HEWLETT-PACKARD

HP 82718A

Expansion Pod

Reference Manual

For use with the HP-75

HEWLETT PACKARD

HP 82718A Expansion Pod

Reference Manual

For Use With the HP-75

July 1984

82718-90001

Printed in U.S.A

Printing History

Edition 1 ..... July 1984

Contents

Section 1
INTRODUCTION

Section 2
SOFTWARE USAGE AND EXAMPLES

Appendix E
KEYWORD AND COMMAND SUMMARY

INTRODUCTION

SECTION 1

Congratulations on your purchase of the HP 82718A Expansion Pod. This device attaches directly to an HP-75 portable computer and enhances its abilities as a powerful tool for portable data collection applications. This introductory section of the manual will highlight some of the key features and capabilities you now have at your fingertips.

HP 82718A HARDWARE

The combination of the HP 82718A with your HP-75 is a single, rugged, integrated package with data communications capability and a large memory for program and data storage. The HP 82718A connects to the right-most ROM port on the front of your HP-75, leaving two ports available for up to 64 kbytes of your ROM-based application software.

Built-In Modem

The built-in modem is a direct-connect 300 baud modem compatible with Bell 103/113 modems. It has originate, answer, and auto-answer capability, and can dial manually or automatically using either tones or pulses. The modem supports both XON/XOFF and ENQ/ACK handshake protocols, and several parity options (odd, even, always 1, always 0, or none). Two modular RJ11 jacks enable you to use the modem as an auto-dialer or use a telephone as a line monitor.

Electronic Disc

The HP 82718A Option 032 or 064 provides either 32 or 64 kbytes of additional random access memory (RAM). This additional RAM simulates a mass storage device such as a digital cassette or disc drive. Unlike a mechanical device, however, you get fast data transfer and rugged, reliable operation.

HP 82718A SOFTWARE

The HP 82718A contains 16 kbytes of ROM-based software that provides 34 new BASIC keywords to the HP-75's already powerful BASIC language, plus 22 commands that control the modem. These 56 commands give the HP-75 the ability to send data to and receive data from the modem, copy files into and out of the electronic disc, and decode two types of bar code labels. The software is divided into four functional areas, described below.

High-Level Modem

The high-level modem software consists of 11 BASIC keywords that provide the ability to turn the modem on and off, transmit strings to and from the modem, and check the status of the modem. Strings sent to the modem can contain data, low-level modem commands, or both.

Low-Level Modem

The low-level modem software consists of 22 commands, all transmitted to the modem via the high-level modem software. These commands directly control such modem operations as dialing, changing modem operating modes, and setting handshake protocol and parity.

Electronic Disc

The electronic disc software consists of 17 BASIC keywords that allow the RAM to simulate a mass storage device. These keywords provide the ability to create, access, and modify text and data files, establish a hierarchical file structure, and copy files into and out of the electronic disc.

Bar Code

The bar code software consists of 6 BASIC keywords that allow the HP-75D to decode Code 39 and Code 11 bar code labels. The commands provide the ability to scan single labels with audible feedback for whether or not the scan was successful. The decoders can automatically verify the check digit of a scanned label or of unscannable labels entered from the keyboard.

INSTALLING THE HP-75 INTO THE HP 82718A

CAUTION

During this installation, you will have to remove the battery pack from the HP-75. You may wish to copy critical files to a mass storage medium before removing the battery pack to avoid the possibility of losing important information if a reset occurs.

You will need only a Phillips or Pozidriv screwdriver to install the HP-75 into the HP 82718A. Here is the installation procedure:

1) Turn off the HP-75. Press [SHIFT][ATTN] or type BYE [RTN].

2) Remove any plug-in module or port cover from the right-most ROM port on the front of the HP-75.

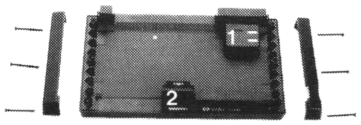
3) Connect the ac adapter/recharger to the HP-75.

4) Take off the HP-75 battery door and remove the battery pack. Save the door, as you will need it if you ever remove the HP-75 from the HP 82718A.

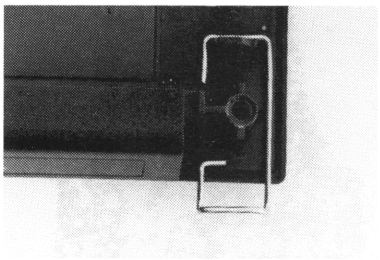
5) Take off the HP 82718A battery door. Insert the HP-75's battery pack into the HP 82718A's battery compartment. Be sure to align the contacts of the battery pack with the spring contacts in the compartment.

6) Replace the HP 82718A battery compartment door and secure its two latches.

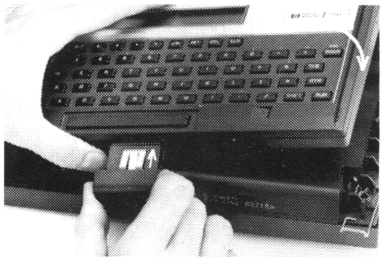
7) Set the HP 82718A right-side up so its nameplate is facing you. The area marked #1 in the photo is the battery box, and #2 is the ROM connector.



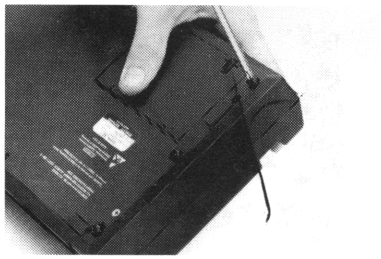
8) Place the carrying clips as shown. Notice that they are both oriented in the same direction. If you do not plan to use the clips, you do not have to install them, or you can place them one rib farther back, and they will be fully concealed by the retainers that hold the HP-75 to the HP 82718A.



9) Plug the HP 82718A ROM connector securely into the right-most port on the HP-75. Then set the HP-75 down onto the battery box. The tab protruding from the bottom of the ROM connector slips inside the slot on the HP 82718A housing.



10) Holding the HP-75 securely to the HP 82718A, attach the two plastic retainers to the unit by screwing in the six long screws provided into the six holes on the bottom of the HP 82718A. Be sure the carrying clips align with the notches on the retainers.



Your HP 82718A Expansion Pod is now ready for use.

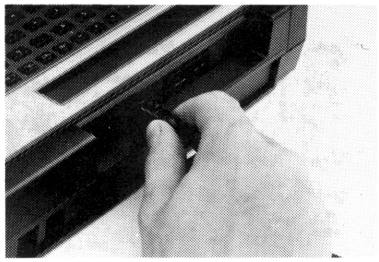
NOTE

When you first turn on the HP-75, you may see either ERROR: modem cold start or ERROR: XMEM INITIALIZEd. These errors only indicate that the battery packs need recharging before they can supply sufficient battery power for the modem or electronic disc.

CONNECTING AND DISCONNECTING THE WAND

1) Turn off the HP-75D. Press [SHIFT] [ATTN] or type BYE [RTN].

2) Insert the wand's plug into the outlet on the back edge of the HP-75D, as shown in the photo at right.



3) To remove the wand, turn off the HP-75D and push down on the button on top of the plug while pulling the plug out.

USING THE BUILT-IN MODEM

The HP 82718A has a built-in modem that provides data communication capability, in the United States, for the HP-75 portable computer. This modem connects directly to the telephone line—a system that is superior to an acoustic coupler connection.

The Federal Communications Commission (FCC) requires specific procedures for installing and operating equipment that uses public communication lines. *Therefore, you should read this section carefully before attempting to connect your modem to the telephone line.*

Notifying the Telephone Company

The HP 82718A is registered with the FCC for direct connection to a standard single-party telephone line; *connection to pay telephones or party lines is prohibited.* If you are unsure about the suitability of your telephone line, contact your telephone company.

Before you connect the HP 82718A modem to the telephone line for the first time, FCC rules require that you notify your telephone company. You should tell them of your intention to connect an FCC-registered device to your telephone line. They need to know the following information:

- Telephone number of the line to which the modem is connected
- Manufacturer: Hewlett-Packard
- Model Number: HP 82718A
- FCC registration number: AU492X-12571-DM-E
- Ringer equivalence number: 0.8 B

The FCC registration number and the ringer equivalence number are located on the bottom face of your HP 82718A.

If you plan to connect the HP 82718A to different telephone lines, furnish the telephone company with a list of these numbers to avoid notifying them every time you move your unit. Notify the telephone company again when the HP 82718A is permanently removed from the line.

Connecting and Disconnecting the Telephone Line

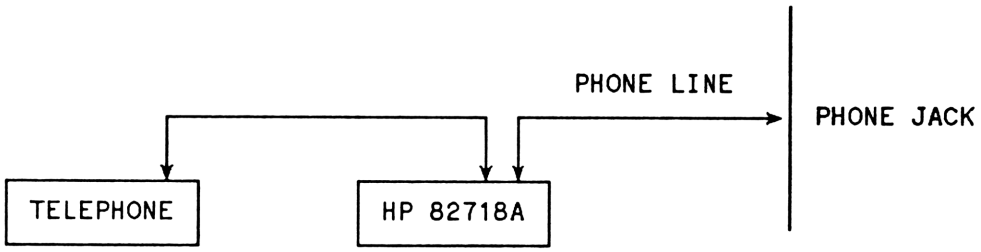
Turn off the modem before plugging a phone line into either modular jack on the HP 82718A. Also turn off the HP-75 unless your application program needs to prompt the user to connect the phone line. The modem can be turned off either by turning off the HP-75, by executing `MODEM OFF`, or by executing `INITIALIZE ":MODM"`.

To connect the modem to the telephone network, you will need a cable with modular telephone plugs on both ends and a modular telephone jack. If your telephone jack has four round holes, you will need an adapter plug that converts your jack from four-prong to modular. Cables and jacks are available from electronics stores or the telephone company.

If you want to plug the HP 82718A into a modular jack into which a telephone set is currently connected (that is, if you want to replace a telephone with the HP 82718A), the telephone should have the same kind of modular jack as the modem. Unplug the cable from the telephone, then plug the loose end of the cable into either of the two jacks on the rear panel of the HP 82718A.

The plugs on the cable can be inserted into the jacks in only one way. The small plastic tab will snap in when the plug is properly connected. When you pull gently on the cable, it should not pull out of the jack until you press the plastic tab.

You may wish to share a single wall jack between a telephone set and the HP 82718A, and use the modem as an auto-dialer for voice communications, or the telephone as a line monitor for data communications. You may do this by purchasing an extra cable with modular plugs on each end from an electronics or telephone supply store. Connect one end of the extra cable to the second jack on the HP 82718A and the other end to the telephone, as shown in the figure. Note that you cannot simultaneously carry on a voice call and use the HP 82718A modem. However, you can listen to the HP 82718A transferring data on the telephone set.



Telephone Company Rights and Responsibilities

The circuitry in your HP 82718A is designed to protect both the phone line and the modem from damage caused by high voltages, and is approved by the Federal Communications Commission. However, the telephone company has the legal right to discontinue service if the HP 82718A should somehow cause harm to the telephone network. In this case, the telephone company will:

- Promptly notify you of the service interruption.
- Give you the opportunity to correct the situation that caused service to stop.
- Inform you of your right to bring a complaint to the FCC concerning the service interruption.

The telephone company may make changes in its facilities and services which may affect the operation of your equipment. However, you will be given adequate notice in writing to allow you to maintain uninterrupted service.

Identifying Problems

If you have problems with normal telephone calls when the HP 82718A is connected to the line (but not in use), you must determine if the HP 82718A is the cause. Disconnect your unit from the phone line. If the trouble continues, contact your telephone company to have your line checked.

If the trouble stops when the HP 82718A is disconnected, you will need to have your unit repaired. Obtaining service is described in appendix A.

Do not attempt to service the HP 82718A yourself. Doing so is a violation of FCC rules and may result in damage to the telephone network.

HOW TO USE THIS MANUAL

This manual is written primarily as a reference for application programmers, and not as a tutorial. The examples, with a heavy emphasis on programmatic use of the commands available in the HP 82718A, assume that you are well-versed in HP-75 BASIC language programming and the operation of the HP-75.

The bulk of the applications written will probably take advantage of the data communications, bar code, and large memory provided by the combination of the HP-75D and HP 82718A. As such, the manual assumes that you understand data communications and the uses of machine-readable (bar code) labels.

The application section is Section 2, "Software Usage and Examples". It provides information about the four software categories, and gives examples of how to use the software.

The main reference sections are Section 3, "Keyword Dictionary", and Section 4, "Low-Level Modem Command Dictionary" and Appendix D, "HP 82718A Errors and Responses". Section 3 contains syntax and descriptions of the high-level modem, electronic disc, and bar code keywords. Section 4 contains syntax and descriptions of the low-level modem commands. Appendix D contains descriptions of all errors and responses reported by the HP 82718A.

We expect that it will take some time to become proficient with the concepts and commands presented here, through a combination of reading the manual and experimenting with the many features of the HP 82718A. Many of the commands can be studied by executing them from the keyboard and observing the results. The low-level modem commands are best studied through the use of a terminal program, as described in Section 2. Once you gain this expertise, you will be able to use the HP-75 as a powerful tool for portable data collection applications.

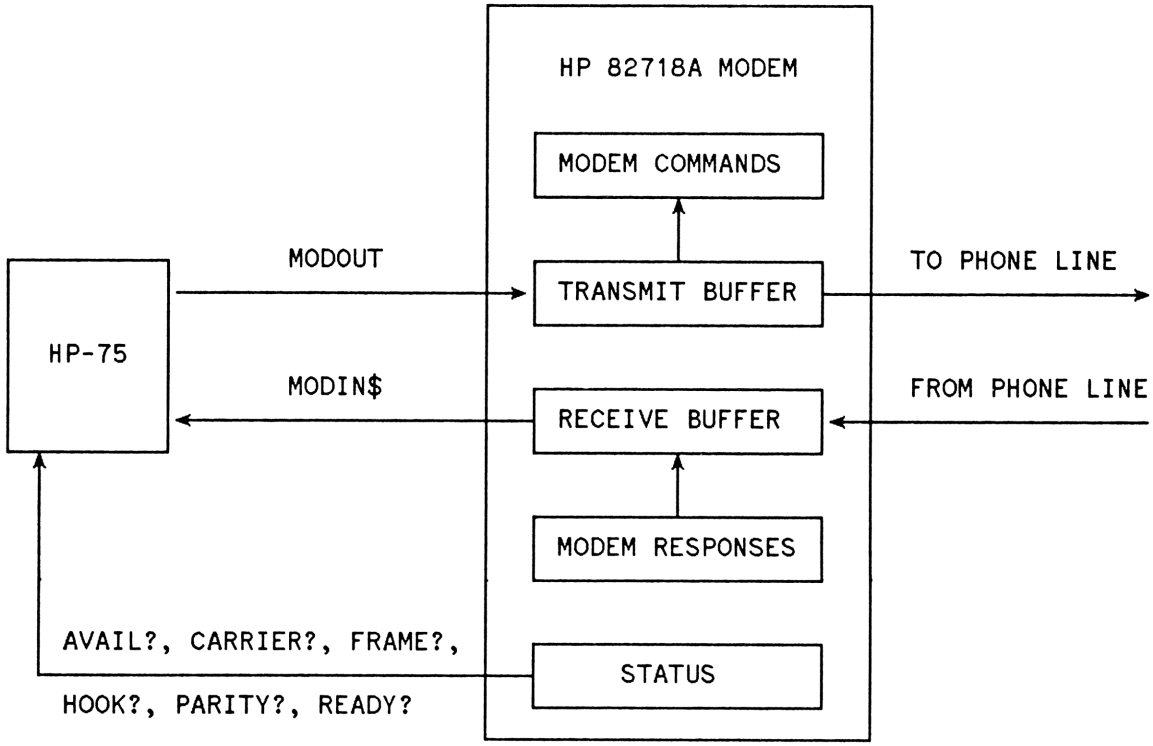
SOFTWARE USAGE AND EXAMPLES

SECTION 2

This section of the manual will provide supplemental information about the operation of the modem, electronic disc, and bar code software, and examples of the use of the software.

OVERVIEW OF MODEM OPERATION

There are two types of modem commands provided by the HP 82718A. *High-level modem commands* are BASIC keywords that add to the set of commands you can execute from the keyboard or put in BASIC programs. Examples of high-level commands are MODEM ON (*turn on the modem*), MODOUT (*send a string to the modem*), and CARRIER? (*check if there is a carrier*). *Low-level modem commands* are commands that tell the modem to perform a specific action. Examples of low-level commands are D (*dial a phone number*), F (*set data format or parity*), and O (*enter ORIGINATE mode*). The HP-75 communicates with the modem primarily with two high-level modem commands: MODOUT and MODIN$. These BASIC keywords are the real workhorses of the HP-75/modem system. MODOUT sends data and low-level modem commands from the HP-75 to the modem. MODIN$ reads the contents of the modem receive buffer and returns that data to the HP-75. In addition, six other high-level modem commands identify certain modem status conditions, such as parity and framing errors, carrier detected, etc. Below is a block diagram of the HP-75/modem system.



When MODOUT sends a string from the HP-75 to the modem, it places the string in the 8-byte modem transmit buffer. The modem then examines the transmit buffer to see what action it should take. If the

string is just data, the data is transmitted over the phone line. If the string contains low-level modem commands, the commands are not transmitted, but are processed locally.

MODIN$ reads the contents of the 64-byte modem receive buffer and returns that data as a string to the HP-75. When data arrives from the phone line, the modem places it in the receive buffer for MODIN$ to read. When low-level commands or data from the HP-75 are echoed by the modem, the modem copies the commands or data from the transmit buffer into the receive buffer for MODIN$ to read. When the modem reports a response (BUSY, CONNECT, etc.), it places the response into the receive buffer for MODIN$ to read.

Adapting Existing Programs to Use the Modem

Existing programs that perform data communications with the HP-75 are designed to work with an HP-IL peripheral, such as the HP 82168A Acoustic Coupler or the HP 82164A HP-IL/RS-232C Interface connected to a modem. The modem provided in the HP 82718A is not an HP-IL data communications peripheral; rather, it is a dedicated internal device, just as the card reader is not an HP-IL mass storage device. Therefore, commands that deal with HP-IL peripherals cannot be used for the modem, and loop configurations and loop protocol are irrelevant to the operation of the modem.

When changing a program that uses HP-IL peripherals to one that uses the modem, there are five areas that need attention:

1) Core input and output commands

2) Device-specific command syntax

3) Status conditions

4) Modem power

5) Additional modem features

CORE INPUT AND OUTPUT COMMANDS. Existing programs use either ENTIO$ and SENDIO from the HPILCMDS LEX file (which exists by itself, in the HP-75 Data Communications Pac, and in the HP-75 I/O ROM), or ENTER and OUTPUT (in the I/O ROM only). You need to replace those core input and output commands with MODIN$ and MODOUT, respectively. Usually, this is a direct, straightforward replacement; for example,

```
D$=ENTIO$(":I1","UNL,TAD#,SDA")
SENDIO ":I1","UNL,LAD#",D$
```

from the HPILCMDS LEX file, or

```
ENTER ":I1" ; D$
OUTPUT ":I1" ; D$
```

from the I/O ROM become

```
D$=MODIN$
MODOUT D$
```

DEVICE-SPECIFIC COMMAND SYNTAX. The device-specific commands that HP-IL devices recognize are different than those used by the modem. The modem has a set of low-level commands that are functionally equivalent to HP-IL remote mode instructions, but use a different syntax. For example, to instruct the HP 82168A Acoustic Coupler to set even parity, ENQ/ACK protocol, and clear its buffers, the following would be done with HPILCMDS:

```
SENDIO ":I1","UNL,REN,LAD#","P0;C1;R;" @ SENDIO "","NRE",""
```

or with the I/O ROM:

```
REMOTE @ OUTPUT ":I1" ; "P0;C1;R;" @ LOCAL
```

The equivalent modem command sequence would be as follows:

```
MODOUT "%F 3,K 203QSQSQ,Z"&CHR$(13)
```

where Q represents control-Q (DC1) and S represents control-S (DC3).

Other remote mode instructions are converted similarly. The discussion of low-level modem command usage later in this section and the "Low-Level Modem Command Dictionary" explain the low-level modem commands. Refer to the owner's manual for the device in question for information about its remote mode instructions. For convenience, here is a table that lists those low-level modem commands which have equivalent remote mode instructions for the HP 82168A and HP 82164A.

Low-Level Modem Command	Remote Mode Instructions HP 82168A	HP 82164A
B (BREAK)	B	B
F (FORMAT)	P	P
K (KONTROL)	C	C,DE,FC,LC,PC,SP
T (TEST)	T	none
Z (ZAP)	R	R

STATUS CONDITIONS. HP-IL peripherals identify status conditions with one or more status bytes. The program takes different actions based on the value of those status bytes. The status bytes are read by requesting status:

```
S$=ENTIO$(":I1","UNL,TAD#,SST")
```

or

```
S$=SPOLL$(":I1")
```

The modem status is identified by certain high-level modem commands, by the low-level Q (QUERY) command, and by other responses that the modem places in the receive buffer. The AVAIL?, CARRIER?, FRAME?, PARITY?, and READY? all have functional equivalents represented by the values of the status bytes. Some of the status information reported by the Q command and the modem responses have equivalents in the status bytes. Refer to the owner's manual for the device in question for information about its status bytes.

