HP Instrument BASIC functions.*

For example, the MASS STORAGE IS command when executed in a program is totally independent of the current storage device found under the **[Save/Recall]** [default disk] key. Conversely, using the [default disk] key to change the default storage device has no impact on any MSI statements within an HP Instrument BASIC program.

Volume Specifiers

HP Instrument BASIC in the HP 89400-series analyzers supports four mass storage devices; the internal disk drive, volatile RAM disk (memory unit 0) and non-volatile RAM disk (memory unit 1) and an external
disk drive (Hewlett-Packard Subset/80). This affects the volume specifier parameter in the following commands:

- ASSIGN
- CAT
- COPY
- CREATE
- CREATE ASCII
- CREATE BDAT
- CREATE DIR
- GET
- INITIALIZE
- MASS STORAGE IS
- PRINTER IS
- PURGE
- RENAME
- RE-SAVE
- SAVE

Valid volume specifiers for each mass storage device are shown in table 10-1.

Table 10-1. Mass Storage Volume Specifiers

Disk	Volume Specifier	MSI Specifier
EXTERNAL	:EXTERNAL,7XX,uu	EXT:
	:EXTERNAL,7XX	
	:EXTERNAL	
	:,7XX	
INTERNAL	:INTERNAL,4,0	INT:
	:,4,0	
	:,4	
VOLATILE RAM	:MEMORY,0,0	RAM:
	:MEMORY,0	
	:,0	
NON-VOLATILE RAM	:MEMORY,0,1	NVRAM:
	:,0,1	

XX bus address
uu unit number

Disk Format

HP Instrument BASIC in the HP 89400-series analyzers recognizes two types of disk and file formats; LIF (Logical Interface Format) and DOS (Disk Operating System). Although the HP 89400-series analyzers recognize DOS format and directories, they do not support HFS (Hierarchical File System). The *HP Instrument BASIC Users Handbook* addresses LIF, DOS, and HFS file systems. In general, disregard all references to HFS throughout the *HP Instrument BASIC Users Handbook*.

Note The HP 89400-series analyzers do not support HFS.

File Types

HP Instrument BASIC can create three types of files: ASCII, BDAT, and untyped. These files can exist on either a DOS or a LIF formatted disk.

If you catalog a DOS disk, these three types show up as "ASCII," "BDAT," and "DOS." If you catalog a LIF disk, these three types show up as "ASCII," "BDAT," and "HP-UX." Table 10-2 indicates these configurations.

Table 10-2. HP Instrument BASIC File Types

File Type	Appears on a LIF disk as	Appears on a DOS disk as
ASCII	ASCII	ASCII
BDAT	BDAT	BDAT
untyped	HP-UX	DOS

HP Instrument BASIC supports LIF protect codes only on BDAT files. An error is generated if a LIF protect code is encountered on an ASCII file. HP Instrument BASIC ignores a LIF protect code on an untyped file.

A special note about file types and file systems:

The "HP Instrument BASIC Language Reference" sometimes uses the terms HP-UX file and HFS interchangeably or refers to HP-UX files only in context of HFS volumes. In fact, HP-UX files can exist on a LIF volume, which the HP 89400-series analyzers support. Be careful when reading the descriptions in the "HP Instrument BASIC Language Reference." The HP 89400-series analyzers support HP-UX files on LIF volumes only. The HP 89400-series analyzers do not support HP-UX files on HFS volumes.

The ASCII file type described in this section is specific to HP's LIF file system and is *not* the same as the standard "ASCII" file type in a DOS environment. If you copy an ASCII file from a LIF disk to a DOS disk, the file appears as an "ASCII" file. However, the file is not usable with DOS-system editors. *Untyped files* are the *only files* you can edit with a DOS ASCII editor on a PC.

An untyped file is automatically generated whenever an HP Instrument BASIC program is SAVEd from the HP 89400-series analyzers to a DOS-formatted disk. A RE-SAVE maintains the original file type if a file exists, otherwise it performs the same action as SAVE.

Formatting Disks

Formatting a disk prepares it for use. The HP 89400-series analyzers recognize both LIF and DOS formats and have the capability to format either type in the [format disk] menu.

You can also use the INITIALIZE statement to format a disk.

Caution

Existing files on the media are destroyed when formatting or executing the INITIALIZE command.

The [format disk] Menu

The [format disk] menu under the [**Disk Utility**] hardkey allows you to define format parameters and to format a disk using these parameters. See figure 10-1.