The areas of your programs that detect and act on status will have to be modified to use the status reporting mechanisms provided by the modem. Refer to the Sections 3 and 4 and to "Reading Modem Responses" later in this section.

MODEM POWER. You must modify the program to turn the modem on and off using MODEM ON and MODEM OFF. If your program powers down the loop (e.g., SENDIO ":I1","UNL,LAD#,LPD,TL+", ""), you can replace that with MODEM OFF. The proper use of MODEM ON will be discussed in "Turning the Modem On".

ADDITIONAL MODEM FEATURES. The modem offers features that you may want to incorporate into your program. Dialing, answering incoming calls, detection of busy signals, etc. are just some of the new features available to you. You will need to study the features of the modem and decide which of them you want to integrate into your application.

HIGH-LEVEL MODEM SOFTWARE

The high-level modem software provides the ability to turn the modem on and off, transmit strings to and from the modem, and check the status of the modem. This section will illustrate the use of some of the high-level modem commands. (The strings sent to the modem can contain low-level modem commands, which are described in the "Low-Level Command Dictionary" and illustrated later in this section.)

Initializing the Modem

The HP-75 and the modem are initialized independently of each other, just as the HP-75 and an HP-IL data communication device are initialized independently. If the HP-75 is reset ([SHIFT] [CTL] [CLR]), the modem is not. Similarly, if the modem is reset (INITIALIZE ":MODM"), the HP-75 is not.

The modem can be reset to its default state by executing INITIALIZE ":MODM". The modem software resets the modem automatically if there is insufficient battery power in the HP 82718A. Since this reset occurred without your intervention, you are warned about the initialization with ERROR: modem cold start.

Turning the Modem On

Turn the modem on by executing MODEM ON. After you turn it on, you should wait about 5 s for the modem to become ready for use while it performs a self-test. In principle, you can tell if the modem is ready by reading the modem input buffer with MODIN$ until the %OK message is received from the modem:

```
10 MODEM ON
20 IF MODIN$#"%OK" THEN 20
```

However, there are two pitfalls in using this approach:

1) The %OK message will only be reported if the modem is reporting long responses. If the modem is reporting short responses when it is turned on, %4 (%4 followed by a space) is the equivalent of%OK (see appendix D).

2) If modem is already on, MODEM ON has no effect, and no %OK or %4 will be reported.

Here is a routine that will turn the modem on, wait until it is ready for use, and avoid these problems as well:

```
10 MODEM OFF @ MODEM ON ! Turn off the modem in case it is on
17 !
18 ! Look for either %OK or %4
19 !
20 M$=MODIN$ @ IF M$#"%OK" AND M$#"%4 " THEN 20
```

Note that line 20 cannot be the following:

```
20 IF MODIN$#"%OK" AND MODIN$#"%4" THEN 20
```

The first MODIN$ would read the buffer and empty it, comparing the data to %OK. The second MODIN$ would not have any data to read. Consequently, whenever data appeared in the buffer, it would be examined to see if it contained %OK or %4, but not both.

If you know that the modem is reporting long responses (the default behavior), the simplest routine is:

```
10 MODEM OFF @ MODEM ON
20 IF MODIN$#"%OK" THEN 20
```

If short responses are known, substitute %4 for %OK.

This is almost the same as the original "obvious" approach to waiting for the modem to be ready. As you now understand, there are two reasons it works: (1) the modem was off before it was turned on, and (2) the modem is reporting long responses. There are actually two additional reasons it works. These will be discussed under "Reading Modem Responses."

If you always want to force the modem into a known state, you can use the following routine. INITIALIZE ":MODM" takes about 18 s to execute, and then turns the modem off.

```
10 INITIALIZE ":MODM" @ MODEM ON
20 IF MODIN$#"%OK" THEN 20
```

Detecting Connection With a Host Computer

Once the modem is on, and a number has been dialed, the connection with the host computer needs to be detected. The example below uses CARRIER? for this task.

```
10 MODOUT "%d 1(800)555-1212"&CHR$(13) ! Dial the number
17 !
18 ! If carrier not found after 30 s, halt the program
19 !
20 ON TIMER # 1,30 DISP "No connection" @ STOP
30 IF NOT CARRIER? THEN 30 ! Loop waiting for carrier
40 OFF TIMER # 1 ! When carrier found, stop the timer
50 DISP "Connection established"
```

The program dials whatever number is specified on line 10. Note that the parentheses and dash in the dial string improve readability but do not cause an error. The program then loops on line 30 waiting for a carrier to be detected.

A timer is set on line 20 to halt the program after 30 s of waiting for the carrier. The 30 s includes the time it takes the modem to process the dial command and actually perform the dialing. If pulse dialing or

waits are used in the dial string, the time to dial would be longer, and the timeout would have to be adjusted accordingly. An alternative to using a timer is:

```
10 MODOUT "%d 1(800)555-1212"&CHR$(13) ! Dial the number
20 T=TIME ! Initialize time
30 IF CARRIER? THEN 60 ! Continue program if carrier found
40 IF TIME<T+30 THEN 30
50 BEEP 220,1 @ DISP "No connection" @ STOP
60 DISP "Connection established"
```

Without a programmatic timeout of some sort, the program could wait forever for a carrier. Because the HP-75 time resets to zero each midnight, running this program within 30 s of midnight will cause it to loop between lines 30 and 40 forever if no carrier is detected.

Another way to detect carrier uses HOOK? and CARRIER? together. This approach does not need a timer—when the D (DIAL) command gives up, the program will stop.

```
10 MODOUT "%d 1(800)555-1212"&CHR$(13) ! Dial the number
20 IF NOT HOOK? THEN 20 ! Wait until modem is off-hook
25 !
26 ! If off-hook, no carrier, halt program
27 ! If on-hook, waiting for carrier, loop
28 ! If on-hook and carrier detected, continue program
29 !
30 ON 1+HOOK?+CARRIER? GOTO 40,30,50
40 DISP "No connection" @ BEEP 220,1 @ STOP
50 DISP "Connection established"
```

If the dial fails for whatever reason, the modem will go back on-hook. By using HOOK? in conjunction with CARRIER?, the transition from off-hook to off-hook with carrier or from off-hook to on-hook will detect whether or not a connection is made, as shown in the table below.

State	HOOK?	CARRIER?	1+HOOK?+ CARRIER?
On-hook	0	0	1
Off-hook, waiting for carrier	1	0	2
Carrier detected	1	1	3
Carrier not detected (back on-hook)	0	0	1

If the modem is pulse dialing, HOOK? should not be used with CARRIER? to detect connection. Pulse dialing works by actually putting the modem on-hook briefly during every pulse of the digit being dialed. During pulse dialing, either adaptively or explicitly, HOOK? can return a 0 that does not reflect the true status of the modem. If the modem is on-hook during a pulse when HOOK? checks the modem status, HOOK? will return a 0, and 1+HOOK?+CARRIER? will indicate that the carrier was not detected, even though dialing has not even finished.

Reading Modem Responses

Frequently when using the modem, an application program will use MODIN$ to read modem responses (CONNECT, OK, etc.) for two reasons: to establish if a certain condition has occurred, or to remove the modem response from the incoming data stream. A single MODIN$ may not read all the characters of a response at once—you may need more than one MODIN$ to collect the entire response. To collect the response, use a general purpose subroutine such as the one below:

```
1000 M$="" ! Initialize scratch variable
1006 !
1007 ! Look for the string to be found, F$, by accumulating
1008 ! successive MODIN$s and searching for F$.
1009 !
1010 M$=M$&MODIN$ @ IF NOT POS(M$,F$) THEN 1010 ELSE RETURN
```

This subroutine is called by passing it the string to be found in F$. For example, to find %CONNECT, the calling sequence is:

```
10 F$="%CONNECT" @ GOSUB 1000
```

This calling sequence will only determine if a connection has been made. If you want to remove the response from the incoming data stream, you need to know the full sequence of characters for a connection response: CR LF %CONNECT CR LF %OK CR LF (long responses only—see appendix D). The calling sequence then becomes:

```
10 E$=CHR$(13)&CHR$(10) @ F$=E$&"%CONNECT"&E$&"%OK"&E$
20 GOSUB 1000
```

This subroutine can also be used when the modem is turned on by setting F$ to %OK for long responses or %4 short ones. The last two examples under "Turning the Modem On" would become:

```
10 MODEM OFF @ MODEM ON
20 F$="%OK" @ GOSUB 1000
```

and

```
10 INITIALIZE ":MODM" @ MODEM ON
20 F$="%OK" @ GOSUB 1000
```

In that same section ("Turning the Modem On"), there were two reasons listed why line 20 of the previous example can be simplified to IF MODIN$#"%OK" THEN 20. There are two more reasons as well. First, when the modem is turned on, the long response is %OK, not %OK CR LF (see appendix D). Second, only when the modem is turned on, any response (long or short) will be read in a single MODIN$.

Reading modem responses is another means of detecting connection with a host computer. Unlike CARRIER? or HOOK?, which are simple to use but only detect connection versus no connection, modem responses such as BAD ANS, BUSY, NO ANS, and NO DIAL identify the condition that prevented the connection.

You may encounter situations which would be simplified if the responses were not reported. However, you cannot suppress these responses from being sent by the modem—the most you can do is shorten the responses with the L (LONG) command (L 1). Even if you don't need to use the responses, you should be careful to strip them out of the incoming data stream so they won't clutter your data. The exact sequence of response characters reported by the modem appears in appendix D, "HP 82718A Errors and Responses".

MODOUT and Echoed Data

As MODOUT sends data and commands to the modem transmit buffer, some of that data may be echoed back into the receive buffer. Possible conditions for echoing include:

1) If the modem is echoing data (G 1), the modem will copy the data into the receive buffer as the data is transmitted.

2) If the modem is echoing commands (E 0), the modem will copy the commands into the receive buffer as it processes them.

3) Regardless of whether the modem is echoing commands, it will always place responses (either long or short) into the receive buffer.

4) As the modem monitors the status of the phone line, some responses (such as CONN LOST) may be placed in the receive buffer even though they were not initiated by a command.

5) If the host is echoing whatever it receives, the modem will put that echoed data in the receive buffer as soon as it is received.

To prevent the possibility of any of these conditions overflowing the modem receive buffer while transmitting, MODOUT reads the modem receive buffer and saves its contents in a buffer in the HP-75. For example, suppose the string *this is sample data* is sent to the modem, and the host echoes *this is sam* back to the modem while the string is transmitted. As the modem receives *this is sam*, it places the string in its receive buffer. MODOUT removes those characters from the receive buffer as they appear, and saves them in the HP-75. The host continues to echo *ple data* to the modem after MODOUT has ended. When MODIN$ reads the data from the modem, it returns the data saved in the HP-75 by MODOUT (*this is sam*), followed by the current contents of the receive buffer (*ple data*).

MODOUT clears the HP-75 buffer before it starts sending data to the modem and reading echoed data from the modem. The buffer is cleared to prevent echoed data from accumulating and using up all available memory in the HP-75. If the echoed data is important, you should read it immediately. If you do not read it, subsequent MODOUTs will throw it away.

Continuing the above example, suppose *this is another sample* is sent using MODOUT. MODOUT will clear the buffer containing *this is sam*, save *ple data* in the buffer, and transmit *this is another sample*. MODIN$ would then return *ple datathis is another sample* (as long as another MODOUT has not cleared the HP-75 buffer again). The relative amount of data MODIN$ returns from each of the two buffers depends on how much MODOUT put in the HP-75 buffer before it finished transmitting.

MODOUT and Out of Memory

As MODOUT places echoed data into the HP-75, MODOUT will use as much memory as it needs, until there is no more memory available. As long as it has data to transmit, MODOUT will continue to read data from the receive buffer to prevent it from overflowing. However, all the echoed data that is read after the HP-75 runs out of memory is thrown away, since there is no place to put it.

In the previous example, if the HP-75 ran out of memory after *thi*, *s is sam* would be lost. The next MODOUT would empty the buffer, and would read *ple* before running out of memory and losing more echoed data. If echoed data is important to you, make sure you have sufficient memory available for it, and read it as soon as it appears.

If you need to know how much data is saved by MODOUT, save the available memory (use MEM) before executing MODOUT, and compare that with the available memory after executing MODOUT (e.g., M=MEM @ MODOUT D$ @ DISP M-MEM;"bytes saved by MODOUT")

Retransmitting Received Data

MODIN$ reads the data from the modem receive buffer, and copies it into the HP-75 buffer behind any data that may already be there from MODOUT. Because MODOUT clears this buffer before it does anything else, the statement MODOUT MODIN$ will not retransmit the data just received by the modem—it will retransmit random, unknown data. If you need to retransmit received data, you could use M$=MODIN$ @ MODOUT M$, but this requires M$ to be dimensioned large enough to hold the input string. A better solution is MODOUT MODIN$&"", which requires no variables to be dimensioned.

If you use MODOUT MODIN$&"" to echo received data back to the host, you should be aware of the possibility of getting into an infinite loop. This may occur because more than just data appears in the receive buffer—responses also appear, and those responses can set up an infinite loop. For example, if the %CONNECT response is retransmitted, the modem will interpret it as %C, the COUNT command. The modem expects to see %C, a space, a number, and a carriage return.

Since %CONNECT followed by carriage return is not a valid syntax for the C command, the %ABORT response would be reported by the modem, and placed in the receive buffer. Upon retransmission, that response would be interpreted as %A, the ANSWER command. It would be processed correctly, but its response could be %BAD ANS, causing %B, the BREAK command, to be retransmitted with invalid syntax, resulting in %ABORT.

The situation soon becomes unwieldy. When you retransmit received data, be sure to strip out all characters of modem responses. To facilitate this, the exact sequence of response characters is given in appendix D.

Terminal Program

Any terminal program, when stripped to its bare essentials, must perform three operations:

1) Read input from the keyboard

2) Transmit the keyboard input to the host

3) Display data received from the host

Here is the simplest terminal program that does all three operations:

```
10 MODEM ON
20 MODOUT KEY$ @ DISP MODIN$; @ GOTO 20
```

(Assumptions that allow this to work are: the modem is in its default state with command echoing enabled (E 0) and data echoing disabled (G 0), WIDTH is INF to avoid spurious CR LFs, and DELAY is zero to avoid long delays after each DISP is executed.)

Line 10 turns on the modem. If the modem is on, line 10 has no effect. MODOUT KEY$ gets the ASCII character for any key pressed and sends the character to the modem. The character is put in the modem transmit buffer. If the character is the command character, the subsequent characters typed will be treated as part of a low-level modem command, until the command is terminated by a carriage return.

DISP MODIN$ displays anything in the modem receive buffer. The semicolon (;) inhibits the CR LF after the DISP statement; without it, each block of characters read by MODIN$ would appear on a separate line. GOTO 20 repeats this sequence forever.

There are three things that can appear in the receive buffer: 1) modem responses, 2) echoed low-level commands, and 3) incoming or echoed data. Responses always appear in the receive buffer. They can only be shortened using the L (LONG) command—they cannot be suppressed entirely. Low-level commands can be echoed either by the modem or by the terminal program. The host cannot echo the commands because they are processed locally by the modem and not transmitted to the host. If command echoing has been enabled (E 0), the modem will do the echoing. If command echoing has been disabled (E 1), the terminal program must do the echoing by using DISP K$ to display everything typed:

```
10 MODEM ON @ MODOUT "%E 1"&CHR$(13)
20 K$=KEY$ @ MODOUT K$ @ DISP K$;MODIN$; @ GOTO 20
```

Data can be echoed by either the host computer, the modem, or the program. If data echoing has been disabled (G 0), the host is expected to echo the data. If data echoing has been enabled (G 1), the modem will echo the data, regardless of the action of the host. If neither the host nor the modem is echoing the data, the program must do so (DISP K$).

These different approaches to command and data echoing are summarized as:

Host echoes	Modem echoes	Program echoes	Modem parameters	Terminal program
data	commands	nothing	G 0,E 0	DISP MODIN$;
nothing	commands and data	nothing	G 1,E 0	DISP MODIN$;
nothing	nothing	commands and data	G 0,E 1	DISP K$;MODIN$;

These three approaches are the simplest to use in a terminal program. A more difficult approach would have the program echo only data or only commands. For this, the program would have to examine everything read from the keyboard. Since a low-level command starts with the command character (usually %) and ends with a carriage return, those characters would be used to determine when to start and stop echoing by the program. If the program echoes only the data, then the modem must echo the commands (E 0). If the program echoes only the commands (E 1), then either the host (G 0) or the modem (G 1) can echo the data.

You can extend this simple terminal program in dozens of ways. It could buffer up all keyboard input, allow editing of it, screen out editing characters (such as backspace), and send it all to the modem when [RTN] is pressed. It could recognize special characters that would drive automatic logon and logoff sequences, or begin file transfers. It could work in conjunction with an autodialer program that looks up and dials phone numbers based on the name of the party being called.

The simplest terminal program is explained here because it is an invaluable learning tool for understanding how to use the HP 82718A modem when used in conjunction with a second HP 82718A. This procedure is described in "Connecting Two Pods Back-to-Back" later in this section.

LOW-LEVEL MODEM SOFTWARE

The low-level modem software provides the ability to directly control the modem for operations such as dialing, changing modem operating modes, and setting handshake protocol and parity. All low-level commands are sent to the modem using the high-level MODOUT command. This section will describe the operation of the modem and examples of the use of its commands. The examples assume that you program has turned the modem on (see "Turning the Modem On" earlier in this section).

Modem Modes

The modem has four primary operating modes that directly affect its operation. These are described below:

Mode	Description
MODEM	MODEM mode is the data communication mode of the modem. In MODEM mode, the modem is waiting for either a dialing operation or a ringing signal. If a dialing operation occurs, the modem is originating the communication session. It will then monitor the phone line for a ring back from the remote modem, and attempt to enter ORIGINATE mode. If a ringing signal occurs, a remote modem is trying to initiate a communication session. The modem will then attempt to enter ANSWER mode and answer the incoming call.
VOICE	VOICE mode allows you to use the modem as an auto-dialer for voice communication. The modem is not monitoring the status of the phone line for either a ring by a remote phone or for incoming calls, and performs none of the data communication activity that occurs in MODEM mode.
ORIGINATE	The modem is issuing an originate tone or waiting for an answer tone.
ANSWER	The modem is issuing an answer tone.

The modem can also be off-hook or on-hook, analogous to a telephone. When a phone handset is in use, it is off the hook—using the phone line. When a phone handset is hung up, it is on the hook—not using the phone line. Similarly, when the modem is off-hook, it is using the phone line, and when the modem is on-hook, it is not using the phone line.

When the modem is on-hook, it can be either in MODEM mode, waiting for a dialing command or an incoming call, or in VOICE mode, not monitoring the phone line. When the modem is off-hook, it can be in ORIGINATE mode, waiting for a ring back from a remote phone or an answer tone, in ANSWER mode, waiting for an originate tone, or in VOICE mode, not monitoring the phone line.

Virtually all high-level modem commands are geared toward use with MODEM mode. Detecting carrier and on-hook/off-hook status (CARRIER? and HOOK?) are relevant to establishing a communication session. Reading from and writing to the modem buffers (MODIN$ and MODOUT) is relevant to VOICE mode only because MODOUT sends the commands and MODIN$ reports the responses to the commands.

Checking the buffer status (AVAIL? and READY?) and looking for data errors (FRAME? and PARITY?) are only pertinent to data transmission. INITIALIZE ":MODM" forces the modem into MODEM mode.

Most low-level commands are geared toward MODEM mode as well. The three that explicitly support VOICE mode are V (VOICE—puts the modem into VOICE mode), P (PICKUP—takes the modem off-hook and enters VOICE mode), and X (XMIT—dial a phone number without call progress reporting after dialing). These are described in the "Low-Level Modem Command Dictionary". All the low-level commands are listed below with a brief description of their behavior as it pertains to MODEM and VOICE modes.

Low-Level Command	Behavior Relative to MODEM and VOICE Modes
A (ANSWER)	Forces MODEM mode
B (BREAK)	Only occurs if carrier present
C (COUNT)	Ignored in VOICE mode
D (DIAL)	Line monitoring for carrier and ring only relevant in MODEM mode
E (ECHO)	Command echoing can occur in MODEM and VOICE modes
F (FORMAT)	Parity only relevant during data transmission
G (GIVEBACK)	Data echoing only relevant during data transmission
H (HANGUP)	Hangs up modem in MODEM and VOICE modes
I (INITIALIZE)	Forces MODEM mode
K (KONTROL)	Handshaking only relevant during data transmission
L (LONG)	Responses can occur in MODEM and VOICE modes
M (MODEM)	Forces MODEM mode
N (NEW)	Commands can be sent in MODEM and VOICE modes
O (ORIGINATE)	Forces MODEM mode
P (PICKUP)	Forces VOICE mode
Q (QUERY)	Status can be requested in MODEM and VOICE modes
R (REDIAL)	Line monitoring for carrier and ring only relevant in MODEM mode
T (TEST)	Tests can be performed in MODEM and VOICE modes
U (UNLISTEN)	Commands as data only relevant during data transmission
V (VOICE)	Forces VOICE mode
X (XMIT)	Dials numbers in VOICE mode only
Z (ZAP)	Buffer can be cleared in MODEM and VOICE modes

Modem Parameters

In addition to the modes just described, the modem has several parameters that affect operation and that can be altered by low-level commands. The parameters and the low-level commands that set them are shown in the following table:

Parameter	Command
Ring count	C (COUNT)
Command echoing	E (ECHO)
Parity and data length	F (FORMAT)
Data echoing	G (GIVEBACK)
Handshake protocol	K (KONTROL)
Response length	L (LONG)
Command character	N (NEW)

The current values of these parameters are preserved by the modem when it is off, except for the command character, which is always set to % when the modem is turned on.

Default State

The table below describes the default state of the modem. The first column is the parameter or state of the modem. The second column shows the default state that is set when the modem is initialized. The third column is the equivalent low-level command that would have to be executed to produce the same effect on the modem. The last column shows which parameters and states are preserved while the modem is turned off. Those that are not preserved revert to their default values when the modem is turned on.

Parameter or Mode	Default Condition	Equivalent Low-Level Command	Preserved While Off
Ring count	0	C 0	yes
Command echoing	yes	E 0	yes
Parity	none	F 4	yes
Data echoing	no	G 0	yes
Modem hung up	yes	H	no
Handshake protocol	Transmitter XON/XOFF		
	Receiver XON/XOFF		
	Receiver ENQ/ACK		
Parity stripped	no	K E03QSQSQ	yes
Chars stripped	none		
Block size	32 bytes		
Handshake chars	QSQSQ		
Response length	long	L 0	yes
MODEM or VOICE mode	MODEM	M	yes
Command character	%	N %	no
Transmit buffer	cleared	none	no
Receive buffer	cleared	Z	no
Cmd char is data (UNLISTEN)	no	none	no

Note: Q represents control-Q (DC1) and S represents control-S (DC3)

Modem Responses

Modem responses identify certain conditions of modem operation. All modem responses, which are placed in the modem receive buffer for MODIN# to read, are preceded by the command character to allow you to distinguish between the responses and incoming data. The L (LONG) command specifies whether the responses are to be long or short. Long responses are spelled out, and short responses are a single hex digit. The responses, in the two lengths, are:

Long Response	Short Response
ABORT	A
BAD ANS	B
BUSY	0
CONNECT	1
CONN LOST	8
FAIL	9
NO ANS	2
NO DIAL	3
OK	4
OV	OV
RING	5
RING IN	7

See appendix D, "HP 82718A Errors and Responses", for the meanings of the responses, commands that report them, and the exact sequence of characters that will be reported.

While you are developing and debugging applications using the modem, the long responses are useful to help monitor what is going on, and are self-documenting as well. In the final version of an application, short responses are preferable. Data from the phone line and responses generated by the modem cannot be put in the modem receive buffer simultaneously. When a response is placed in the buffer, the incoming data flow is halted for the time needed to put it there. There is a possibility of data loss because of this, depending on how fast the host is transmitting. Short responses reduce this time interval and the possibility of data being lost; in addition, the program length will decrease slightly, and the program will be able to process received data slightly faster.

Low-Level Modem Command Syntax

All low-level modem commands begin with a command character so the modem can distinguish between data to be transmitted and commands to be processed. Also, when the modem reports a response, it puts a command character before the response so you can distinguish between responses and incoming data.

The default command character is %, but you can change it to something different using the N (NEW) command. When the modem receives the command character from the HP-75, it interprets all characters received between the command character and a carriage return as commands. The general syntax of a low-level modem command is as follows: the command character, any number of commands (together with their parameters) separated by commas, and a carriage return. The carriage return may be preceded by a comma if the comma is preceded by a recognized command. This syntax is shown below; CM symbolizes the command character (usually %), and CR the carriage return.

CM *command*[,*command*, ... ,*command*] CR

or

CM *command*[,*command*, ... ,*command*], CR

For a single command, this simplifies to CM *command* CR. Some examples are shown below:

```
%G 0 CR
%E 1 CR
%F 3 CR      are equivalent to   { %G 0,E 1,F 3,A CR
%A CR                            { %G 0,E 1,F 3,A,CR
```

Two characters will abort the command string if they appear anywhere between the CM and the CR: BS (backspace—ASCII 8 or control-H) and CAN (cancel—ASCII 24 or control-X). All characters after the BS or CAN and before the CR will be ignored (neither executed nor transmitted).

NOTE

If you incorrectly type a low-level modem command using a terminal program, the modem will respond with \ %ABORT followed by %OK (or \ %A followed by %4 for short responses). The modem will continue to treat all subsequent characters as part of the aborted command until it receives a carriage return. Therefore, if you type a low-level command incorrectly, press [RTN] before trying to repeat the command.

Each command consists of the command name followed by any parameters. The exact syntax of each command is listed in the "Low-Level Modem Command Dictionary". Commands can be upper- or lowercase. If a command requires a parameter, all characters from the first character of the command name up to the first space are ignored. For example, several equivalent representations of the G (GIVEBACK) command are:

```
G 0
GI 0
giveback 0
GivethedatabacktotheHP-75 0
```

If you need to send the command character as data, see "Sending Command Characters as Data" later in this section.

Software Handshake Protocol

Software handshakes are used for controlling the flow of information between the modem and another computer. They prevent problems that could arise from the increased use of time-sharing and multitasking on computer networks. If the remote computer is temporarily busy, it needs a method to

notify other devices on the network that they need to wait. Software handshakes can also prevent such problems as data being lost because the host is busy or a buffer is overflowing.

The K (KONTROL) command allows you to define the software handshake protocols for the modem. It expects eight single-byte parameters identified as K1-K8. Protocols are specified using the K command, as described in the "Low-Level Command Dictionary", and are defined in this section.

ENQ/ACK PROTOCOL. ENQ/ACK protocol is used by the primary transmitting device (host) to control the transfer of data to the receiving device (terminal). Typically, the host device is a computer; the terminal device is a remote terminal or peripheral. ENQ/ACK protocol is used for batch, or block, information transmission with either full or half-duplex devices.

If the host is ready to send data to the terminal, the host sends an ENQ (*enquire*) to check if the terminal is ready to receive a block of data. When the terminal is ready, it sends an ACK (*acknowledge*) to the host, and the host transfers one block of data and another ENQ. This cycle is repeated as required by the host. ENQ is called the *request* character, and ACK is called the *answer* character.

The HP 82718A modem can be a host by using Transmitter ENQ/ACK Protocol (bit 0 of K1 set), or a terminal by using Receiver ENQ/ACK Protocol (bit 1 of K1 set). The request and answer characters are fixed to be ENQ (ASCII 5) and ACK (ASCII 6).

XON/XOFF PROTOCOL. XON/XOFF protocol is used by the receiving device to tell the transmitting device when to start and stop sending. XON/XOFF protocol is mainly for character rather than block transmission; however, block transmission can be used if desired. XON/XOFF protocol requires full-duplex devices and operation. Both devices can be using XON/XOFF protocol to control data that they're receiving.

With XON/XOFF protocol, the receiving device monitors its receive buffer. When the buffer is close to being full (or close to not being able to accept another block), the receiving device sends an XOFF (*transmit off*) to tell the sending device to stop sending. The sending device then suspends data transmission until the receiving device sends an XON (*transmit on*). The sending device then resumes sending, starting with the character that was next to be sent when the XOFF was received. XON (ASCII 17, or DC1) is called the *ready* character, and XOFF (ASCII 19, or DC3) is called the *not ready* character.

The HP 82718A modem can be the primary transmitter by using Transmitter XON/XOFF Protocol (bit 3 of K1 set), or the primary receiver by using Receiver XON/XOFF Protocol (bit 2 of K1 set). When a transmitter, it uses K4 as the ready character and K5 as the not ready character. When a receiver, it uses K6 as the ready character and K7 as the not ready character. You can define these protocol characters because some computer systems (e.g., Dow Jones News/Retrieval Service) reverse the sense of the ready and not ready characters.

PROMPT MODE. The host is presumed to have infinite input capability, so that the terminal can send data to the host without using any software handshakes and without observing the block size—that is, the terminal can send any amount of data to the host any time it wants. Because this situation is not always practical, a host may control this type of transfer using a *prompt* character. Whenever the host is ready to accept data, it sends a prompt character to the terminal. The terminal then sends one line of data to the host. (A line of data is terminated by a carriage return—its length is not related to the block size.) The host sends another prompt when it's ready for another line.

If the modem is acting as a host (Transmitter ENQ/ACK Protocol), it never uses a prompt character. That is, the modem doesn't automatically prompt the external device for data. However, you can program the

HP-75 to send an appropriate prompt character to the external device via the modem. For example, the HP-75 can check the receive buffer for a carriage return while it is reading in data with MODIN‡, and then send the prompt when it is through processing the line.

If the modem is using Prompt Mode (bit 3 of K3 set), the prompt character is defined by K4. The modem stops sending data to the host whenever it sends a carriage return. If it finds a K4 character, and K4 is not the same as K5, the modem will also strip off any K5 characters received between the carriage return and the K4 character. If the modem is not set to use a prompt character (bit 3 of K3 clear), then it doesn't wait for a prompt before sending data and doesn't stop sending data after a carriage return. Prompt Mode has higher priority than Transmitter XON/XOFF Protocol.

COMBINED PROTOCOLS. Some devices use both ENQ/ACK and XON/XOFF protocol. This enables them to initiate the transfer of data according to ENQ/ACK protocol, and to stop and start the transfer as required by the receiver.

When the HP 82718A modem uses ENQ/ACK and XON/XOFF protocols together, it operates according to both of these protocols as described above, with one exception. Prompt Mode and Transmitter XON/XOFF Protocols cannot be used simultaneously because they both use the K4 and K5 characters. Therefore, if the modem is using Transmitter XON/XOFF Protocol, Prompt Mode should be disabled—if both protocols are enabled, Prompt Mode has priority.

Buffer Control

The modem monitors the number of data bytes in its receive buffer to determine when to halt transmission from the remote device by sending the appropriate software handshake character. By not allowing the external device to send data until there is enough room in the buffer for the next block of data, the modem can prevent the loss of data.

For ENQ/ACK protocol, the host normally sends data to the terminal in blocks followed by an ENQ. The modem's receive buffer can hold up to 64 bytes. If the modem is using Receiver ENQ/ACK Protocol, the modem will not send an ACK until MODIN‡ empties the receive buffer.

If the modem is using Transmitter ENQ/ACK Protocol, then the external device will be expecting to receive data in blocks that it can handle. If the external device is expecting a block of data of 40 characters, for example, then the modem will send its data in blocks of 40 characters, each followed by an ENQ. The modem will not send out another block until it receives an ACK.

The block size used by the modem with Transmitter ENQ/ACK Protocol is 8 to 64 bytes, in 8-byte increments. The block size is specified in bits 2-0 of K3. Refer to the owner's manual of the external device to determine the block size required.

For XON/XOFF protocol, the device that is sending data will continue to send data until it receives an XOFF. The device may send a block of data before it can actually stop sending. Thus it is important to reserve a block of empty bytes in the receive buffer that is large enough to accept the maximum block that the external device may continue sending upon receiving an XOFF. If the receive buffer overflows, received data is ignored until the number of bytes in the receive buffer drops below 56. Then the buffer overflow response (OV) is placed in the buffer, and receiving continues.

When the modem is using Receiver XON/XOFF Protocol, it counts the number of bytes in its receive buffer. When it has received a number of bytes equal to the block size, the modem sends an XOFF (actually, a K7 character). An XON (actually, a K6 character) is sent after MODIN‡ empties the receive buffer.

The block size used by the modem with Receiver XON/XOFF Protocol is 8 to 64 bytes, in 8-byte increments. If the external device may send 16 characters after receiving an XOFF, for example, then the block size can be any value of 64-16=48 bytes or less without risking loss of data. The block size is specified in bits 2-0 of K3.

When the modem is using Transmitter XON/XOFF Protocol, it will send a maximum of 3 characters after it receives an XOFF (actually, a K5 character).

Modem Deadlock

Modem deadlock occurs when the modem transmit buffer is full, but the modem cannot transmit the data out the phone line to the host computer. If the transmit buffer is full, the HP-75 cannot send any commands or data to the modem, since everything sent to the modem by MODOUT is placed in the transmit buffer for processing. The modem cannot send the data and thus empty the buffer, and cannot accept anything from the HP-75 until the buffer is empty, hence the term deadlock.

The situations that can cause deadlock are as follows:

1) The host computer is not ready to receive data. This could be because of multitasking, time-sharing, slow data communications software, or any of a variety of reasons.

2) The modem is using a handshake protocol that disables data transmissions until a certain character is received from the host computer (or terminal if the HP-75 is the host). If the host does not send the character, data transmissions will not be resumed. The applicable protocols are Transmitter XON/XOFF, Transmitter ENQ/ACK, and Prompt Mode.

MODOUT waits until the transmit buffer is ready to accept data before sending anything to the modem. If the modem is deadlocked, MODOUT will wait forever. Once MODOUT starts to wait, you can halt it by pressing [ATTN] (a low battery condition in the HP-75 will halt it as well). You can then only use high-level modem commands to clear a deadlock. INITIALIZE ":MODM" will reset the modem to its default state, thereby clearing the deadlock. MODEM OFF or turning the HP-75 off will end the data transmission; when the modem is turned back on, the deadlock condition will be cleared.

Because the transmit buffer is full, no low-level modem commands can be used to clear a deadlock condition. In particular, I (INITIALIZE) will not reset the modem to its default state, K (KONTROL) will not change to a different handshake protocol, and T (TEST) will not perform the power-on test and clear the buffers.

If you anticipate a possible deadlock, you should plan for this in your program. The READY? high-level modem command returns a 1 if the modem transmit buffer is ready to receive data from the HP-75, or a 0 if the modem is not ready, and thus identifies deadlock. If deadlock may occur temporarily because of delays by the host, your program could just wait for the deadlock condition to go away (READY?=1). If deadlock may occur because of handshake protocol, your program may only be able to use INITIALIZE ":MODM" or MODEM OFF to clear the condition. If deadlock is of great concern, your program should only send one character at a time to the modem with MODOUT, waiting before each character is sent until READY? =1.

Adaptive Dialing

Telephone systems in the United States accommodate one (or both) of two alternate dialing schemes. The older scheme, used for rotary-dial phones, is called interrupt or pulse dialing. The newer scheme, used for

push-button phones, is called Touch-Tone* dialing. In this manual, we will refer to these two dialing schemes as *pulse* or *tone* dialing.

Applications using the HP 82718A modem may involve transmitting or receiving data from different locations using different phone systems. A general purpose data communications program should be able to adapt automatically to the the characteristics of these phone systems.

Adaptive dialing is the process of identifying whether the phone system can accommodate tones, and using them if it can, or using pulses if it cannot. Adaptive dialing can be done in two ways: either the modem can do the adapting using the D (DIAL) command, or the data communications program itself can do the adapting using modem responses.

Adaptive Dialing Using the DIAL Command

When dialing with the D command, you can specify tone dialing, pulse dialing, or adaptive dialing. When the modem dials adaptively, the following occurs:

1) The first digit of the dial string is tone dialed.

2) If the modem hears no dial tone, all the remaining digits of the dial string are tone dialed as well.

3) If the modem does hear a dial tone, it assumes the phone system cannot accommodate tone dialing, so it pulse dials the entire number, including the unsuccessful first digit.

The difficulty in using this scheme with many phone systems arises when access codes are present. *Access codes* are the number or numbers you must dial from a phone to gain access to the outside phone system. Business PBXs or switchboards typically require a single digit to be dialed. Before you can dial the number you want, you must wait for a new dial tone. Some phone systems require a sequence of access codes, with a new dial tone appearing after each access code is dialed.

Access codes cause problems when the modem adaptively dials because the dial tone is used to determine whether or not tones were recognized. Suppose the access code for a business is a 9, and the phone number to be dialed is 1(800)555-1212. To dial this adaptively, you would use the following command (notice that the parentheses and dash do not cause an error in the dial string—they just improve readability):

```
MODOUT "%D 9 1(800)555-1212"&CHR$(13)
```

Following the procedure above, the modem tone dials the 9. If it hears a dial tone, it will assume that the phone system cannot accommodate tones, so the entire number will be redialed using pulses. Since the 9 actually just attained access, the actual number dialed is 918-0055 (the phone system will ignore the 51212).

The time between dialing an access code and hearing the new dial tone varies greatly. Because of this variability from phone system to phone system (and even within a particular phone system), you cannot predict if the modem will interpret the second dial tone as meaning tones not allowed. Consequently, you cannot count on adaptive dialing using the D command to work in the desired manner with access codes.

---

*Touch-Tone is a registered trademark of American Telephone and Telegraph.

If the phone systems you will be using do not have access codes, this approach to adaptive dialing will work fine. The modem would then switch to pulse dialing only if the phone system could not accommodate tones. If access codes will be present in your phone systems, adaptive dialing should be done using modem responses.

Adaptive Dialing Using Modem Responses

For adaptive dialing purposes, the significant aspects of the D command can be summarized as follows: P forces adaptive dialing as just described, W forces tone dialing, and I forces pulse dialing. This means you can disable the adaptive dialing behavior of the D command by using W or I at the beginning of each dial string.

If you use W to force tone dialing, the modem will successfully dial both access codes and phone numbers in most parts of the United States. However, the problem of dialing on those phone systems which cannot accommodate tones still exists. The program itself must then adapt to the phone system by examining the modem responses.

If you use I to force pulse dialing, the modem will successfully dial both access codes and phone numbers nearly everywhere, since virtually all phone systems accommodate pulse dialing. However, pulse dialing lengthens dialing time, and requires that the user not use a telephone as a line monitor until the dialing is completed (see "Using a Telephone as a Line Monitor" later in this section).

If you use I to pulse dial the access codes, and W to tone dial the phone number, the modem will not misinterpret the second dial tone as meaning tones not allowed. The modem can then dial both access codes and phone numbers successfully (assuming the phone system can accommodate tones), and dialing time will be short. However, some phone systems do not allow using both tone and pulse when dialing access codes and phone numbers. Consequently, if you have access codes, tone and pulse dialing should not be mixed unless you know the phone system being used allows it.

A general approach to adaptive dialing that avoids all these problems and minimizes dialing time is as follows: Dial the phone number using only tones. If the modem responses indicate that tone dialing did not work, redial the number using only pulses. This mimics the behavior of the D command, and allows the use of access codes without dialing wrong phone numbers.

The BAD ANS and NO ANS modem responses are used to identify the phone system characteristics. BAD ANS occurs under two conditions: (1) the modem hears a dial tone after dialing, or (2) the called phone responds with a bad answer tone. Condition (1) means that the tones were not recognized by the phone system, so pulse dialing must be used. Condition (2) means that the tones or pulses were recognized by the phone system, but the answering modem is malfunctioning.

NO ANS also occurs under two conditions: (1) the modem hears no dial tone or an unsteady tone after dialing, or (2) the called phone actually did ring, but there was no answer after either the maximum number of rings (the ring count + 4, up to a limit of nine) or after 50 s have elapsed since dialing ended. Condition (1) means that the tones were not recognized by the phone system, but the dial tone has stopped, and the line is either dead or issuing the receiver off-hook sound. Condition (2) means that the tones or pulses were recognized by the phone system, but the answering modem is unavailable.

When either BAD ANS or NO ANS are reported, the distinguishing factor between conditions (1) and (2) is whether the modem reported RING first. BAD ANS (or NO ANS) by itself means condition (1), and one or more RINGs followed by BAD ANS (or NO ANS) means condition (2).

If the program finds either BAD ANS or NO ANS while it is tone dialing, it should redial using pulses if condition (1) occurs. If condition (2) occurs, it is a condition beyond the control of the program that

switching to pulse dialing will not correct. The program should ask the user what action to take if condition (2) occurs. If the program finds either BAD ANS or NO ANS while it is pulse dialing, it should ask the user what action to take, regardless of condition.

Dial Strings for Adaptive Dialing

Here is a more detailed explanation of the special characters that can be used in a dial string:

1) The special character P means wait for dial tone, then dial the next digits adaptively. If no dial tone is heard within 3-4.5 s, the NO DIAL response is reported. The first digit (*only* the first digit) after the P will be dialed using tones. If no dial tone is heard, the remaining digits will be dialed using tones. If a dial tone is heard, the entire string after the P will be dialed using pulses.

   Normally, the D command places an implied P at the beginning of the dial string. This causes any dial string to be dialed adaptively. If P is the first character of a dial string, it will override the implied P and force adaptive dialing. The modem will wait 3 s for a dial tone for the implied P, and 4.5 s for a P in the dial string.

2) The special character W means wait 5 s, then dial the next digits using tones (until the next special character is encountered). If W is the first character of a dial string, it will override the implied P and force tone dialing.

3) The special character I means dial the next digits using pulses (until the next special character is encountered). The I does not cause a wait. If I is the first character of a dial string, it will *not* override the implied P—it will force pulse dialing after the implied P.

Examples of different dial strings are illustrated below.