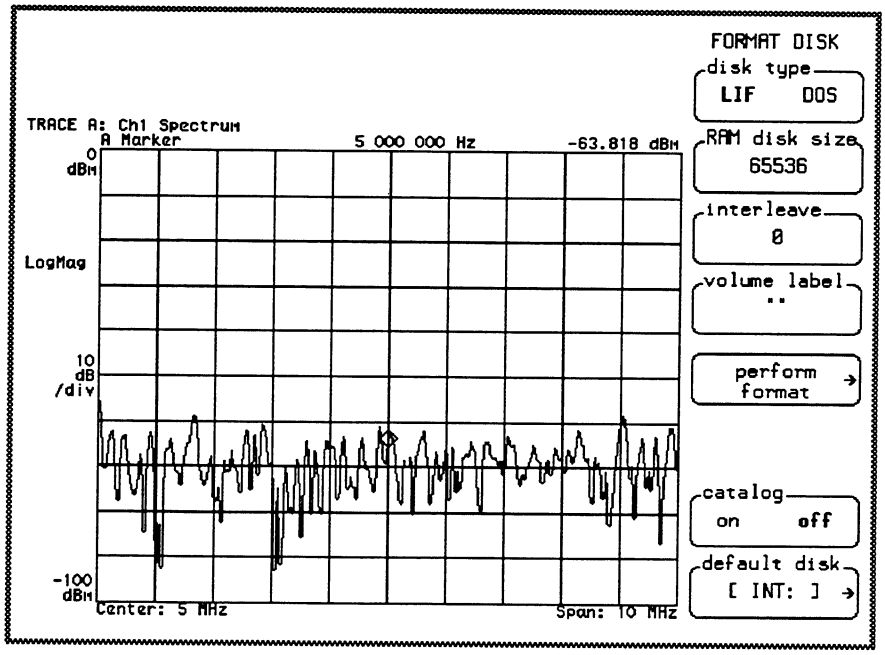

Figure 10-1. The [FORMAT DISK] Menu

The [disk type LIF DOS] softkey allows you to select the type of format. The default type is DOS. Press this key to toggle the selection to LIF.

The [RAM disk size] softkey specifies the storage capacity of the Volatile RAM disk. The default value is 64 KBytes of storage. Use the numeric keypad or the arrow keys to enter a new value. Values are rounded up to the next highest KByte (1024 byte) increment. The entry window appears at the top of the display when you press the softkey.

Note

You can use the [format disk] menu to format non-volatile RAM (NVRAM), the internal disk drive or the external disk drive. However, the [RAM disk size] specification is ignored.

The [interleave] softkey defines the ordering of the sectors on the 3.5 inch flexible disks. To specify the default value for the disk, specify 0. This value is ignored when formatting the non-volatile RAM (NVRAM).

Press [perform format] to start the format process. An entry window at the top of the screen displays the specifier for the default disk. See table 10-1 for the MSI specifiers. Use the alpha entry keys or the keyboard to modify the field. The current values for [disk type], [RAM disk size] and [interleave] are used.

The INITIALIZE Statement

The disk format, LIF or DOS, is specified when the media is initialized. The INITIALIZE statement takes the following form:

INITIALIZE "*<disk format>*: *<volume specifier>*" *<interleave factor>*,
<format option>

If *<disk format>* is not specified, the default format is LIF.

The *<volume specifier>* for the volatile RAM disk includes a *<unit size>* parameter,that specifies the number of 256-byte sectors. The actual size is memory dependent and ranges from 4 thru 32767.

The *<format option>* parameter specifies the capacity of the flexible disk drive (internal or external). See table 10-3 for the valid format options.

Table 10-3. Flexible Disk Format Options

Media	Format Option	Bytes/Sector	Sectors/Track	Tracks/Surface	Maximum Capacity (bytes)
1-MByte	0	256	16	77	630,784
	1*	256	16	77	630,784
	2	512	9	77	709,632
	3	1,024	5	77	788,480
	4**	256	16	77	270,336
	16	512	9	80	737,280
2-MByte	0	256	32	77	1,261,568
	1***	256	32	77	1,261,568
	2	512	18	77	1,419,264
	3	1,024	10	77	1,576,960
	4***	256	32	77	1,261,568
	16	512	18	80	1,474,560

* Same as Format Option 0 (default) when using 1-MByte media.
** Format Option 4 (singled sided disk format) is not supported in internal disk drive (INT:).
*** Same as Format Option 0 (default) when using 2-MByte media.