TONE DIAL STRING. A dial string that dials a phone number using only tones starts with W and has W between the access codes:

```
W 9 W 1(800)555-1212
↑ ↑ ↑
└─┼─┼─Wait 5 s to attain access, then force tone dialing
  └─┼─Dial access code
    └─Override implied P, wait 5 s, then force tone dialing
```

With this dial string, the modem will not report the NO DIAL response. The implied P waits for a dial tone and reports NO DIAL if one is not heard, but the W overrides the implied P. The modem will eventually report NO ANS, but it takes 50 s to report this condition. To allow NO DIAL to be reported earlier (within 5 s), build the tone dial string starting with PW and placing W between the access codes:

```
P W 9 W 1(800)555-1212
└─Override implied P, wait 4.5 s for dial tone, report NO DIAL if none, then force adaptive dialing
  └─Wait 5 s, then force tone dialing
    └─Dial access code
      └─Wait 5 s to attain access, then force tone dialing
```

The penalty of this approach is that it adds 4.5 s before the access code will be dialed.

PULSE DIAL STRING. A dial string that dials a phone number using only pulses starts with PI (the P is so NO DIAL will be reported) and has WI between the access codes:

```
P I 9 W I 1(800)555-1212
└─Override implied P, wait 4.5 s for dial tone, report NO DIAL if none, then force adaptive dialing
  └─Force pulse dialing
    └─Dial access code
      └─Wait 5 s to attain access, then force tone dialing
        └─Force pulse dialing
```

The PI could actually be simplified to I. This uses the implied P instead of the explicit P, since the I does not override the implied P the way W or P do. The only reason to use the implied P is that it is 1.5 s shorter than the P, and thus reduces the total dial time by that amount.

TOTAL DIALING TIME. The total time to tone dial PW9W1(800)555-1212 is 16.3 s. The total time to pulse dial PI9WI1(800)555-1212 is 25 s. By using the implied P, these would be reduced to 14.8 and 23.5 s. These times are calculated using worst-case times for the operation of the dialer in the modem, as follows:

Tone Dialing		Pulse Dialing	
Tone duration	65-75 ms	Pulse rate	10-11 pulse/s
Intertone delay	65-75 ms	Interdigit delay	700-800 ms

Using a Telephone as a Line Monitor

You can connect a telephone to the second modular jack of the HP 82718A and use it as a line monitor to listen to dialing and data communication activity. If the modem must dial using pulses, the user of the program cannot listen to a phone plugged into the second modular jack until after the dialing has been completed. This is because of the way pulse dialing works.

Pulse dialing actually puts the modem back on-hook briefly during every pulse of the digit being dialed. For example, if a 7 is pulse dialed, seven pulses will be heard, and the modem goes on-hook seven times. A telephone connected to the second modular jack is connected in parallel with the modem. If the telephone is off-hook because the user is listening to it, the modem cannot go on-hook while pulse dialing

because of the parallel circuit, and the pulses will not be dialed. Consequently, a phone cannot be used as a line monitor while pulse dialing.

The best solution is to identify when it is safe to pick up the phone and listen to it. This point in time can be predicted by calculating how long it will take to pulse dial the number. From the worst-case pulse dialing times just listed, it is easy to develop an algorithm to compute the time T to pulse-dial the phone number. Once the modem begins to dial and echoes the dialed digits of the phone number, T seconds must elapse before the phone can be picked up by the user.

The program below illustrates this timing technique. D$ represents the dial string for the phone number itself—no access codes. The second echoing of the phone number is used as the starting point for the timing because the first echoing is spurious—it occurs as the modem receives the D (DIAL) command, not when dialing actually begins. The first echoing of the dial string is read at line 100. The second echoing occurs as the modem actually dials the number, and is detected at line 160.

The computation appears on lines 120-150, and does not take into account W, P, *, or #, none of which would normally appear in a pulse-dialed phone number with no access codes. As the program converts each character of the dial string into a number (line 140), illegal characters are trapped out (lines 120 and 150). Then the program accumulates the time to pulse each digit plus the interdigit delay (line 140). The computation provides an extra margin of safety because of the time needed to compute the total dialing time.

```
100 F$="%D "&D$&CHR$(13) @ MODOUT F$ @ GOSUB 1000
110 DISP "Do not pick up phone!"
120 T=0 @ ON ERROR GOTO 150
130 FOR I=1 TO LEN(D$)
140 V=VAL(UPRC$(D$[I,I])) @ T=T+(V+10*(NOT V))/10+.8
150 NEXT I @ OFF ERROR
160 F$=D$ @ GOSUB 1000 @ T=T+TIME
170 IF TIME<T THEN 170 ELSE DISP "OK to pick up phone"
180 ! Program continues
1000 S$=""
1010 S$=S$&MODIN$ @ IF POS(S$,F$) THEN RETURN ELSE 1010
```

If access codes separated by waits appeared at the beginning of the dial string, the waits between the access codes will halt echoing for the duration of the waits (refer to the D command). In that situation, you should split the access codes off from the dial string, and use only the echoing of the phone number itself as the starting point for timing.

Manual Dialing

Your application may allow the user to dial a phone number manually. The general procedure is:

1) Ask if the user wants to dial the number manually.

2) If yes, the program must put the modem on-hook with the H (HANGUP) command in case the phone being used to dial generates pulses. Then the program should put up a message telling the user to dial the number, and to press a key on the HP-75 after dialing. Then the program must wait for the key to be pressed.

3) When finished dialing, the user must leave the phone off-hook or the call will be terminated. Then the user presses the key to signal the end of dialing. The program must then take the modem off-hook and monitor the line status using the O (ORIGINATE) command. The program would then proceed as if it had dialed the number itself. The user can put the phone back on-hook any time after the O command is executed.

This program gives an example of manual dialing. The Wait... message is needed because the H command takes 3 s to hang up the modem.

```
10 INPUT "Manual dial? y/n: ";A$
20 IF UPRC$(A$)="Y" THEN 30 ELSE BEEP @ DISP "Done" @ STOP
30 DISP "Wait..." @ MODOUT "%h"&CHR$(13) @ F$="%OK" @ GOSUB 1000
40 DISP "Press [RTN] when done dialing"
50 IF KEY$#CHR$(13) THEN 50
60 MODOUT "%o"&CHR$(13) @ F$="%CONNECT" @ GOSUB 1000
70 BEEP @ DISP "Connected!" @ STOP
1000 Z$=""
1010 Z$=Z$&MODIN$ @ IF POS(Z$,F$) THEN RETURN ELSE 1010
```

Connecting Two Pods Back-to-Back

The modem can be placed in either of two opposing modes of operation: ANSWER mode and ORIGINATE mode. When the modem enters ANSWER mode, it goes off-hook, transmits an answer tone, and waits 20 s for an originate tone. Conversely, when the modem enters ORIGINATE mode, it goes off-hook and monitors the phone line for an answer tone for about 50 s.

If one HP 82718A is in ANSWER mode and another is in ORIGINATE mode, the two modems can communicate with each other directly. The only hardware necessary for this modem-to-modem connection is a piece of telephone cable with a male modular connector on each end. By connecting the two modems with the cable, placing one modem in ANSWER mode and the other in ORIGINATE mode, and synchronizing them so that the answer tone and the originate tone occur concurrently, the communication link is established.

Below are two programs, one for each of two HP 82718As; the first one will put its modem into ANSWER mode, and the second one will put its modem into ORIGINATE mode. The programs then monitor the status of the modems to see when they are communicating with each other. The two programs need to be run at approximately the same time, since the originating modem only has 20 s to detect the answer tone issued by the answering modem.

The program for the answering modem is as follows:

```
10 ! This modem is set to ANSWER mode
20 MODEM ON ! Turn on modem
30 MODOUT "%a"&CHR$(13) ! Enter ANSWER mode
40 IF NOT HOOK? THEN 40 ! Wait until modem is off-hook
50 ON 1+HOOK?+CARRIER? GOTO 60,50,70
60 DISP "No connection" @ BEEP 220,1 @ STOP
70 DISP "Connection established"
80 MODOUT KEY$ @ DISP MODIN$; @ GOTO 80
```

The program for the originating modem is as follows:

```
10 ! This modem is set to ORIGINATE mode
20 MODEM ON ! Turn on modem
30 MODOUT "%o"&CHR$(13) ! Enter ORIGINATE mode
40 IF NOT HOOK? THEN 40 ! Wait until modem is off-hook
50 ON 1+HOOK?+CARRIER? GOTO 60,50,70
60 DISP "No connection" @ BEEP 220,1 @ STOP
70 DISP "Connection established"
80 MODOUT KEY$ @ DISP MODIN$; @ GOTO 80
```

Line 30 is the only difference between the two programs—the first program enters ANSWER mode by executing the A (ANSWER) command, and the second program enters ORIGINATE mode by executing the O (ORIGINATE) command. Line 50 controls the program flow while the two modems are establishing the communication link. It is the same self-timing technique described in the section "Detecting Connection With a Remote Computer". Line 40 is necessary so that the program will not halt immediately before the A and O commands have taken the modem off-hook. Line 80 allows one modem to transmit data typed in from the keyboard and transmit it to the other modem, and to display data received from the other modem.

Notice that lines 20 and 80 together constitute the simple terminal program described in "Terminal Program" earlier in this section. You can use just this terminal program on both HP 82718As to establish the communication link between the them. Enter the terminal program on both HP 82718As. Run both programs, and type %a [RTN] on one and %o [RTN] on the other. The connection will be established when you see %CONNECT followed by %OK (or %1 followed by %4) in both displays. If the connection is not established, you will see %NO ANS followed by %OK (or %2 followed by %4) in both displays. If you connect a telephone to either HP 82718A with another piece of telephone cable, you will be able to listen to the originate and answer tones and to the sound of data being transmitted from one modem to the other.

When the %CONNECT message appears, the two HP 82718As will be on-line to each other, and one can serve as the host for the other. By configuring the host HP 82718A to act like the host computer you expect to use, you can quickly begin to see what will happen when the other HP 82718A is used as a terminal. And you have complete control of the configuration of the "host", so you can experiment to see how the modem behaves in different circumstances.

Auto-Answer

The modem can auto-answer—that is, automatically answer incoming calls and respond to a received originate tone with an answer tone, thus establishing a communication link with the calling computer. The ring count distinguishes whether the modem behaves as AUTO-ANSWER or ORIGINATE-ONLY. If the ring count is zero, the modem will report incoming rings, but not answer them (ORIGINATE-ONLY). If the ring count is non-zero, the modem will answer incoming calls after *ring count* rings (AUTO-ANSWER).

AUTO-ANSWER and ORIGINATE-ONLY apply only when the modem is in MODEM mode. In VOICE mode, the modem only reports incoming rings, regardless of the ring count. For further details, see the description of the C (COUNT) command in the "Low-Level Modem Command Dictionary".

To specify AUTO-ANSWER, you must put the modem in MODEM mode and set the ring count to a value greater than zero. Here is a sample program:

```
10 INPUT "Number of rings (0-15): ";N$
20 N=VAL(N$) @ IF N>15 THEN BEEP @ GOTO 10 ! N>15 not allowed
30 IF N>9 THEN N$=CHR$(65+N-10) ! Convert to hex digit
40 MODOUT "%h,m,c "&N$&CHR$(13) ! Set MODEM mode and ring count
50 IF N THEN DISP "Will answer phone after";N;"rings" @ STOP
60 DISP "Will not answer phone"
```

The program hangs up the modem (line 40) since the modem will enter either ANSWER or ORIGINATE mode if it is off-hook when the M command is executed. The program will not accept a number of rings greater than 15 (line 20), and converts numbers from 10 to 15 into single hex digits for the C command (line 30).

Monitoring a Dialing Operation

The following example illustrates an approach to monitoring the progress of a dialing operation programmatically. The program dials a number, then redials if the number was busy. The program halts if 10 redials occur without connection, if a bad answer tone is received, or if there is no answer after the maximum number of rings (ring count + 4, up to a limit of nine) or after 50 s have elapsed since dialing ended. Modem responses are used to identify what conditions have occurred.

```
10 DIM M$[100]
20 MODOUT "%L 0,d i9 pw1(800)555-1212"&CHR$(13)
30 M$=""
40 GOSUB 1000 @ IF NOT POS(M$,"%BUSY") THEN 40
50 MODOUT "%r"&CHR$(13)
60 FOR B=1 TO 10 @ M$=""
70 GOSUB 1000 @ IF NOT POS(M$,"%BUSY") THEN 70
80 NEXT B
90 DISP "Phone is busy" @ STOP
100 DISP "Connection established"
110 ! Program continues here when connection made
990 !
1000 M$=M$&MODIN$ @ DISP M$;
1010 IF POS(M$,"%CONNECT") THEN POP @ GOTO 100
1020 IF POS(M$,"%BAD ANS") THEN 1040
1030 IF NOT POS(M$,"%NO ANS") THEN RETURN
1040 DISP "No connection made" @ STOP
```

Line 20 specifies long responses and dials 9-1(800)555-1212 (the 9 is an access code). To use short responses, substitute 1, B, 2, and 0 for the long responses CONNECT, BAD ANS, NO ANS, and BUSY respectively, and specify L 1 at line 20. The program could try to identify either the long or short response in each situation. For example, line 1020 could be substituted with the following:

```
1020 IF POS(M$,"%BAD ANS")+POS(M$,"%B") THEN GOTO 1040
```

Each line that looks for a modem response would look for either the long or the short response, but there is a potential problem with this. Recall from "Reading Modem Responses" that a single MODIN$ may not read all the characters of a response at once—it may take more than one MODIN$ to collect the entire

response. If the BUSY response is not all read at once by MODIN#, the response will be identified wrong. If only %BU is read, for example, the check for the response would be identified as the short form of %BAD ANS—%B. To avoid this conflict, the program should look for either long or short responses, but not both.

If the result of dialing is the BUSY response, detected on line 40, the program redials with the R command on line 50. Since the D command on line 20 stored the dial string, the dial string is optional for the R command. The R command will dial the string a maximum of 10 times. Variable E keeps track of the number of BUSY responses reported while redialing; after 10, the program stops.

If the result of dialing is CONNECT, detected at line 1010, the program will inform the user and proceed starting at line 110. If the result of dialing is BAD ANS or NO ANS, detected at lines 1020-1030, the program halts.

Uploading a Text File to a Host Computer

One of the most common applications using the modem is to upload a file from the HP-75 to a host computer. The file could be bar code data collected during inventory, data typed in while running an application program, a memo, or whatever you happen to have. This example will outline a simple, straightforward procedure for uploading an HP-75 text file to an arbitrary host.

To upload a file, you will need two programs. The first program, which will reside on the HP-75, will transmit the file. You can write this program by following the sample which is shown in this section. The second program, which will reside on the host computer, must accept input from the HP-75 as a remote terminal, and place the input in a file. Fortunately, you will not have to write this program, because every host computer has a program already available to perform this function—an editor. The purpose of any editor is to accept input from a remote terminal and place the input in a file. For this file upload procedure, a simple line editor is all that is needed.

Below is the general procedure to upload a file using a line editor. The numbers in parentheses correspond to the line numbers in the sample program. This procedure will work with any host; you only need to change the commands in the sample program to accommodate the syntax of your system. (The sample program illustrates the syntax for the HP 3000.)

GENERAL FILE UPLOAD PROCEDURE

1) Find out what file is to be uploaded (lines 30-60).

2) Set up protocol to match your host, and establish connection (lines 70-80).

3) Log on (line 90).

4) Purge any file with the same name as the file to be uploaded off the host system (line 100).

5) Invoke the line editor (line 110).

6) Enter the mode for inserting data into the editor workspace (line 120).

7) Read lines from the HP-75 text file and transmit them to the host (lines 130-150). Each line should be followed by the end-of-line indicator (line 20) for the host.

8) End the data insertion mode (line 160).

9) Save the file on the host system (line 170).

10) Exit the line editor (line 180).

11) Log off (line 190).

12) End the communication session (line 200).

```
10 DIM D$[255]
20 R$=CHR$(13) ! End-of-line sequence
30 INPUT "File to send: ";F$
40 ON ERROR BEEP @ DISP "Not a text file" @ WAIT 1 @ GOTO 30
50 ASSIGN # 1 TO F$,TEXT
60 OFF ERROR
70 MODOUT "%k e3aQSQSQ,f 4"&R$ @ GOSUB 2000 ! Protocol and parity
80 MODOUT "%d (xxx)yyy-zzzz"&R$ @ GOSUB 2000 ! Call the host
90 MODOUT "HELLO user.account"&R$ @ GOSUB 2000 ! Log on
100 MODOUT "PURGE "&UPRC$(F$)&R$ @ GOSUB 2000 ! Purge this file
110 MODOUT "EDITOR"&R$ @ GOSUB 2000 ! Invoke the editor
120 MODOUT "A"&R$ @ GOSUB 2000 ! Enter ADD mode
130 ON ERROR GOTO 160 ! Stop when EOF found
140 READ # 1 ; D$ @ MODOUT D$&R$ ! Send each line of the file
150 GOSUB 2000 @ GOTO 140 ! Read anything received and continue
160 MODOUT "//"&R$ @ GOSUB 2000 ! Exit ADD mode
170 MODOUT "KEEP "&UPRC$(F$)&R$ @ GOSUB 2000 ! Save the file
180 MODOUT "EXIT"&R$ @ GOSUB 2000 ! Exit the editor
190 MODOUT "BYE"&R$ @ GOSUB 2000 ! Log off
200 MODOUT "%h"&R$ @ GOSUB 2000 ! End communication session
210 STOP
2000 D$=MODIN$ @ IF LEN(D$) THEN 2000 ELSE RETURN
```

Line 10 allows text file lines of up to 255 characters. If your application actually creates lines that long, make sure your editor can accommodate them. Changing line 50 to `ASSIGN # 1 TO F$&":xmem",TEXT` will read lines out of text files on the electronic disc.

Line 70 sets the handshake protocol. The K (KONTROL) parameters, K1-K8 are set to E3AQSQSQ, where Q represents control-Q (DC1) and S represents control-S (DC3). Setting K1 to E enables the following protocols: Transmitter XON/XOFF (bit 3 set), Receiver XON/XOFF (bit 2 set), and Receiver ENQ/ACK (bit 1 set). Setting K2 to 3 causes DEL (bit 1 set) and NUL (bit 0 set) characters to be stripped from the incoming data stream. Setting K3 to A enables Prompt Mode (the HP 3000 editor sends a DC1 as the prompt character when it is ready to receive the next block of data). It also sets the receive buffer block size to 24 bytes. The modem will send an XOFF when the receive buffer has received 24 bytes (40 bytes empty).

The K4-K8 characters have their default values. K4 is the prompt character since Prompt Mode is enabled, and the default value for K4 (DC1) happens to match the prompt character that the HP 3000 editor issues.

Line 70 also sets parity and data length. The HP 3000 expects 8-bit data, which corresponds to F 4.

Line 110 invokes the editor. Your editor may require the file name as a parameter. If you include it here, you may not need it at line 170.

Line 120 enters ADD mode for the HP 3000 editor. Your editor may not have an explicit data insertion mode; instead, it may require a character preceding each line inserted. If the character was an I, then each line would be added by changing the MODOUT on line 140 to MODOUT "I"&D$&R$. Line 160, which ends ADD mode, would then probably not be needed.

Your editor may exit as it writes the data from its workspace into the file. If so, line 180 will not be needed.

Depending on your host system behavior, you may want to send a break (using the B (BREAK) command) before hanging up (line 200).

The subroutine on line 2000 simply reads the contents of the receive buffer and throws it away until the buffer is empty. This prevents echoed data (i.e., all the editor commands and all the data that is entered into the file) from filling up the receive buffer and halting transmission. If the modem has to send an XOFF because the buffer begins to fill up with echoed data, and never sends an XON because the buffer is not emptied, the transmission from the host may be halted. Your editor may have a mode which inhibits echoing the data. If so, you can invoke that mode when you enter the editor, or you could configure the port you dial up to not echo.

Command, Data, and Response Echoing

Three commands control command, data, and response echoing: E (ECHO), G (GIVEBACK), and L (LONG). (Remember that when the modem echoes anything (commands, data, or responses), it does so by placing the item being echoed in its receive buffer for MODIN$ to read.) Even though these commands are described in the "Low-Level Command Dictionary" and illustrated in different examples in Section 2, they are discussed again here to clarify their relationships.

The E (ECHO) command enables (E 0) and disables (E 1) echoing of low-level *commands*. The G (GIVEBACK) command enables (G 1) and disables (G 0) echoing of *data* sent from the HP-75 to the modem. The E and G commands are totally independent of each other—command echoing occurs independently of data echoing, and vice versa.

The L (LONG) command selects whether modem responses will be *long* (L 0) or *short* (L 1). Responses are unaffected by the E and G commands. Regardless of whether the modem is echoing commands or data, responses will be reported—the L command only controls their length.

Sending Command Characters as Data

There are three ways you can send the command character as data:

1) Use the N (NEW) command to change the command character to something else.

2) Use the U (UNLISTEN) command to instruct the modem not to monitor the transmit data stream for the command character.

3) Send the command character twice in a row.

An example of the use of the N (NEW) command is:

```
10 MODOUT "%N !"&CHR$(13)
20 MODOUT "%"
30 MODOUT "!N %"&CHR$(13)
```

Line 10 changes the command character to !. Line 20 sends the old command character. A carriage return is not needed because data only is being sent, not a command. Line 30 changes the command character back to %; if the ! character is subsequently sent to the modem, it will be seen as data. Notice that while the command character is !, it must be used in all commands, including !N for the NEW command on line 30. These three lines can be combined into one string for MODOUT as follows (note the placement of the carriage returns only after each N command):

```
10 MODOUT "%N !"&CHR$(13)&"%"&"!N %"&CHR$(13)
```

An example of the use of the U (UNLISTEN) command is:

```
10 MODOUT "%U 1"&CHR$(13)
20 MODOUT "%"
```

Line 10 tells the modem to treat the next 1 byte as data only—ignore any command characters in that data. The byte at line 20 is then sent on unprocessed. No carriage return is needed on line 20 because only data is being sent. These can be combined into one string for MODOUT as follows:

```
10 MODOUT "%U 1"&CHR$(13)&"%"
```

The use of two command characters is shown below:

```
10 MODOUT "%%"
```

To send a string containing one or more command characters (assuming there are no deliberately embedded commands), the three approaches are applied slightly differently. The N command is used as follows:

```
10 P=POS(S$,"%") @ IF P THEN MODOUT "%N !"&CHR$(13)
20 MODOUT S$
30 IF P THEN MODOUT "!N %"&CHR$(13)
```

Line 10 changes the command character to ! if there are any command characters in S$. Line 20 sends the data, and line 30 restores the old command character if it had been changed. This will not work, of course, if the string contains one or more ! characters.

The U command is used as follows:

```
10 IF POS(S$,"%") THEN MODOUT "%U "&STR$(LEN(S$))&CHR$(13)
20 MODOUT S$
```

If there are any command characters in S$, line 10 uses the length of S$ as the argument for U. However, the U command expects the number of bytes to be in *hex*, not decimal, so this approach only works for strings up to 9 characters long. To provide the U command a hex argument, you must either write a decimal-to-hex conversion routine or use the conversions in the HP-75 I/O ROM. Alternatively, you could send the string in blocks of fixed length, preceding each block with U *block length*. Or you could send the string in blocks of variable length, with each block consisting of all the data between command characters. Each command character would then be sent using U 1 or by sending it twice, as shown below:

```
10 P=POS(S$,"%") @ IF NOT P THEN 30
20 MODOUT S$[1,P]&"%" @ S$=S$[P+1] @ GOTO 10
30 MODOUT S$
```

Line 10 locates a command character. The string up to and including the command character is transmitted, followed by a second command character. Then the string is set to be the substring from past the location of the command character to the end of the string. This process is repeated for each command character in the string until there are no more, at which time the rest of the string is transmitted.

Note: because the previous examples do not read the modem receive buffer with MODIN$ after each MODOUT, any echoed data saved by MODOUT will be lost, and modem responses may fill up the receive buffer and halt data transmissions.

Decoding the QUERY Response

The Q (QUERY) command asks the modem for information on ring count, data communication status, and operating mode. This status information is placed in the modem receive buffer for MODIN$ to read in the form of a two or three byte string. The first byte is a hex digit (0-F) that represents the current ring count. The second byte identifies whether the modem is on-hook or off-hook, and in voice or data communication:

Character	Meaning
H	On-hook
C	Off-hook, carrier detected (data communication)
N	off-hook, VOICE mode (not data communication)

If the modem is in VOICE mode, then a third byte, a V, is returned after the first two bytes.

If command echoing is enabled (E 0) and the command character is %, the Q command is echoed into the receive buffer as %q CR LF. Then, the query response is preceded by a command character and followed by a CR LF. If responses are long (L 0), the query response will end with CR LF %OK CR LF (see appendix D, "HP 82718A Errors and Responses"). The entire response is as follows:

%q CR LF *%query response* CR LF CR LF %OK CR LF

Starting at the leftmost **%** in this string, the ring count will appear at character 6, the on-hook/off-hook status at character 7, and the V (if present) at character 8. To decode the query response, use a program like this:

```
100 R$=CHR$(13) @ MODOUT "%L 0,E 0"&R$ @ GOSUB 1000
110 MODOUT "%q"&R$ @ GOSUB 1000
120 C$=M$[6,6] @ IF C$<="9" THEN C=VAL(C$) ELSE C=NUM(C$)-65+10
130 DISP "Ring count =";C
140 ON POS("HCN",M$[7,7]) GOTO 150,160,170
150 DISP "On-hook"; @ GOTO 180
160 DISP "Off-hook with carrier"; @ GOTO 180
170 DISP "Off-hook"; @ GOTO 180
180 IF M$[8,8]="V" THEN DISP ";VOICE mode"
190 STOP
1000 M$=""
1010 M$=M$&MODIN$ @ IF NOT POS(M$,"%OK"&R$&CHR$(10)) THEN 1010
1020 RETURN
```

Line 100 specifies long responses and enables command echoing, then clears the response out of the receive buffer. Line 110 executes the Q command and reads its response from the buffer. Line 120 turns the hex digits A-F into a ring count of 10-15. Line 140 branches to statements where messages are displayed that identify the status: to line 150 if the next character of the response is an H, to line 160 if a C, and to 170 if an N.

At line 180, the next character is examined. If it is a V, it means VOICE mode, and this information is appended to the already displayed messages. This approach is used to accommodate all cases, including the unusual one (OCV) described in the caution under Q in the "Low-Level Modem Command Dictionary". The messages displayed by this program are:

Query Characters	Message Displayed
H	On-hook
HV	On-hook;VOICE mode
C	Off-hook with carrier
CV	Off-hook with carrier;VOICE mode
NV	Off-hook;VOICE mode

ELECTRONIC DISC SOFTWARE

The HP 82718A provides 32 or 64 kbytes of additional memory for your use. This memory does not expand the main memory of the HP-75; that is, you cannot create and run programs that are larger than the 14,000 bytes available in the HP-75 (22,000 bytes with the 82700A Memory Module). The extra memory behaves like a fast, reliable mass storage device—you can copy files to and from it, catalog it, and purge and rename files on it, just as you would on a cassette or disc drive. You can also create text and data files as large as the available memory in the HP 82718A. Operations are fast because memory can be accessed quicker than a mechanical device, and reliable because there are no moving parts. Because this RAM behaves like mass storage, the extra memory is called an *electronic disc.*

The electronic disc software lets you create, access, and modify text and data files on the electronic disc, establish a hierarchical file structure, and copy files into and out of the electronic disc. This software is designed to be a logical extension to the HP-75 operating system. Everything you understand about random and serial access of text and data files can be easily extended to include the electronic disc, and HP-75 commands that apply to mass storage devices will behave the same way when used with the electronic disc.

Adapting Existing Programs to Use the Electronic Disc

The electronic disc is identified by the device code XMEM (for *extended* or *external memory*). You can use this code in all the places where you would normally use an HP-IL device code for a mass storage device. For example, to copy file ABC from HP-75 main memory to a cassette drive named M1, you would type:

```
COPY "ABC" TO ":M1" [RTN]
```

To copy the same file to the electronic disc, type:

```
COPY "ABC" TO ":XMEM" [RTN]
```

Commands that are extended to use the XMEM device code for the electronic disc are ASSIGN #, CAT, COPY, PURGE, and RENAME.

Programs that interact with mass storage or with text and data files can usually be modified to use the electronic disc with only trivial changes. If a program is accessing files in main memory or on mass storage, all that you need to do is add or change the device code. One way to do that is to change all the file references so they append the device code as a string variable. Then set the string variable to reflect the use of main memory, mass storage, or the electronic disc:

```
100 COPY "ABC:M1" TO G$
200 PURGE "DEF:M1"
```

becomes

```
100 COPY "ABC"&C$ TO G$
200 PURGE "DEF"&C$
```

If C$=":M1", an HP-IL mass storage device named M1 will be used. If C$=":XMEM", the electronic disc will be used. If C$="" (null string), main memory will be used. Once this change has been made throughout the program, the file's location can be changed globally within the program by just changing the line where the string assignment occurs.

If a program is using text or data files, the only code that needs to change is that where the files are opened. For example, to open a file in main memory:

```
ASSIGN # 1 TO "XYZ",TEXT
```

becomes

```
ASSIGN # 1 TO "XYZ:XMEM",TEXT
```

Subsequent READ #, PRINT #, and RESTORE # statements will read from, write to, and move the data pointer within file XYZ on the electronic disc. Again, set a string variable to either the null string or :XMEM, and change file references to append the string variable (e.g., ASSIGN # 1 TO "XYZ"&C$).

NOTE

If you have existing programs that were written on a stand-alone HP-75 (i.e., not installed in an HP 82718A), you will need to reenter certain program lines for them to properly use the electronic disc. For details about this, see appendix C.

Initializing the Electronic Disc

The HP-75 and the electronic disc are initialized independently of each other, just as the HP-75 and an HP-IL mass storage device are initialized independently. If the HP-75 is reset ([SHIFT] [CTL] [CLR]), the electronic disc is not. Similarly, if the electronic disc is reset (INITIALIZE ":XMEM"), the HP-75 is not.

The electronic disc can be reset to its default state by executing INITIALIZE ":XMEM". This is the only way for the user to erase the contents of the electronic disc. The electronic disc software resets the disc automatically at three different times:

1) The HP-75 is turned on and the electronic disc has not previously been initialized.

2) The HP-75 is turned on and the electronic disc contents have been corrupted.

3) The electronic disc RAM is bad.

Since the reset occurred without your intervention, you are warned about the initialization with ERROR: XMEM INITIALIZEd. When the electronic disc is initialized, all files are erased, and the root directory ^ is created. XCAT$(0) returns the catalog for ^ and shows the time and date that the electronic disc was last initialized.

Interaction With the I/O ROM

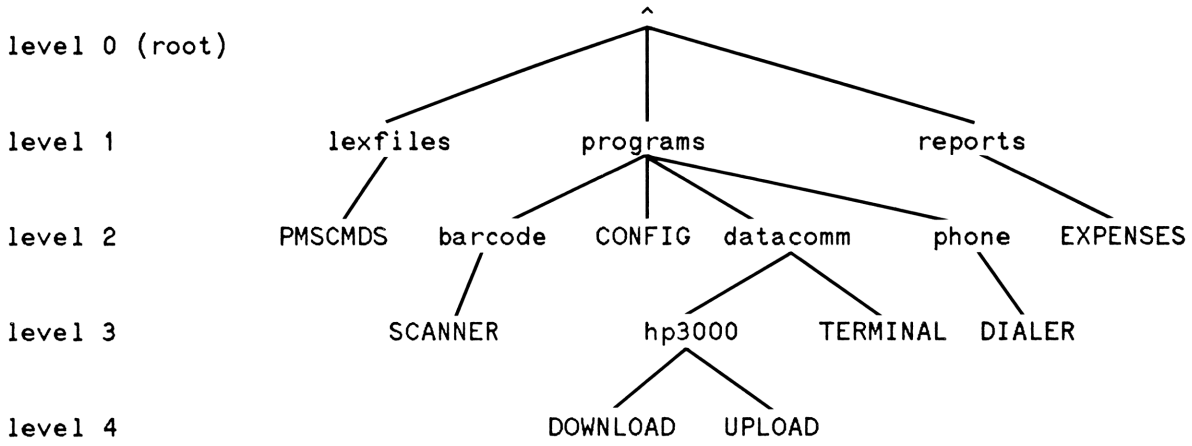
The HP 00075-15001 I/O ROM includes an extension of the HP-75 CALL statement that allows you to specify a list of parameters to be passed, and a device code specifying where the called file resides. For example, if you have an HP-IL mass storage device assigned the name M1, CALL "*filename*:M1" will copy that file into HP-75 main memory and call it. When the I/O ROM is plugged into an HP-75 installed in the HP 82718A, the electronic disc software extends the I/O ROM's CALL statement to include files on the electronic disc. If you use XMEM as the device code (i.e., CALL "*filename*:XMEM"), the program file on the electronic disc will be copied into main memory and called. CALL only works with filenames, not pathnames—the entire electronic disc will be searched to find the specified file.

The CAT# and CLEAR ASSIGN# commands provided by the I/O ROM will also work for electronic disc files, but the other 13 file manipulation commands will not. ADVANCE#, DELETE#, INDEX#, INSERT#, LASTLN?, LINELEN#, PRINT # USING, REPLACE#, SEARCH#, and SEEK# all report ERROR: invalid filespec for electronic disc files. FILE$ returns the name of an electronic disc file underlined, and LINE# returns a negative line number, unless a file of the same name exists in HP-75 main memory. ITEM# always returns 1 for an electronic disc file.

Hierarchical File Structure

The electronic disc software gives you the option of creating a hierarchical file structure on the disc. This allows you to organize your files into logical groups. For example, all your BASIC programs can be organized under a single directory. You could create a directory for all text files, or one directory for memos and another for files containing collected inventory data. Or you might want all programs and data relating to inventory under a single directory. Any file can be accessed by name only or by specifying the directory it is in.

Here is an example of how a series of files and directories might be organized on the electronic disc. For clarity, directories are distinguished from individual files in the diagram by lowercase letters. The level of each file or directory within the hierarchy is shown along the left edge of the diagram.



In this hierarchical file structure, files and directories are organized as branches in a tree, and each must have a unique name. Only directories can have other files or directories attached, called child files or directories—the files or directories one level down in the hierarchy. Individual files (BASIC, text, LEX, etc.) cannot have children in the hierarchy. Any file or directory will always be attached to a directory, called its parent directory—the directory one level up in the hierarchy. (Note that a directory is not a file—it is a file organizer.)

The only fixed directory is the root directory, called ^, at the beginning of the tree. Its level is zero, and its parent directory is also ^.

On any level within a directory, all files and directories are in alphabetical order. Examples are files on level 4 within the hp3000 directory, the directories on level 1 within the root directory, and the files and directories on level 2 within the programs directory. Consequently, on any level across the hierarchy, files and directories are not necessarily in alphabetical order (for example, the files and directories on levels 2 and 3).

File searches start at the location specified in the pathname, or at the root directory if no location is given. A search goes along the left-most branch (alphabetically) of the tree until the end of the branch is reached. Then the search proceeds back up the branch to a directory and from there to the next left-most branch until that branch has been searched. The search continues in this manner across the hierarchy until the file has been found or the end of the file system is reached.

For example, to find the TERMINAL file, the electronic disc software searches the lexfiles directory, then the programs directory. Within the programs directory, the barcode directory is searched, then the branch containing CONFIG, then the datacomm directory. Within the datacomm directory, the hp3000 directory is searched, then the branch containg TERMINAL. Because of this search order, TERMINAL will be found after UPLOAD, even though TERMINAL is lower alphabetically.

You can select the current directory (initially ^) with DIR IS. Then you have direct access to files within the directory without including the directory name as part of the pathname. This is similar to having the files in that directory as the only files on the electronic disc. Wild cards can be used to ease file selection (discussed later).

Directory Levels

The maximum nesting of directory levels is 7 levels, including the root directory. ^ is level 0, so the directories you create will be levels 1-6. The child files of a level 6 directory will be level 7, but no directory can be on level 7. Directory levels are shown in the length field of the indented catalog listing produced by CAT ALL ":XMEM". Directory levels are also shown in character position 17 in the string returned by XDIR$. File and directory levels are shown in the tenth character position of the string returned by XCAT$ (for directories, the level is at both positions 10 and 17).

Pathnames

You specify a file on the electronic disc with "*pathname*:XMEM". A pathname is the path that must be followed through the hierarchy to reach the desired file. Syntactically, a pathname looks like this:

pathname = [*location*][*filename*]

The location is the sequence of directories leading to the one containing the desired file, as follows:

location = start[>*first directory*>*second directory*>...>*last directory*>]

Start indicates the location within the hierarchy at which the location sequence begins, and defaults to the current directory. *Start* can be a directory name, ^ (the root directory), > (the next name is the name of the child file or directory), < (parent directory—can be used more than once anywhere, except after >), or ? (discussed shortly).

The use of location is optional. If you do not include location in the pathname when searching for a file, (that is, the pathname is of the form *filename*:XMEM), the entire electronic disc is searched, just like HP-75 main memory or a mass storage device. If a file is copied to (if it doesn't already exist) or created on the electronic disc and has no location, it is put in the current directory. The use of location implies a knowledge of the file structure, and file searches and creations are performed as indicated by the pathname.

Each directory occupies 25 bytes of the electronic disc. If you do not want to take up that memory, you do not have to use the hierarchical file structure. In that case, all files will be organized alphabetically under the root directory.

Unlike mass storage file specifiers, pathnames do not allow passwords. There is no file protection mechanism provided for files on the electronic disc.

WILD CARDS IN PATHNAMES. Two wild card characters can be used in pathnames on the electronic disc. The dash (−), used in either filenames or directory names, replaces one character, and will match any character. The question mark (?), used for location, indicates zero or more directories, and causes file searches to begin at the location up to the ? (default is the current directory). The question mark must be used in the form " [*location*]?>*filename*:xmem".

All pathnames used to reference electronic disc files or directories can use wild cards except PURGE DIR. Wild cards cannot be used to reference files in HP-75 main memory except when using the REMOVE statement.

EXAMPLES OF PATHNAMES. Here are some examples of ways to specify the file DOWNLOAD in the example hierarchy.

download:xmem	Search entire electronic disc
^>programs>datacomm>hp3000>download:xmem	Follow this path to DOWNLOAD
hp3000>download:xmem	Search for DOWNLOAD starting at the hp3000 directory
^?>do------:xmem	Find the first occurrence of a file with the first 2 letters do
^?>datacomm>?>download:xmem	Search only the datacomm directory

```
^programs>datacomm>?>download:xmem
```

Search only the datacomm directory

Comprehensive Example

This example will create the file system illustrated earlier.

```
initialize ":xmem"
```

Reset electronic disc

```
cat all ":xmem"
  Name    Type Len  Time    Date
^           D    0
```

Show everything on electronic disc

```
create dir "lexfiles:xmem"
```

Create lexfiles directory as child of ^

```
create dir "^>programs:xmem"
```

Create programs directory as child of ^

```
copy "config" to "programs>:xmem"
```

Copy CONFIG into programs directory

```
create dir "programs>datacomm:xmem"
```

Create datacomm directory as child of programs

```
create dir "datacomm>hp3000:xmem"
```

Create hp3000 directory as child of datacomm

```
create dir "programs>phone:xmem"
```

Create phone directory as child of programs

```
copy "terminal" to "datacomm>:xmem"
```

Copy TERMINAL to datacomm directory

```
dir is "hp3000:xmem"
```

Set current directory to hp3000

```
copy "download" to ":xmem"
```

Copy DOWNLOAD to hp3000 directory

```
copy "upload" to ":xmem"
```

Copy UPLOAD to hp3000 directory

```
copy "dialer" to "ph------>:xmem"
```

Copy DIALER to first directory starting with ph

```
copy "pmscmds" to "lex----->:xmem"
```

Copy PMSCMDS to first directory starting with lex

```
xdir$
HP3000      D     3 20:19 06/05/84
```

Identify current directory

```
create dir "<<>barcode:xmem"
```

Create barcode directory on same level as datacomm and phone

```
copy "scanner" to "^>programs>barcode>:xmem"
```

Copy SCANNER to barcode directory

```
create dir "^>reports:xmem"
```

Create reports directory on same level as lexfiles and programs

```
dir is "reports:xmem"
```

Set current directory to reports

```
assign # 1 to "expenses:xmem",text
```

Open EXPENSES text file in reports directory

```
print # 1; "Expenses for vendor trip"
```

Write a string to the file

```
cat ">:xmem"
  Name    Type Len  Time   Date
REPORTS     D       20:20 06/05/84
EXPENSES    T    42 20:20 06/05/84
```

Show files within this directory

```
cat ":xmem"
  Name    Type Len  Time   Date
PMSCMDS     L 2512 21:17 10/25/82
SCANNER     B 8011 20:27 04/10/84
CONFIG      B  864 14:02 03/21/84
DOWNLOAD    B  987 13:02 01/11/84
UPLOAD      B  580 13:47 04/17/84
TERMINAL    B 2626 22:18 12/20/83
DIALER      B 6595 23:36 12/23/83
EXPENSES    T    42 20:20 06/05/84
```

Show all files on electronic disc

```
cat all ":xmem"
  Name    Type Len  Time   Date
^           D    0
  LEXFILES    D    1
    PMSCMDS     L 2512
  PROGRAMS      D    1
    BARCODE     D    2
      SCANNER     B 8011
    CONFIG      B  864
    DATACOMM    D    2
      HP3000      D    3
        DOWNLOAD    B  987
        UPLOAD      B  580
      TERMINAL    B 2626
    PHONE       D    2
      DIALER      B 6595
  REPORTS     D    1
    EXPENSES    T    42
```

BAR CODE SOFTWARE

The HP 82718A includes software that decodes two types of bar code labels, Code 39 and Code 11, when scanned by a digital bar code wand and the HP-75D. The decoders are compatible with switched and unswitched wands, high resolution infrared wands (wavelength 830-920 nm), and medium resolution visible light wands (wavelength 700 nm).

The CODE39$ and CODE11$ keywords scan Code 39 and Code 11 bar code labels up to 42 characters long (including the check digit) and decode them into strings of ASCII characters. Audible feedback indicates good and bad scans. The keywords CDIGIT ON and CDIGIT OFF (default behavior) allow you to specify whether the decoders should automatically verify the check digit while scanning. For unreadable labels, CDV39 and CDV11 verify the check digits of labels entered from the keyboard.