Note

Table 10-3 specifies the *maximum* capacity for each format option. Actual capacity is dependent upon the file system type, LIF or DOS.

For example, to INITIALIZE a flexible disk in the internal disk drive in LIF format, use the following:

```
INITIALIZE "LIF:,4,0",1
```

To INITIALIZE a DOS disk in an external disk drive, use the following:

```
INITIALIZE "DOS:,700,0",1,16
```

In this example, the format option, 16, is important when initializing a DOS disk. An incorrect format option results in a disk that the HP 89400-series analyzers can use, but other DOS systems cannot use. This potential problem can be avoided by formatting the disk on a DOS system, rather than the
HP 89400-series analyzers.

Caution

An incorrect format option may prevent other DOS systems from using the DOS disk.

You can use the INITIALIZE statement to format the non-volatile disk (NVRAM). However, the size of NVRAM is fixed and the interleave factor is ignored. You can only change the file format.

Once initialization is complete, file format, "LIF" or "DOS", is not specified in any other file operations. HP Instrument BASIC automatically determines the format of the disk.

Miscellaneous Command Differences

COS

The range of the COS command is all absolute values less than 1.7083127722 e+10 degrees.

SYSTEM$

The HP 89400-series analyzers do not support the topic specifier, (SYSTEM VERSION:).

Note

Since the *HP Instrument BASIC Users Handbook* is continually revised to support all implementations of HP Instrument BASIC, there may be other commands that appear in that documentation that are not supported in the HP 89400-series analyzers. Table 10-4 in the following section lists all HP Instrument BASIC keywords supported by HP Instrument BASIC in the HP 89400-series analyzers.

Specific Exceptions

Table 10-4 summarizes the HP Instrument BASIC keyword implementation in the HP 89400-series analyzers. The table indicates if the keyword has front panel support. (If it does, the key path is given.) Table 10-4 also lists the major differences between the descriptions of these keywords in the "HP Instrument BASIC Language Reference" and the way they are implemented in the HP 89400-series analyzers. Where differences are too extensive to be summarized, references to their explanation in the "Global Exceptions" section are given.

Any keywords or functions found in the "HP Instrument BASIC Language Reference" that do not appear in this table, do not apply to HP Instrument BASIC in the HP 89400-series analyzers and should be ignored.

Table 10-4. HP 89400-series Keyword Implementation

Command	Front Panel Support	Exceptions
ABORT	None	Interface Select Code = 7 or 8
ABS	None	None
ACS	None	None
ALLOCATE	None	None
ALPHA ON | OFF	[**BASIC**] [DISPLAY SETUP] [ALPHA ON OFF]	None
AND	None	None
AREA	None	Not supported
ASN	None	None
ASSIGN	None	One HP-IB device per ASSIGN statement LIF protect code supported in BDAT files only Does not support HFS volumes See "Disk I/O"
ATN	None	None
AXES	None	None
BASE	None	None
BEEP	None	None
BINAND	None	None
BINCMP	None	None
BINEOR	None	None
BINIOR	None	None
BIT	None	None
CALL	None	None

Table 10-4. HP 89400-series Keyword Implementation (Continued)

Command	Front Panel Support	Exceptions
CAT	None (Independent of **[Disk Util]** and **[Save/Recall]** functions)	Does not support HFS catalogs See "Volume Specifiers" in "Disk I/O"
CAUSE ERROR	None	None
CHR$	None	None
CLEAR	None	None
CLEAR ERROR	None	None
CLEAR SCREEN	None	None
CLIP	None	None
COM	None	None
CONT	**[BASIC]** [continue] or **[BASIC]** [debug] [continue]	No line number or label support
CONTROL	None	Not supported
COPY	None (Independent of **[Disk Util]** functions)	LIF protect code in BDAT files only Does not support HFS volumes See "Volume Specifiers" in "Disk I/O"
COPYLINES	None	Not supported
COS	None	Absolute range values less than 1.7083127722 e+10
CREATE	None	Does not support HFS volumes See "Disk I/O"
CREATE ASCII	None	Does not support HFS volumes See "Disk I/O"
CREATE BDAT	None	LIF protect code allowed Does not support HFS volumes See "Disk I/O"
CREATE DIR	None	Does not support HFS volumes See "Volume Specifiers" in "Disk I/O"
CRT	None	ENTER CRT (ENTER 1) not supported
CSIZE	None	None
DATA	None	None
DATE	None	None
DATE$	None	None
DEALLOCATE	None	None
DEF FN	None	None