The bar code commands duplicate commands that exist in the HP 82725A Bar Code Reader Module (see appendix C for details).

Scanning a Bar Code Label

To scan a label, enter the keyword for the type of label to be decoded (either CODE39$ or CODE11$), then scan the label with the wand. If the HP-75D can decode the label, it will return the scanned data as a string, and issue a high tone. If the label did not decode properly, or if 2-1/2 minutes have gone by with no scanning, a null string will be returned, and a low tone will be issued. If CDIGIT ON has been executed, labels without check digits will not decode properly.

Here is the recommended procedure for scanning a bar code label with a wand:

1) Connect the wand to the HP-75D, and turn the computer on.

2) Execute the keyword for the type of label you are scanning (CODE39$ for Code 39 labels or CODE11$ for Code 11 labels).

3) If you are using a switched wand, depress and hold the wand switch.

4) Place the wand tip into the quiet zone (the white area preceding the bar code label), with a tilt angle between 0 and 30 degrees normal to the surface.

5) Scan the entire label smoothly, making sure the wand tip does not go outside the rectangular boundary of the bar code label.

6) If you are using a switched wand, release the switch after scanning the label to conserve battery power.

Using Audible and Visual Prompts

Most applications will use some type of programmed prompting to signal the user when to scan the label. The BEEP command gives audible prompts, and the DISP command gives visual prompts.

The prompting done by the program and the scanning done by the user must be synchronized. The program should not wait so long after a signal that the user scans before the decoder is ready. Here is a program guaranteed to frustrate anyone trying to scan labels:

```
5 DELAY 3
10 DISP "Scan label"
20 A$=CODE39$
30 IF A$="" THEN 10
```

A subtlety of the DISP command is its behavior with the current DELAY rate. When a DISP statement is executed, the program waits for the DELAY time before continuing execution. This wait causes difficulty in scanning labels. The user begins scanning when the Scan label prompt appears, scans two or three times during the 3 s wait, and gets no audible feedback. Perhaps in the middle of the fourth scan, CODE39$ begins to execute, sees only the last half of the label, and issues a low tone to indicate a bad scan. Then the cycle repeats until the user hits it just right.

The user has experienced a tremendous frustration at the perceived poor performance of the software, when in fact poor program design has caused all the problems. The time from when the prompt appears to when it is valid to scan should be as short as possible to prevent the program from being out of sync with the user. Setting DELAY 0 ensures this:

```
5 DELAY 0
10 DISP "Scan label"
20 A$=CODE39$
30 IF A$="" THEN 10
```

Checking Field Length

You can check the field length of decoded bar code labels by using the BASIC function LEN. This provides an additional verification of the scan; you only need to know the actual length of the label. Here is an example of using LEN for field length checking.

```
100 DELAY 0 @ L=9 ! Define desired length
110 DISP "Scan label" ! Prompt
120 A$=CODE39$ ! Execute decoder
130 IF LEN(A$)=L THEN 180 ! Field length = L
140 DISP "Length error" @ WAIT .5 ! Bad length
150 DISP "Scan again" ! Rescan prompt
160 BEEP 1000,.1 ! Audible prompt
170 GOTO 120
180 DISP "Good read" ! Length verified
```

Verifying Check Digits

Use CDIGIT ON when you want the decoders to automatically verify the check digit while scanning. However, not all labels can be scanned successfully. When an unreadable bar code label is encountered, it may be necessary for the user to enter the label from the keyboard. In that case, the functions CDV39 and CDV11 allow the program to verify that the label was entered correctly from the keyboard. These functions perform the check digit verification that is normally done by CODE39$ and CODE11$ with CDIGIT ON.

CDV39 and CDV11 accept a single string parameter, and return a 1 if the check digit verified or a 0 if the verification failed (Boolean functions). Here is an example of using CDV39 to verify Code 39 labels:

```
100 INPUT "Label: ";A$ ! Keyboard entry
110 IF CDV39(A$)=0 THEN 100 ! Compute and verify check digit
120 ! If CDV39=1 then label was good otherwise input again
```

Comprehensive Example

In this example, we will update inventory on 50 different products on a regular basis. Each item has a Code 39 label with a check digit character. Since all labels are the same length, the length can be used as an additional check on the validity of the input.

The example program collects decoded bar code labels and stores them in a text file in the HP-75D. The program does not show what happens to that file after all the data has been collected—it could be transmitted to a host computer via the built-in modem, saved on a mass storage device, used to determine what orders to place, etc.

The program saves the collected data in a text file, rather than a BASIC data file. Data files contain overhead that text files do not, so text files conserve memory. The main loop (lines 50-160) collects the label and quantity for each of the 50 items. The program allows up to three bad scans before asking the user to type in the label. Every time a new item's label is scanned, the bad read count C is initialized (line 70). The count is incremented for each bad scan until three consecutive bad reads have occurred. Then the program asks the user to enter the label from the keyboard (line 260), and verifies its check digit (line 280).

Once a valid label has been input, its length is checked using LEN (line 120 for scanned input or 290 for keyboard input). Finally, the label, quantity, and current date are written into the text file. A space is placed in each record between the label, quantity, and date so that those items can be individually extracted at a later time.

It is important that the program establish the operating state of the decoder, so automatic check digit verification is set on line 40. Line 170 turns off automatic verification. If you then use either decoder for labels without check digits, they will still scan properly.

```
10 DIM A$[42] ! Allow for maximum length label
20 ASSIGN # 1 TO "INVEN:XMEM",TEXT ! Open electronic disc file
30 DELAY 0 ! Prevent DISP from waiting
40 CDIGIT ON ! Automatic check digit verification
50 INPUT "Length of label: ";L ! Get field length
60 FOR I=1 TO 50 ! Start main loop for 50 readings
70 C=0 ! Initialize bad scan count
80 DISP "Scan label" ! Visual prompt to start scan
90 BEEP 1000,.1 ! Audible prompt to start scan
100 A$=CODE39$ ! Decode label
110 IF A$="" THEN 230 ! Bad read
120 IF LEN(A$)#L THEN 230 ! Bad length
130 INPUT "Quantity: ";Q$ ! Number of items with this label
140 ! Write the label, quantity, and date
150 PRINT # 1 ; " ",A$," ",Q$," ",DATE$
160 NEXT I ! Continue loop
170 CDIGIT OFF ! Turn off check
180 ASSIGN # 1 TO * ! Close INVEN file
190 END
200 !
210 ! Bad scan or bad length
220 !
230 C=C+1 ! Increment bad scan count
240 IF C<3 THEN DISP "Rescan label" @ GOTO 90 ! Try again
250 A$="" ! Initialize for keyboard input
260 INPUT "Label: ",A$;A$ ! Three bad scans, use keyboard
270 ! If good keyboard input, go check length else try again
280 IF CDV39(A$)=0 THEN 300
290 IF LEN(A$)=L THEN 130 ! If length OK, get quantity
300 DISP "Invalid label" @ WAIT .5 @ GOTO 260
```

KEYWORD DICTIONARY

SECTION 3

This section describes all the BASIC keywords provided by the HP 82718A Expansion Pod. First, the name of the keyword is given, followed by the category of keyword (high-level modem, electronic disc, or bar code) and whether the keyword is a function (returns a number or string) or a statement (performs an operation but returns nothing). The next line lists the syntax of the keyword and its expected parameters. One or more examples of the use of the keyword follows. The description of the keyword appears next, and includes any cautions, set off by bullets, that should be observed when using the keyword. Finally, a list of possible errors that may be reported by the keyword is given.

Notes on syntax: Optional parameters are surrounded with brackets ([ ]). XMEM is the device code for the electronic disc. The syntax of pathnames, locations, and directories is described in "Electronic Disc Software" in Section 2.

Notes on error numbers: Error numbers of 150 or greater are those reported by the HP 82718A, and are described in appendix D, "HP 82718A Errors and Responses." The other error numbers are included for completeness, but are described in the *HP-75 Owner's Manual.*

ASSIGN

electronic disc **statement**

```
ASSIGN # file number TO "filename:XMEM" [,BASIC]
                                        [,TEXT ]
ASSIGN # file number TO *
```

```
ASSIGN # 1 TO "EXPENSES:XMEM",TEXT
ASSIGN # 1 TO *
```

Associates the specified *file number* with the specific file on the electronic disc and sets the corresponding data pointer to the first line of the file. A new file is created on the electronic disc if the named file doesn't exist. If the file type is not specified, the default is BASIC.

- ASSIGN # must be used with a filename, not a pathname. To open a file that already exists within the file hierarchy, execute ASSIGN # to the filename. To create and open a file that does not exist within the file hierarchy, first specify the directory with DIR IS, and then execute ASSIGN # to the filename.

- Refer to the *HP-75 Owner's Manual* and the *HP-75 Reference Manual* for further information. Everything that applies to the use and behavior of ASSIGN # in main memory applies to the electronic disc as well.

Errors: 16, 62, 63, 64, 68, 212

AVAIL?

high-level modem **function**

```
AVAIL?
```

```
100 IF NOT AVAIL? THEN 100
```

Checks to see if there is data available in the modem receive buffer. Returns a 0 if there is no data available, or a 1 if there is data available. Unlike MODIN$, which reads any data in the receive buffer, the data remains in the buffer after AVAIL? is executed. AVAIL? can therefore be used as an alternative to MODIN$ for detecting data in the buffer, but not reading it.

- AVAIL? only identifies the presence of data in the modem receive buffer, not the presence of data saved by MODOUT (see MODOUT).

Errors: 150, 151, 153

CARRIER?

high-level modem **function**

```
CARRIER?
```

```
100 IF NOT CARRIER? THEN 100 ELSE DISP "Now on-line"
```

Checks if there is a carrier. Returns a 0 if there is no carrier present, or a 1 if there is a carrier present.

Errors: 150, 151, 153

CAT

electronic disc **statement**

```
CAT "pathname:XMEM"
CAT ALL ":XMEM"
```

See examples below.

Catalogs files and directories on the electronic disc. If you use directory-related characters (^, <, >) or wild cards (−, ?), they imply knowledge of the hierarchical file structure. Directories are listed only when this knowledge is detected. Below are examples of typical use of CAT. The examples are based on the hierarchical file system illustrated in "Electronic Disc Software" in Section 2. Note that the [↑], [↓], [SHIFT] [↑], and [SHIFT] [↓] keys work the same as for the HP-75 CAT commands.

```
CAT ":XMEM"
```

Catalog all files on the electronic disc (no directories).

```
CAT ":XMEM" [RTN]
  Name    Type Len  Time   Date
PMSCMDS     L 2512 21:17 10/25/82
SCANNER     B 8011 20:27 04/10/84
CONFIG      B  864 14:02 03/21/84
DOWNLOAD    B  987 13:02 01/11/84
UPLOAD      B  580 13:47 04/17/84
TERMINAL    B 2626 22:18 12/20/83
DIALER      B 6595 23:36 12/23/83
EXPENSES    T   42 20:20 06/05/84
```

CAT "*filename*:XMEM"

Catalog first file that fits filename.

```
CAT "UPLOAD:XMEM" [RTN]
  Name    Type Len  Time   Date
UPLOAD      B  580 13:47 04/17/84

CAT "D-------:XMEM" [RTN]
  Name    Type Len  Time   Date
DOWNLOAD    B  987 13:02 01/11/84
```

CAT ">:XMEM"

Catalog all the files and directories in the current directory.

```
CAT ">:XMEM" [RTN]  (assume current directory is datacomm)
  Name    Type Len  Time   Date
DATACOMM    D       20:16 06/05/84
HP3000      D       20:16 06/05/84
TERMINAL    B 2626 22:18 12/20/83
```

CAT "*location*:XMEM"

Catalog all the files in specified directory.

```
CAT "REPORTS:XMEM" [RTN]
  Name    Type Len  Time   Date
REPORTS     D       20:20 06/05/84
EXPENSES    T   42 20:20 06/05/84
```

CAT ALL ":XMEM"

Catalog all the files on the electronic disc, indented and sorted by level. The Len field in the catalog line for directories is the level of the directory. The catalog produced by CAT ALL ":XMEM" does not show the time and date.

```
CAT ALL ":XMEM" [RTN]
  Name    Type Len  Time    Date
^           D    0
  LEXFILES   D    1
    PMSCMDS    L 2512
  PROGRAMS    D    1
    BARCODE    D    2
      SCANNER    B 8011
    CONFIG     B  864
    DATACOMM   D    2
      HP3000     D    3
        DOWNLOAD   B  987
        UPLOAD     B  580
      TERMINAL   B 2626
    PHONE      D    2
      DIALER     B 6595
  REPORTS    D    1
    EXPENSES   T   42
```

- Refer to the *HP-75 Owner's Manual* and the *HP-75 Reference Manual* for further information. Everything that applies to the use and behavior of CAT in main memory and on a mass storage device applies to the electronic disc as well.

Errors: 62, 63, 212

CDIGIT OFF

bar code **statement**

```
CDIGIT OFF
```

CDIGIT OFF

Turns off automatic check digit verification while scanning with the Code 39 and Code 11 decoders. After CDIGIT OFF, check digit verification will not be performed by CODE39$ or CODE11$ until CDIGIT ON is executed. The default check digit verification behavior after the HP-75 has been reset ([SHIFT] [CTL] [CLR]) is CDIGIT OFF.

Errors: none

CDIGIT ON

bar code **statement**

```
CDIGIT ON
```

CDIGIT ON

Turns on automatic check digit verification while scanning with the Code 39 and Code 11 decoders. After CDIGIT ON, check digit verification will be performed by CODE39$ or CODE11$ until

CDIGIT OFF is executed. The default check digit verification behavior after the HP-75 has been reset ([SHIFT] [CTL] [CLR]) is CDIGIT OFF.

Errors: none

CDV11 bar code function

```
CDV11("Code 11 label")
```

```
IF NOT CDV11(L$) THEN DISP "Bad label"
```

Verifies the check digit for a *Code 11 label* entered from the keyboard because the label cannot be scanned. Returns a 0 if the check digit of the label is not valid, or a 1 if the check digit is valid.

Errors: none

CDV39 bar code function

```
CDV39("Code 39 label")
```

```
IF NOT CDV39(L$) THEN DISP "Bad label"
```

Verifies the check digit for a *Code 39 label* entered from the keyboard because the label cannot be scanned. Returns a 0 if the check digit of the label is not valid, or a 1 if it is valid.

Errors: none

CODE11$ bar code function

```
CODE11$
```

```
100 L$=CODE11$ @ IF NOT LEN(L$) THEN 100
```

Scans a single Code 11 label. Returns the decoded label as a string of ASCII characters and issues a high tone if the label decoded successfully. Returns a null string and issues a low tone if the label did not decode successfully or if 2-1/2 minutes go by with no scanning.

If CDIGIT ON has been specified, the decoder will calculate the check digit and compare it to the check digit scanned as part of the label (usually the last character of the label). The label will not decode successfully unless these two check digits match. If CDIGIT OFF has been specified, the check digit will not be computed or compared with the check digit in the label.

The longest label that can be decoded is 42 characters long, including the check digit. Labels longer than 42 characters will not decode successfully.

Because CODE11$ requires a large amount of scratch space while it is executing (1400 bytes), there might not be enough memory available to execute CODE11$.

- During the 2-1/2 minutes while CODE11$ is waiting for a label to be scanned, the keyboard is disabled. Only the expiration of the time or the scanning of a label will stop the decoder—pressing the [ATTN] key will not. If CODE11$ is executed with an HP-75C installed in your HP 82718A rather than an HP-75D, you will have to wait the 2-1/2 minutes for the decoder to halt, since the HP-75C has no wand connector and thus cannot scan labels. If you are waiting for CODE11$ to halt while running a program, press the [ATTN] key, and the program will halt as soon as the 2-1/2 minutes have passed.

Errors: 16

CODE39$ bar code function

```
CODE39$
```

```
100 L$=CODE39$ @ IF NOT LEN(L$) THEN 100
```

Scans a single Code 39 label. Returns the decoded label as a string of ASCII characters and issues a high tone if the label decoded successfully. Returns a null string and issues a low tone if the label did not decode successfully or if 2-1/2 minutes go by with no scanning.

If CDIGIT ON has been specified, the decoder will calculate the check digit and compare it to the check digit scanned as part of the label (usually the last character of the label). The label will not decode successfully unless these two check digits match. If CDIGIT OFF has been specified, the check digit will not be computed or compared with the check digit in the label.

The longest label that can be decoded is 42 characters long, including the check digit. Labels longer than 42 characters will not decode successfully.

Because CODE39$ requires a large amount of scratch space while it is executing (1400 bytes), there might not be enough memory available to execute CODE39$.

- During the 2-1/2 minutes while CODE39$ is waiting for a label to be scanned, the keyboard is disabled. Only the expiration of the time or the scanning of a label will stop the decoder—pressing the [ATTN] key will not. If CODE39$ is executed with an HP-75C installed in your HP 82718A rather than an HP-75D, you will have to wait the 2-1/2 minutes for the decoder to halt, since the HP-75C has no wand connector and thus cannot scan labels. If you are waiting for CODE39$ to halt while running a program, press the [ATTN] key, and the program will halt as soon as the 2-1/2 minutes have passed.

Errors: 16

COPY

electronic disc **statement**

```
COPY [ "filename"                          ] TO "pathname:XMEM"
     [ APPT                                ]
     [ KEYS                                ]
     [ "mass storage file specifier"       ]
     [ "pathname:XMEM"                     ]

COPY "pathname:XMEM" TO "filename"
                        APPT
                        KEYS
                        "mass storage file specifier"
                        "pathname:XMEM"
```

```
COPY "DOWNLOAD" TO "^>PROGRAMS>DATACOMM>HP3000>:XMEM"
COPY "DOWNLOAD:XMEM" TO ":M1"
```

Copies files within the electronic disc and between the electronic disc, the HP-75, and HP-IL mass storage devices.

- Files cannot be copied directly between the electronic disc and magnetic cards. To copy electronic disc files to (or from) cards, copy the files to main memory first, then from main memory to cards (or to the electronic disc).
- When a file is copied to a new name, the file does not receive a new time and date in its catalog except when copying files from the electronic disc to a mass storage device.
- Refer to the *HP-75 Owner's Manual* and the *HP-75 Reference Manual* for further information. With the above exceptions, everything that applies to the use and behavior of COPY in main memory and on a mass storage device applies to the electronic disc as well.

Errors: 16, 57, 58, 60, 62, 63, 64, 65, 66, 68, 92, 93, 94, 95, 96, 211, 212, 213

CREATE DIR

electronic disc **statement**

```
CREATE DIR "location:XMEM"
```

```
CREATE DIR "^>PROGRAMS>DATACOMM>HP3000:XMEM"
CREATE DIR "HP3000:XMEM"
```

Creates a new directory on the electronic disc. The new directory will be the child of the directory specified by the *location*, or the child of the current directory. This is the only way to create a new directory. No more than six levels of directories can be created.

Errors: 16, 62, 63, 212, 213

DIR IS

electronic disc **statement**

```
DIR IS "location:XMEM"
```

```
DIR IS "^>PROGRAMS>DATACOMM>HP3000:XMEM"
DIR IS "HP3000:XMEM"
```

Specifies the current directory. This is the only command that can change the current directory. If wild cards are used, DIR IS will only match directories; it will skip over files when trying to find a match. Once a directory is specified, subsequent use of filenames only, rather than pathnames, will cause files to be created and opened (assigned to) as children of the specified directory.

Errors: 63, 212

FRAME?

high-level modem **function**

```
FRAME?
```

```
IF FRAME? THEN DISP "Framing error"
```

Checks for a framing error. Returns a 0 if the last MODIN$ had no framing error, or a 1 if there was a framing error. Remember that MODIN$ returns not only the contents of the modem receive buffer, but also any data echoed by MODOUT. FRAME? reports a framing error for data returned by MODIN$ from either source.

Errors: 150, 151, 153

HOOK?

high-level modem **function**

```
HOOK?
```

```
IF NOT HOOK? THEN DISP "Modem on-hook"
```

Checks if modem is on-hook. Returns a 0 if the modem is on-hook (not on phone line), or a 1 if the modem is off-hook (using phone line).

If the modem is pulse dialing, HOOK? should not be used until after dialing has been completed. Pulse dialing works by actually putting the modem on-hook briefly during every pulse of the digit being dialed. During pulse dialing, either adaptively or explicitly, HOOK? can return a 0 that does not reflect the true status of the modem.

Errors: 150, 151, 153

INITIALIZE ":MODM" high-level modem statement

```
INITIALIZE ":MODM"
```

INITIALIZE ":MODM"

Resets the modem to its default state (shown below) and turns it off. INITIALIZE ":MODM", which takes about 18 s to execute, can be used even if the modem is off, and will always reset the modem to its default state, even if the modem is deadlocked (see "Modem Deadlock" in Section 2).