Table 10-4. HP 89400-series Keyword Implementation (Continued)

Command	Front Panel Support	Exceptions
DEG	None	None
DEL	[**BASIC**] [edit][delete line]	Deletes only the current line
DELSUB	[**BASIC**] [utilities] [delsub]	None
DET	None	None
DIM	None	None
DISABLE	None	None
DISABLE INTR	None	Interface Select Code = 7 or 8
DISP	None	None
DIV	None	None
DOT	None	None
DRAW	None	Maximum x,y coordinates: Full partition (452,341) Upper partition (452,169) Lower partition (452,169)
DROUND	None	None
DUMP	None	Not supported
DVAL	None	None
DVAL$	None	None
EDIT	[**BASIC**][edit]	Editing functions described in chapter 5.
ELSE	None	None
ENABLE	None	None
ENABLE INTR	None	Interface Select Code = 7 or 8 Must not precede an ON INTR statement.
END	None	None
END IF	None	None
END LOOP	None	None
END SELECT	None	None
END WHILE	None	None
ENTER	None	Select Code = 2, 7, or 8 only
ERRL	None	None
ERRLN	None	None
ERRM$	None	None

Table 10-4. HP 89400-series Keyword Implementation (Continued)

Command	Front Panel Support	Exceptions
ERRN	None	None
EXIT IF	None	None
EXOR	None	None
EXP	None	None
FILL	None	None
FN	None	None
FNEND	None	None
FOR...NEXT	None	None
FORMAT	None	None
FRACT	None	None
FRAME	None	None
GCLEAR	None	None
GESCAPE	None	None
GET	None (Independent of **[Save/Recall]** functions)	Does not support HFS volumes See "Volume Specifiers" in "Disk I/O"
GINIT	None	None
GLOAD	None	None
GOSUB	None	None
GOTO	None	None
GRAPHICS ON | OFF	**[BASIC]** [BASIC display format] [graphics on off]	None
GRID	None	None
GSTORE	None	None
IDN	None	None
IDRAW	None	None
IF...THEN	None	None
IMAGE	None	None
IMOVE	None	None
INDENT	**[BASIC]** [utilities] [indent]	None
INITIALIZE	None (Independent of **[Disk Util]** functions)	Does not support HFS volumes See "Disk I/O"

Table 10-4. HP 89400-series Keyword Implementation (Continued)

Command	Front Panel Support	Exceptions
INPUT	None	See INPUT command in "Keyboard Differences" section
INT	None	None
INTEGER	None	None
IVN	None	None
IPLOT	None	None
IVAL	None	None
IVAL$	None	None
KBD	None	External Keyboard (2) or front-panel alpha keys
LABEL	None	None
LDIR	None	None
LEN	None	None
LET	None	None
LGT	None	None
LINE TYPE	None	Line types are different from those listed in the Language Reference
LIST	[**BASIC**][print program]	Default width = 58 (see PRINTER IS)
LOAD	None	None
LOADSUB	None	None
LOADSUB ALL FROM	None	None
LOCAL	None	None
LOCAL LOCKOUT	None	Interface Select Code = 7 only
LOG	None	None
LOOP	None	None
LORG	None	None
LWC$	None	None
MASS STORAGE IS	None (Independent of [**Save/Recall**] and **Disk Util**] functions)	Does not support HFS volumes See "Disk I/O" External disks must be online
MAT	None	None
MAT REORDER	None	None

Table 10-4. HP 89400-series Keyword Implementation (Continued)

Command	Front Panel Support	Exceptions
MAX	None	None
MAXREAL	None	None
MERGE ALPHA WITH GRAPHICS	None	Not supported
MIN	None	None
MINREAL	None	None
MOD	None	None
MODULO	None	None
MOVE	None	Maximum x, y coordinates: Full partition (452,341) Upper partition (452,169) Lower partition (452,169)
MOVELINES	None	Not supported
NOT	None	None
NUM	None	None
OFF CYCLE	None	None
OFF ERROR	None	None
OFF INTR	None	Interface Select Code = 7 or 8 Must precede ENABLE INTR statement.
OFF KEY	None	Key selectors are 0 thru 9
OFF TIMEOUT	None	Interface Select Code = 7 or 8
ON	None	None
ON CYCLE	None	None
ON ERROR	None	None
ON INTR	None	Interface Select Code = 7 or 8
ON KEY	None	Key selectors are 0 thru 8
ON TIMEOUT	None	Interface Select Code = 7 or 8
OPTION BASE	None	None
OR	None	None
OUTPUT	None	Select Code = 1, 7 or 8
PASS CONTROL	None	Interface Select Code 8 (pass control of external bus to analyzer)
PAUSE	None	None

Table 10-4. HP 35665A Keyword Implementation (Continued)

Command	Front Panel Support	Exceptions
PDIR	None	None
PEN	None	0 = erase 1 = white 2 = red 3 = yellow 4 = green 5 = cyan 6 = blue 7 = magenta 8 = black 9 = olive green 10 = Aqua 11 = Royal Blue 12 = maroon 13 = brick red 14 = orange 15 = brown
PENUP	None	None
PI	None	None
PIVOT	None	None
PLOT	None	None
PLOTTER IS	None	Not supported
POLYGON	None	Solid fill only
POLYLINE	None	None
POS	None	None
PRINT	None	PRINTER IS default width = 58
PRINTER IS	None	default width = 58 LIF protect code in BDAT files only Does not support HFS volumes See "Volume Specifiers" in "Disk I/O"
PROUND	None	None
PRT	None	None
PURGE	None (Independent of **[Disk Util]** functions)	LIF protect code in BDAT files only Does not support HFS volumes See "Volume Specifiers" in "Disk I/O"
RAD	None	None
RANDOMIZE	None	None
RANK	None	None
RATIO	None	None
READ	None	None
REAL	None	None
RECTANGLE	None	Solid fill only
RECOVER	None	None
REDIM	None	None
REM	None	None

Table 10-4. HP 89400-series Keyword Implementation (Continued)

Command	Front Panel Support	Exceptions
REMOTE	None	Does not support Interface Select Code 8
REN	[**BASIC**] [utilities] [renumber]	No line label support
RENAME	None (Independent of [**Disk Util**] functions)	LIF protect code in BDAT files only Does not support HFS volumes See "Volume Specifiers" in "Disk I/O"
REORDER	None	None
REPEAT...UNTIL	None	None
RE-SAVE	None (Independent of [**Save/Recall**] functions)	Does not support HFS volumes See "Disk I/O"
RESTORE	None	None
RE-STORE	None	None
RETURN	None	None
RETURN . . .	None	None
REV$	None	None
RND	None	None
ROTATE	None	None
RPLOT	None	None
RPT$	None	None
RSUM	None	None
RUN	[**BASIC**] [run] or [**BASIC**] [debug] [run]	None
SAVE	None (Independent of [Save/Recall] functions)	Does not support HFS volumes See "Disk I/O"
SCRATCH	[**BASIC**] [utilities] [scratch]	Does not support HFS volumes
SECURE	[**BASIC**] [utilities] [secure]	None
SELECT...CASE	None	None
SEPARATE ALPHA	None	Not supported
SET PEN	None	Not supported
SET TIME	None	None
SET TIMEDATE	None	None

Table 10-4. HP 89400-series Keyword Implementation (Continued)

Command	Front Panel Support	Exceptions
SGN	None	None
SHIFT	None	None
SHOW	None	None
SIN	None	None
SIZE	None	None
SPOLL	None	None
SQR	None	None
SQRT	None	None
STATUS	None	Not supported
STOP	None	None
STORE	None	None
SUB	None	None
SUBEND	None	None
SUBEXIT	None	None
SUM	None	None
SYSTEM PRIORITY	None	None
SYSTEM$	None	Does not support "VERSION:"
TAB	None	None
TABXY	None	None
TAN	None	None
THEN	None	None
TIME	None	None
TIME$	None	None
TIMEDATE	None	None
TRIGGER	None	None
TRIM$	None	None
TRN	None	None
UPC$	None	None
VAL	None	None
VAL$	None	None

Table 10-4. HP 89400-series Keyword Implementation (Continued)

Command	Front Panel Support	Exceptions
VIEWPORT	None	None
WAIT	None	None
WHERE	None	None
WHILE	None	None
WIDTH	None	None
WILDCARDS	None	Does not support UX
WINDOW	None	None

Index

Index

E

F

G

H

T

U