Parameter or Mode	Default Condition	Equivalent Low-Level Command
Ring count	0	C 0
Command echoing	yes	E 0
Parity	none	F 4
Data echoing	no	G 0
Modem hung up	yes	H
Handshake protocol	Transmitter XON/XOFF Receiver XON/XOFF Receiver ENQ/ACK	
Parity stripped	no	K E03QSQSQ
Chars stripped	none	
Block size	32 bytes	
Handshake chars	QSQSQ	
Response length	long	L 0
MODEM or VOICE mode	MODEM	M
Command character	%	N %
Transmit buffer	cleared	none
Receive buffer	cleared	Z
Cmd char is data (UNLISTEN)	no	none

Note: Q represents control-Q (DC1) and S represents control-S (DC3)

- INITIALIZE ":MODM" must be used instead of the I (INITIALIZE) low-level modem command if the command character is not known, or if U 0 has been executed (see the U (UNLISTEN) command).
- If the modem is bad and reports the FAIL response when the modem is turned on, INITIALIZE ":MODM" will continue to execute until you press [ATTN].

Errors: 151

INITIALIZE ":XMEM" electronic disc statement

```
INITIALIZE ":XMEM"
```

```
INITIALIZE ":XMEM"
```

Resets the electronic disc to its default state. All files and directories will be erased. The root directory (^) will be created at the beginning of the file system at this time. If you need to know when the electronic disc was initialized, examine the time and date fields of the catalog returned by XCAT$(0). Refer to "Electronic Disc Initialization" in Section 2 for information about other times that the electronic disc is initialized.

Errors: 17, 63, 94, MEMIC didn't select

MAX XMEM electronic disc function

```
MAX XMEM
```

```
M=MAX XMEM
```

Returns total amount of memory on the electronic disc. This amount will be either 32768 (HP 82718A Option 032) or 65536 (HP 82718A Option 064).

Errors: 212

MODEM OFF high-level modem statement

```
MODEM OFF
```

```
MODEM OFF
```

Turns the modem off. The modem is also turned off by turning off the HP-75 and by INITIALIZE ":MODM". If a program halts prematurely before it has a chance to turn off the modem, the modem will continue to consume battery power until you turn the modem off or until the HP-75 is turned off (or turns itself off). MODEM OFF (and turning off the HP-75) will always turn the modem off, even if the modem is deadlocked (see "Modem Deadlock" in Section 2). When the modem is off, INITIALIZE ":MODM", MODEM OFF, and MODEM ON are the only high-level modem commands that can be executed. Any dial string saved by the D, R, or X low-level modem commands will be lost when the modem is turned off, and will have to be reentered after the modem is turned on.

- If the HP-75 STANDBY ON command is executed while the modem is on, the HP-75 will not turn itself off, and the modem will consume battery power continuously.

Errors: 151

MODEM ON

high-level modem

statement

```
MODEM ON
```

MODEM ON

Turns the modem on. MODEM ON needs only be executed once each time the HP-75 is turned on. Once on, the modem will remain on until MODEM OFF or INITIALIZE ":MODM" is executed, or until the HP-75 is turned off (or turns itself off).

When the modem is turned on, it performs a self-test. If the test fails, the modem reports %FAIL or %9. If the test passes, the modem reports %OK or %4 (%4 followed by a space—see "Turning the Modem On" in Section 2). The command character will be set to %, the modem will be in MODEM mode and on-hook, and command characters will be processed normally (see the U (UNLISTEN) command). The following parameters will be restored to their most recent settings (the letter in parentheses is the low-level command that sets the parameter):

Modem Parameters Restored at Power On

Ring count (C)
Command echoing (E)
Data format (parity) (F)
Data echoing (G)
Handshake protocol (K)
Response length (L)

Additionally, the transmit and receive buffers are cleared. If the modem is already on, however, MODEM ON has no effect—no parameters or buffers are changed.

Errors: 151, 152

MODIN$

high-level modem

function

```
MODIN$
```

D$=MODIN$

Reads data from the modem receive buffer. Returns a string containing not only the current contents of the receive buffer, but also whatever data was saved by MODOUT (see MODOUT). Returns a null string if there is no data from either location.

Errors: 16, 150, 151, 153

MODOUT

high-level modem **statement**

```
MODOUT "string"
```

```
MODOUT D$
```

Sends the string to the modem transmit buffer. If the string contains just data, the data is transmitted out the phone line. If the string includes low-level modem commands, those commands are not transmitted, but are processed locally by the modem. MODOUT also reads the modem receive buffer and saves its contents in the HP-75 to prevent the possibility of overflowing the buffer from echoed data or commands (see "MODOUT and Echoed Data" and "MODOUT and Out of Memory" in Section 2).

- MODOUT waits until the transmit buffer is ready to accept data before sending anything to the modem. If the modem is deadlocked, MODOUT will wait forever. (see "Modem Deadlock" in Section 2). Once MODOUT starts to wait, you can halt it by pressing [ATTN] (a low battery condition in the HP-75 will halt it as well). Then MODEM OFF, turning the HP-75 off, or INITIALIZE ":MODM" will clear the deadlocked condition.

  If you expect a deadlock, you can use READY? to identify that condition before MODOUT begins to wait forever. If deadlock is of great concern, your program should only send one character at a time to the modem with MODOUT, waiting before each character is sent until READY? =1.

Errors: 16, 150, 151, 153

PARITY?

high-level modem **function**

```
PARITY?
```

```
IF PARITY? THEN DISP "Parity error"
```

Checks for a parity error. Returns a 0 if the last MODIN$ had no parity error, or a 1 if there was a parity error. Remember that MODIN$ returns not only the contents of the modem receive buffer, but also any data echoed by MODOUT. PARITY? reports a parity error for data returned by MODIN$ from either source.

Errors: 150, 151, 153

PRINT

electronic disc **statement**

```
PRINT # file number [, line number] ; print list
PRINT # file number , line number
```

```
PRINT # 1,1000 ; S$
```

Prints the items in the *print list* to the electronic disc file specified by *file number* at the next or specified line. If the *print list* is omitted, deletes the specified line, but leaves the data pointer positioned at the beginning of the specified line.

- Using PRINT #, you might create a file on the electronic disc that will be too big to fit into HP-75 main memory. The maximum number of lines in a file created on the electronic disc is 10,000.

- Refer to the *HP-75 Owner's Manual* and the *HP-75 Reference Manual* for further information. Everything that applies to the use and behavior of PRINT # in main memory applies to the electronic disc as well.

Errors: 16, 28, 34, 45, 51, 62, 65, 90, 211, 212

PURGE

electronic disc **statement**

```
PURGE "pathname:XMEM"
```

```
PURGE "^>PROGRAMS>CONFIG:XMEM"
PURGE "UPLOAD:XMEM"
```

Deletes from the electronic disc all files that match the *pathname*. Does not delete directories—see PURGE DIR.

- Because pathnames allow the use of wild cards in filenames, you might purge many files inadvertently—for example, PURGE ":XMEM" will purge all files on the electronic disc. Unless you are sure which files will be deleted with PURGE, you should use REMOVE to delete files when using wild cards.

- Refer to the *HP-75 Owner's Manual* and the *HP-75 Reference Manual* for further information. Everything that applies to the use and behavior of PURGE in main memory and on a mass storage device applies to the electronic disc as well.

Errors: 62, 63, 212

PURGE DIR

electronic disc **statement**

```
PURGE DIR "location:XMEM"
```

```
PURGE DIR "^>PROGRAMS>DATACOMM>HP3000:XMEM"
PURGE DIR "HP3000:XMEM"
```

Deletes directory and child files from electronic disc. The root directory ^ cannot be purged. If you want to delete every file on the electronic disc, use INITIALIZE ":XMEM". If the current directory is deleted, the current directory will be set to ^.

- PURGE DIR is even more dangerous than PURGE, because purging a directory removes its child files and directories as well. Because of this danger, wild cards are not allowed in the directory name specified for PURGE DIR.

Errors: 62, 63, 212

READ

electronic disc **statement**

```
READ # file number [, line number] ; read list
READ # file number, line number
```

```
READ # 1,1000 ; S$
```

Reads data from an electronic disc file and assigns the data to the variables in the *read list*. If the *read list* is omitted, positions the data pointer to the beginning of the specified line.

- In certain instances when READ # generates an error condition on an electronic disc file, the data pointer will be positioned to the wrong line. For example, suppose READ # 1,N ; X produces ERROR: no data because N is an invalid *line number* for the specified electronic disc file. The next random READ # of that file (i.e., no *line number*, as in READ # 1 ; X) will read the first item on the first line of the file. If this same operation were done on a file in HP-75 main memory, the random READ # would read the first item on the line after the invalid *line number*. Rather than relying on the data pointer after an error occurs, use RESTORE # to position the data pointer to the desired location.

- Refer to the *HP-75 Owner's Manual* and the *HP-75 Reference Manual* for further information. With the above exception, everything that applies to the use and behavior of READ # in main memory applies to the electronic disc as well.

Errors: 16, 33, 34, 45, 62, 65, 90, 212

READY?

high-level modem **function**

```
READY?
```

```
100 IF READY? THEN MODOUT D$ ELSE 100
```

Checks to see if the modem transmit buffer is ready to accept data from the HP-75. Returns a 0 if modem is not ready for data, or a 1 if the modem is ready for data. READY? therefore identifies modem deadlock (see "Modem Deadlock" in Section 2).

Errors: 150, 151, 153

REMOVE

electronic disc **statement**

```
REMOVE "pathname:MEM"
```

See below for examples.

REMOVE behaves like PURGE except that you are asked specifically whether each file should be deleted. For example, if the file DOWNLOAD is to be purged, the message Purge DOWNLOAD? [N,Y,ATTN] will appear. If you press [y] or [Y], the word Yes will be echoed and the file will be

purged. If you press [n] or [N] (actually, anything other than [y] or [Y]), the word No will be echoed and the file will not be purged. If you press [ATTN], the word No will be echoed and REMOVE will end. This process is repeated for each file defined by the *pathname*.

In contrast to PURGE, the real power of REMOVE is in the use of wild cards. For example, REMOVE "--------:XMEM" will try to delete all files on the electronic disc, asking you before each one is purged. The examples below are based on the file system illustrated in "Electronic Disc Software" in Section 2.

`REMOVE "--------:XMEM" [RTN]`	Purge all files on the electronic disc
`Purge PMSCMDS? [N,Y,ATTN] Yes`	[y] pressed
`Purge SCANNER? [N,Y,ATTN] No`	[n] pressed
`Purge CONFIG? [N,Y,ATTN] No`	[ATTN] pressed, and REMOVE ends
`REMOVE "DATACOMM>--------:XMEM" [RTN]`	Purge files in the datacomm directory
`Purge TERMINAL? [N,Y,ATTN] No`	[n] pressed, and REMOVE ends since there are no more files in the datacomm directory
`REMOVE "DATACOMM>?>--------:XMEM" [RTN]`	Purge files in the datacomm directory and in any child directories. Notice that you are not asked to purge the hp3000 directory itself—only its child files.
`Purge DOWNLOAD? [N,Y,ATTN] No`	[n] pressed
`Purge UPLOAD? [N,Y,ATTN] No`	[n] pressed
`Purge TERMINAL? [N,Y,ATTN] No`	[n] pressed, and REMOVE ends since there are no more files in the hp3000 directory

- REMOVE can also be used for files in HP-75 main memory. This is the only time wild cards can be used to reference filenames in HP-75 main memory.

Errors: 62, 63, 212

RENAME

electronic disc **statement**

```
RENAME "old pathname[:XMEM]" TO "new pathname:XMEM"
RENAME "old pathname:XMEM" TO "new pathname[:XMEM]
```

See below for examples.

Changes the name of a file or directory on the electronic disc. RENAME can also move a file to a new location, or move a directory and all its child files to a new location. The *old pathname* defaults to the current directory. The root directory ^ cannot be renamed. The examples below are based on the file system illustrated in "Electronic Disc Software" in Section 2.

```
RENAME "^>PROGRAMS>DATACOMM>HP3000>UPLOAD:XMEM" TO "SEND"
```

Renames UPLOAD to SEND

```
RENAME "UPLOAD:XMEM" TO "SEND"
```

Renames UPLOAD to SEND

```
RENAME "DATACOMM:XMEM" TO "REPORTS>:XMEM"
```

Moves the entire subtree starting at the datacomm directory to a subtree under the reports directory

- Refer to the *HP-75 Owner's Manual* and the *HP-75 Reference Manual* for further information. Everything that applies to the use and behavior of RENAME in main memory and on a mass storage device applies to the electronic disc as well.

Errors: 62, 63, 64, 65, 68, 212, 213

RESTORE

electronic disc **statement**

```
RESTORE # file number [,line number]
```

```
RESTORE # 1,1000
```

Moves the data pointer in the data file specified by *file number* to the beginning of the specified or lowest-numbered statement in the file.

- Refer to the *HP-75 Owner's Manual* and the *HP-75 Reference Manual* for further information. Everything that applies to the use and behavior of RESTORE # in main memory applies to the electronic disc as well.

Errors: 34, 45, 62, 65, 212

XCAT$ electronic disc function

```
XCAT$(number)
```

```
S$=XCAT$(7) @ DISP "File is on level ";S$[10,10]
```

Returns the catalog of the *number*th file or directory on the electronic disc. The *number* of the file or directory is the position of the file in a CAT ALL ":XMEM" listing. If *number* is greater than the number of files and directories on the electronic disc or less than 0, XCAT$ returns a null string. The root directory is number 0. XCAT$(0) returns a catalog that identifies when the electronic disc was last initialized, since the root directory is only created at that time. The format of the returned string is the same as the HP-75's CAT$(*number*) function, except that the tenth character of the XCAT$ string has the level of the file or directory (note that directories also have the level in the length field at character 17).

Errors: 212

XDIR$ electronic disc function

```
XDIR$
```

```
S$=XDIR$
```

Returns the catalog of the current directory on the electronic disc.

Errors: 212

XMEM electronic disc function

```
XMEM
```

```
M=XMEM
```

Returns the amount of memory remaining on the electronic disc.

Errors: 212

LOW-LEVEL MODEM COMMAND DICTIONARY

SECTION 4

This section describes all the low-level modem commands provided by the HP 82718A Expansion Pod. First, the name of the command is given. The next line lists the syntax of the command and its expected parameters. An example of the use of the command follows. The description of the command appears next, and includes any cautions, set off by bullets, that should be observed when using the command. Finally, a list of possible responses that may be reported by the command is given.

Notes on syntax: Optional parameters are surrounded with brackets ([ ]). All examples assume the command character is **%**.

A (ANSWER)

```
A
```

```
MODOUT "%A"&CHR$(13)
```

Places the modem in ANSWER mode, takes the modem off-hook, transmits the answer carrier tone, and waits about 20 s for an originate tone. If the modem receives a originate tone before the 20 s expires, the CONNECT response is reported. If not, NO ANS is reported and the modem goes back on-hook and into MODEM mode. If the modem receives a carrier that is not in the answer band, it reports BAD ANS, and goes back on-hook and into MODEM mode.

Responses: 1, 2, B

B (BREAK)

```
B duration
```

```
MODOUT "%B 4"&CHR$(13)
```

Sends a break (binary zero, or long-space disconnect). The *duration* is a single hex digit (0-F) that specifies the number of 250-ms intervals during which the break will be sent (0-15 intervals, which equals 0 to 3.75 s). To send a break longer than 3.75 s, use more than one B command in the command string. If the modem is in VOICE mode, the B command will just wait for the duration of the break.

Responses: none

C (COUNT)

```
C ring count
```

```
MODOUT "%C 2"&CHR$(13)
```

Sets the *ring count*. The *ring count* is a single hex digit (0-F) that specifies either the number of rings (0-15) to wait before answering an incoming call, or the number of rings to wait before aborting a dialing operation. The *ring count* is set to 0 when the modem is initialized, and its current value is preserved while the modem is off.

If the modem is on-hook and in MODEM mode, it will answer an incoming call after *ring count* rings. After each incoming ring, the modem will report RING IN and OK. After *ring count* rings have been detected, the modem will automatically report RING IN, go off-hook (i.e., answer the incoming call, and enter ANSWER mode. The behavior at this point is as if the A (ANSWER) command was executed. See the A command for details.

If *ring count* is zero, the modem will be ORIGINATE-ONLY when in MODEM mode; each incoming ring will be reported, but the incoming call will not be answered. Only if *ring count* is a number other than zero will the modem be AUTO-ANSWER when in MODEM mode, and answer the incoming call.

If the modem is on-hook and in VOICE mode, it will not answer incoming calls. However, after each incoming ring, the modem will report RING IN and OK, regardless of the value of *ring count*. The F (PICKUP) command can be used to answer the incoming call in VOICE mode.

While the modem is counting incoming rings, any command sent to the modem will cause the modem to restart counting incoming rings (in effect, extending the *ring count*).

The *ring count* is also used when dialing. When dialing, the modem will wait *ring count* + 4 rings before aborting the dialing operation. Refer to the D (DIAL) and R (REDIAL) commands.

Responses: none

D (DIAL)

```
D [dial string]
```

```
MODOUT "%D I9W 1(800)555-1212"&CHR$(13)
```

Takes the modem off-hook and dials a sequence of numbers. If you have already stored a dial string using one of the the three D, R (REDIAL), or X (XMIT), the string is an optional parameter; the D command will dial the most-recently entered dial string. If no dial string has been stored since the modem was last turned on or initialized, or if the last dialing command containing a dial string was aborted, the dial string is a required parameter.

A dial string can contain any number of characters, but only the first 32 recognized characters will be dialed or stored. The dial string recognizes several special characters, listed below. All other characters are ignored except for the characters related to low-level command strings: BS (backspace—ASCII 8 or control-H), CAN (cancel—ASCII 24 or control-X), comma, and carriage return.

Characters	Meaning
0-9,#,*	Digits to be dialed.
I	The following digits, until the next P or W, should be dialed using interrupt (pulse) dialing. The I should be preceded by a P if you want the modem to detect no dial tone and report the NO DIAL response.
P	Wait for dial tone, then resume dialing using adaptive dialing. If no dial tone is detected within the timeout, the dial command will be aborted, and the NO DIAL response will be reported. The dial string will be executed as if the first character was a P unless it has a P or W as its first character. The timeout is 3 s for an implied P, and 4.5 s for a P that actually appears in the dial string. These times consist of a 2 s delay, followed by detection of a continuous dial tone for the remainder of the timeout.
W	Wait for 5 s, then resume dialing next part of string using tones.

After the modem dials, it begins to look for a busy signal or a ring. The modem response depends on the mode it was in when dialing commenced. If the modem was in VOICE mode, it waits until it determines that the line is not busy, then responds with CONNECT. It may respond with RING first. Then the modem will remain off-hook regardless of the status of the called phone (which will continue to ring). If the line is busy, the modem goes on-hook and responds with BUSY.

If the modem is in MODEM mode, it dials as above, then continues to monitor the status of the line. If a busy signal is detected, it hangs up and responds with BUSY. If a carrier is detected, the modem enters ORIGINATE mode. The behavior at this point is as if the O (ORIGINATE) command was executed. See the O command for details.

If command echoing is enabled (E 0), the modem will echo the dial string when it receives the command string containing it. The dial string will be echoed in exactly the same form as it appeared in the command string. The modem will echo the dial string again each time it is dialed by any of the three dialing commands. The D and X commands echo the string once (unless they are repeated in the command string—see below). The R command echoes it up to 10 times as it redials.

The dial string echoed during dialing will be in a different form than in the command string. Characters that are ignored in the dial string are not echoed. The special characters P and W will not be echoed, but will break the dial string into segments separated by spaces. Echoing will stop for the duration of the waits imposed by those characters. The special character I will not be echoed, but will cause all digits between the I and the next P or W (or the end of the dial string) to be echoed. The digits # and * will be echoed as ; and <. There will be a trailing space but no carriage return or line feed after the entire dial string has been echoed.

A low-level command string can contain multiple dialing commands, but it probably will not have the behavior you expect. All the dialing commands are counted, and their dial strings are concatenated together. If the dialing operation results in a BUSY response, the operation will be repeated for the number of times that there are dialing commands, up to a maximum of ten times per command (D, R, or X). Because X depends on VOICE or MODEM mode, and only partially checks line status, multiple X

commands may behave in an unexpected manner. In general, the behavior of multiple dialing commands is unusual enough that it should be avoided.

Responses: 0, 1, 2, 3, 5, B

E (ECHO)

```
E echo specifier
```

```
MODOUT "%E 1"&CHR$(13)
```

Enables and disables echoing of low-level modem commands. E 0 enables command echoing—all low-level commands will be echoed back to the HP-75 by the modem (actually echoed into the receive buffer for MODIN$ to read). E 1 disables command echoing (the carriage return that ends the command will still be echoed as a carriage return-line feed). The *echo specifier* is set to 0 when the modem is initialized, and its current value is preserved while the modem is off.

Responses: none

F (FORMAT)

```
F format specifier
```

```
MODOUT "%F 3"&CHR$(13)
```

Sets the data format—parity and data length. The *format specifier* is a single digit (0-4) that specifies the parity and data length. The *format specifier* is set to 4 when the modem is initialized, and its current value is preserved while the modem is off. The *format specifier* has these meanings:

Format Specifier	Parity	Data Length
0	Always 1 (mark)	7 bits
1	Always 0 (space)	7 bits
2	Odd	7 bits
3	Even	7 bits
4	None	8 bits

The modem generates parity for all data that is transmitted according to the above table. If the high-order bit of a data byte is set, the parity will be generated with the opposite sense. Consequently, if you need to send legitimate 8-bit data, you should only use F 4, and you will not be able to check parity.

The modem checks for parity errors in all data received for *format specifiers* 0, 1, 2, and 3. If a parity error is detected, it can be identified with the PARITY? high-level modem command. If bit 3 of K2 (the second KONTROL parameter) is clear, parity errors will also be flagged by setting the high-order bit of the bad data byte. In that case the bad data will appear underlined in the HP-75 display or in inverse

video on a video monitor. If bit 3 of K2 is set, parity errors will not be flagged in that manner—the high-order bit will be cleared for all 7-bit data received. Only PARITY? can identify parity errors if bit 3 of K2 is set.

Responses: none

G (GIVEBACK)

G *giveback specifier*

```
MODOUT "%G 1"&CHR$(13)
```

Enables and disables echoing of data. G 0 disables data echoing. In this state, data will be echoed back to the HP-75 only if the host computer is "full duplex". G 1 enables data echoing for operation with "half duplex" computers. All data will be echoed back to the HP-75 by the modem (actually echoed into the receive buffer for MODIN$ to read). The *giveback specifier* is set to 0 when the modem is initialized, and its current value is preserved while the modem is off.

Responses: none

H (HANGUP)

H

```
MODOUT "%H"&CHR$(13)
```

Hangs up the modem. Places the modem on-hook and ends the communication session. The H command takes a total of 3 s to place the modem on-hook. If the modem is in MODEM mode, H sends a break for 2 of the 3 s, then places the modem on-hook.

Responses: none

I (INITIALIZE)

I

```
MODOUT "%I"&CHR$(13)
```

Resets the modem to its default state, shown in the table with the INITIALIZE ":MODM" high-level modem command in the "Keyword Dictionary". If command echoing is enabled (E 0), I is echoed as this command string:

```
C 0,E 0,F 4,G 0,H,K E03QSQSQ,L 0,M,N %
```

Note: Q represents control-Q (DC1) and S represents control-S (DC3).

- The I command will not reset the modem to its default state if the modem is deadlocked (see "Modem Deadlock" in Section 2).
- INITIALIZE ":MODM" must be used instead of the I (INITIALIZE) low-level modem command if the command character is not known or if U 0 has been executed (see the U (UNLISTEN) command).

Responses: none

K (KONTROL)

K *handshake protocol specifier*

```
C$=CHR$(17)&CHR$(19)&CHR$(17)&CHR$(19)&CHR$(17)
MODOUT "%K E03"&C$&CHR$(13)
```

Sets the serial handshake protocol. The *handshake protocol specifier* consists of 8 single-byte parameters, called K1-K8. The first three parameters determine the handshake protocol, parity error behavior, and receive buffer block size. Each of these three parameters are single hex digits (0-F), defined by setting or clearing its component bits according to the particular requirements of your application. All the bits taken together constitute the single hex digit for that parameter. (In the descriptions below, note that bits are numbered 0 to 3, from right to left.) The last five parameters are definable characters that are used in conjunction with the different handshake protocols. Each of the last five parameters can be any 7-bit ASCII character except BS (backspace—ASCII 8 or control-H), CAN (cancel—ASCII 24 or control-X), comma, or carriage return. Note that since these are 7-bit characters, they will not be recognized as protocol or strip characters if they appear with a parity error.

Character	Bit	Description
K1	bit 3	Transmitter XON/XOFF Protocol. A received K5 character will halt data transmission by the modem, and a received K4 character will restart data transmission by the modem.
	bit 2	Receiver XON/XOFF Protocol. When the number of bytes in the receive buffer exceeds the block size, the modem will transmit a K7 character. When the number of bytes in the receive buffer drops to zero, the modem will transmit a K6 character.
	bit 1	Receiver ENQ/ACK Protocol. The modem will respond to a received ENQ with an ACK when its receive buffer becomes empty.
	bit 0	Transmitter ENQ/ACK Protocol. When transmitting, the modem will transmit an ENQ character after sending *block size* characters, and stop transmitting. The modem will resume transmitting when an ACK is received.

K2	bit 3	Strip Parity. If this bit is set, the modem will strip off the parity bit of incoming data bytes for all 7-bit data formats (F 0-3). If this bit is clear, the modem will show parity errors by setting the parity bit for any data bytes with a parity error.
	bit 2	Strip K8. If this bit is set, the modem will strip off any received K8 characters from the received data stream.
	bit 1	Strip DEL. If this bit is set, the modem will strip off any received DEL (ASCII 127) characters from the received data stream.
	bit 0	Strip NUL. If this bit is set, the modem will strip off any received NUL (ASCII 0) characters from the received data stream.
K3	bit 3	Prompt Mode. Whenever the modem transmits a CR, it will stop transmitting until it receives a K4 character from the remote computer. If K4 is not the same as K5, the modem will strip off any K5 characters received between the carriage return and the K4 character. Prompt Mode has higher priority than Transmitter XON/XOFF Protocol.
	bits 2-0	Buffer Partition. The receive buffer *block size* is 8*(1+*buffer partition*). The *block size* is used in Receiver XON/XOFF Protocol to specify when the modem should transmit a K5 character, and in Transmitter ENQ/ACK Protocol to specify how often the modem should transmit an ENQ character. If the receive buffer overflows, received data is ignored until the number of bytes in the receive buffer drops below 56. Then the buffer overflow response (OV) is placed in the buffer, and receiving continues.
K4		Transmitter XON/XOFF Ready Character and Prompt Character.
K5		Transmitter XON/XOFF Not Ready Character.
K6		Receiver XON/XOFF Ready Character.
K7		Receiver XON/XOFF Not Ready Character.
K8		Strip Character.

All eight parameters taken together constitute the *handshake protocol specifier*. The eight parameters are set to E03QSQSQ when the modem is initialized, where Q represents control-Q (DC1) and S represents control-S (DC3). Their current values are preserved while the modem is off.

- The modem will behave unpredictably if K4-K8 are defined as the command character.
- The K command specifies handshake protocols that can cause modem deadlock. If you are using Transmitter XON/XOFF, Transmitter ENQ/ACK, or Prompt Mode protocols, you should write your program so that it can detect and respond to deadlock. K will not change protocols if the modem is deadlocked (see "Modem Deadlock" in Section 2).

Responses: none

L (LONG)

```
L response specifier
```

```
MODOUT "%L 1"&CHR$(13)
```

Specifies long or short echoing of low-level modem responses. L 0 specifies long echoing—all responses are spelled out and followed by a carriage return-line feed. L 1 specifies short echoing—all responses are single ASCII hex digits between 0 and B.

The low-level responses, in the two lengths, are as follows:

Long Response	Short Response
ABORT	A
BAD ANS	B
BUSY	0
CONNECT	1
CONN LOST	8
FAIL	9
NO ANS	2
NO DIAL	3
OK	4
OV	OV
RING	5
RING IN	7

The *response specifier* is set to 0 when the modem is initialized, and its current value is preserved while the modem is off. See appendix D, "HP 82718A Errors and Responses", for the meanings of the responses, commands that report them, and the exact sequence of characters that will be reported.

Responses: none

M (MODEM)

```
M
```

```
MODOUT "%M"&CHR$(13)
```

Places the modem in MODEM mode. The modem will monitor the phone line to determine the line status. If the modem was already in MODEM mode and a valid data connection was already established when this command was entered, then the modem will return the CONNECT response.

If the modem was in VOICE mode and the voice connection had already been made, the modem will enter ORIGINATE mode. The behavior at this point is as if the O (ORIGINATE) command was executed. See

the O command for details. If the modem was in VOICE mode and no voice connection had yet been made, the modem will enter MODEM mode and remain on-hook. In this case, no response is returned.

- If you attempt to enter MODEM mode when the modem is off-hook, the modem will enter either ANSWER or ORIGINATE mode depending on whichever was the last mode the modem operated in. When off-hook, either hang up the modem before executing M, or use the A (ANSWER) or O (ORIGINATE) commands to define explicitly the off-hook mode you want to enter.

Responses: 0, 1, 2, 3, 5, B

N (NEW)

N *command character*

```
MODOUT "%N !"&CHR$(13)
```

Changes the low-level modem command character to a new value. The command character can be any 7-bit ASCII character except BS (backspace—ASCII 8 or control-H), CAN (cancel—ASCII 24 or control-X), comma, and carriage return. The command character is set to **%** when the modem is turned on or initialized, and is *not* preserved while the modem is off.

Responses: none

O (ORIGINATE)

O

```
MODOUT "%O"&CHR$(13)
```

Takes the modem off-hook and places it into ORIGINATE mode. The modem then monitors the phone line to determine its status. If the modem was already off-hook and detects an answer tone, it issues an originate tone and reports CONNECT. If no answer tone is detected within about 50 s, the modem reports BAD ANS (out-of-frequency answer tone) or NO ANS (no dial tone or an unsteady tone). If the modem hears a dial tone when the O command is executed, it will wait about 7 s, then report BAD ANS. If the modem detects a busy signal from a previous dialing operation, it reports BUSY.

If the modem detects a ring from a previous dialing operation, it starts counting rings and reports RING after each one. If the maximum number of rings is exceeded (the ring count + 4, up to a limit of nine), or if no answer tone is received within 50 s after the last dialing operation ended, the modem reports NO ANS. If an answer tone is detected, the modem issues an originate tone and reports CONNECT.

After the BAD ANS, BUSY, or NO ANS responses are reported, the modem goes on-hook and into MODEM mode.

The significance of the BAD ANS and NO ANS responses as they relate to dialing is explained in "Adaptive Dialing" in Section 2.

Responses: 0, 1, 2, 5, B

P (PICKUP)

```
P
```

```
MODOUT "%P,X I9W 1(800)555-1212"&CHR$(13)
```

Takes modem off-hook (picks it up) and places it in VOICE mode. The P command can be used either to set the modem into the proper mode for the X (XMIT) command, or to answer an incoming call in VOICE mode.

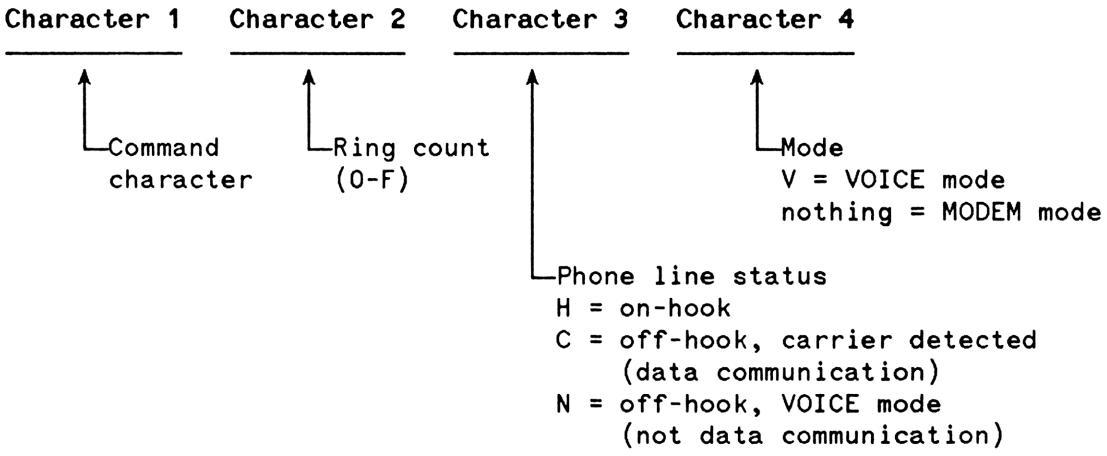
Responses: none

Q (QUERY)

```
Q
```

```
MODOUT "%Q"&CHR$(13)
```

Asks the modem for information on ring count, data communication status, and operating mode. The modem returns this status information in the form of two or three single-character codes, preceded by the command character, as follows:



Any parameter or mode changes are executed after the command string is terminated. Therefore, multiple Q commands in a command string will usually be redundant. The only exceptions are commands that alter the on-hook/off-hook state, since they are executed as they appear in the command string. If multiple Q commands are intertwined with those commands, the responses will reflect the status of the modem at the time each Q is executed.

- If the modem enters VOICE mode with the P (PICKUP) or V (VOICE) commands during a data communication session, Q will return a temporary response of the form 0CV (carrier detected, in VOICE mode) for as long as the remote modem is responding with its carrier. This response seems to contradict the definition of VOICE mode, since VOICE mode and data communications are designed

to be mutually exclusive. No data will be transmitted or received in this state. Since changing to VOICE mode halts any carrier tone being issued by the modem, and the remote modem will end its carrier when the HP 82718A's is lost, the data communication session will end, and the unusual response will revert to the form ONV shortly after it appears.

Responses: see above

R (REDIAL)

R [*dial string*]

```
MODOUT "%R I9W 1(800)555-1212"&CHR$(13)
```

Redials the last dialed number. Behaves exactly like the D (DIAL) command unless the called phone is busy. Then, instead of hanging up after reporting the BUSY response, the R command automatically redials the *dial string* at intervals of about 40 s until the number is not busy, or until ten dials have been attempted. See the D command for further details and for the syntax of a *dial string*.

You can string multiple R commands together (up to 10, as described for the D command) to redial up to 100 times.

Responses: 0, 1, 2, 3, 5, B

T (TEST)

T *test number*

```
MODOUT "%T 1"&CHR$(13)
```

Instructs the modem to perform self-tests. The *test number* is a single digit (0-2) that specifies one of three different tests. The modem mode and parameters are not affected by any of the different tests, except that the data in the transmit and receive buffers may be lost, and the modem will be on-hook after test 1 is completed.

Test 0 is the hardware integrity test. The modem goes on-hook and executes the power-up self-test. As when the modem is turned on, the modem will report OK if the self-test is successful (the OK reported after a command is executed, not the special OK reported when the modem is turned on). If the test fails, the FAIL response is reported, and the test is repeated. The test will continue to report FAIL and will be repeated until the modem is turned off or the test passes.

Test 1 is the analog loopback test. The modem goes off-hook, connects its analog output to its analog input, and switches the receive circuitry so that it receives and transmits in the same frequency band. The modem will route all data through its analog circuitry and back to the HP-75 (actually echoed into the receive buffer for MODIN$ to read) until it receives any low-level modem command.

Test 2 is the local digital loopback test. The modem goes off-hook and retransmits any data that it receives over the phone line. Data from the HP-75 will be ignored. If the modem is in MODEM mode, it tries to establish a data connection by issuing either an answer or an originate tone, depending on whether

the last mode the modem operated in prior to entering MODEM mode was ANSWER or ORIGINATE mode. If it cannot establish a connection within about 4 s, CONN LOST is reported.

If the modem is in VOICE mode, it will issue a continuous answer or originate tone, depending on whether the last mode the modem operated in prior to entering VOICE mode was ANSWER or ORIGINATE mode. It will issue this tone until you end the test by sending the modem any low-level modem command except Q (QUERY).

You can use test 2 to test the modem in MODEM mode as either the originating or the answering modem. To exercise another modem, enter VOICE mode before executing test 2 so the modem issues continuous tones. `MODOUT "%A,H,V,T 2"&CHR$(13)` will cause the modem to issue a continuous answer tone, and `MODOUT "%O,H,V,T 2"&CHR$(13)` will cause the modem to issue a continuous originate tone.

- The T command will not perform any modem tests if the modem is deadlocked (see "Modem Deadlock" in Section 2).

Responses: 8, 9

U (UNLISTEN)

U *number of bytes*

```
MODOUT "%U 1"&CHR$(13)&"%"
```

Allows sending the command character to the modem as if it were data. After the U command has been executed, the modem will not monitor the transmit data stream for the command character—that is, any command characters are passed on without invoking a low-level modem command. The modem goes back to normal operation after *number of bytes* characters have been transmitted. The *number of bytes* is 1 to 4 hex digits (1-FFFF), specifying 1 to 65,535 consecutive bytes that will not be monitored for the command character.

See also "Sending Command Characters as Data" in Section 2.

- If *number of bytes* is zero, the modem will never monitor the transmit data stream for the command character. To regain the ability to process low-level modem commands, the modem must either be initialized using `INITIALIZE ":MODM"` or turned off.

Responses: none

V (VOICE)

V

```
MODOUT "%V"&CHR$(13)
```

Places the modem in VOICE mode. In VOICE mode, no originate or answer tones are placed or looked for on the telephone line, and the modem does not monitor the line for rings. Any tones or carriers already

being issued will be halted. VOICE mode is for voice communication where the modem is used as an auto-dialer only. In VOICE mode, you can dial phone numbers using the X (XMIT) command. If the modem was off-hook, it will remain off-hook.

Responses: none

X (XMIT)

X [*dial string*]

```
MODOUT "%P,X I9W 1(800)555-1212"&CHR$(13)
```

Dials a number without call progress reporting after dialing. The X command is similar to the D (DIAL) and R (REDIAL) commands in that it dials a *dial string*. However, that is all that X does. Unlike the D and R commands, it only reports three responses: BUSY if the line is busy, CONNECT if there is no busy signal, and NO DIAL if there was no dial tone.

The X command simplifies the use of the modem as an auto-dialer in VOICE mode. If the D command is used for auto-dialing, it will still report all its responses (except BAD ANS and NO ANS), even when they are irrelevant to a voice connection. Using X eliminates almost all phone line monitoring before, during, and after the dialing operation. The only exception is that an implied or explicit P in the *dial string* will check for a dial tone before dialing, and report NO DIAL if there is none.

The X command only works in VOICE mode and if the modem is off-hook. The simplest way to put the modem in VOICE mode and take it off-hook is with the P (PICKUP) command. Consequently, a common command string when using X is P,X followed by the *dial string*, as in the example.

Responses: 3

Z (ZAP)

Z

```
MODOUT "%Z"&CHR$(13)
```

Clears the modem receive buffer. If the command string contains more than just the Z command, the buffer will be cleared after all the other commands in the string have been executed and their responses have been placed in the receive buffer. Once the buffer has been cleared, the normal response to any completed command (OK or 4) is placed in the receive buffer. Consequently, Z does not leave the buffer completely empty.

Also, the Z command does not clear the HP-75 buffer containing data saved by MODOUT (see MODOUT). Any characters saved by MODOUT before Z is executed will still be returned by MODIN$. Because of this, and because of the appearance of the normal command response, the most effective method for clearing the receive buffer is to use MODIN$ until it returns a null string.

- The response to the Z command is almost the same response that is reported when the modem is turned on—%OK without a leading or trailing CR LF for the long response, or %4 without a leading

CR LF for the short response. Note that the power on response will always have % as the command character, but the Z response will have the current command character.

Responses: see above

OWNER'S INFORMATION

APPENDIX
A

SERIAL NUMBER

Each HP 82718A carries an individual serial number on the bottom of the unit. We recommend that you keep a separate record of this number. Should your unit be lost, the serial number is often necessary for tracing and recovery, as well as for insurance claims. Hewlett-Packard does not maintain records of individual owners' names and unit serial numbers.

ENVIRONMENTAL LIMITS

Observe the following temperature and humidity limits of the HP 82718A.

- Operating Temperature: 0 to 45 C (32 to 113 F)
- Recharging Temperature: 10 to 40 C (50 to 104 F)
- Storage Temperature: -40 to 55 C (-40 to 131 F)
- Operating and Storage Humidity: 0 to 95% relative humidity

Your HP 82718A should never be operated or stored outside of the specified range. High temperatures are especially damaging to the HP 82718A and its batteries. Temperature cycling—from one extreme to another—will cause stresses in your HP 82718A that may also to decrease its reliability. Normal room temperature will give maximum reliability (that is, the smallest likelihood of failure).

OPERATING PRECAUTIONS

Certain electronic circuits in the HP 82718A operate continuously, and they are susceptible to disruption or damage at all times. Disruption or damage may be caused by installing or removing the battery packs or the HP-75 while the ac adapter/recharger is connected, by electrostatic discharge to the unit, or by strong magnetic fields. Observe the precautions listed below.

CAUTION

Hold or touch the HP 82718A while preparing to install the battery packs or the HP-75. Touching the insert on the bottom of the HP 82718A is particularly effective for neutralizing an electrostatic charge.

Do not place your fingers, tools, or other foreign objects into the modem's modular jacks or the wand connector or ROM ports on the HP-75D.

Turn off the unit before installing or removing the battery packs.

HP 82718A SPECIFICATIONS

General Specifications

Dimensions (including the HP-75): 16.51 cm (6.5 in) x
29.21 cm (11.5 in) x
5.72 cm (2.25 in)

Weight (including the HP-75): 1.60 kg (3.53 lb)

Modem Specifications

Transmission Mode: serial, asynchronous, full duplex
Handshake: XON/XOFF, ENQ/ACK, none
Transmit Buffer: 8 bytes
Receive Buffer: 64 bytes
Modulation: FSK
Baud Rate: 300
Transmit Frequencies (Hz): 1070, 1270 (originate mode)
2025, 2225 (answer mode)
Receive Frequencies (Hz): 2025, 2225 (originate mode)
1070, 1270 (answer mode)
Data Bits: 7 or 8
Start Bits: 1
Stop Bits: 1
Parity: odd, even, always 1, always 0, none
Modes: originate, answer, auto-answer
Modem Compatibility: Bell 103/113
Dialing: automatic or manual, tone or pulse
Tone Dialing:
- Tone Duration: 65-75 ms
- Intertone Delay: 65-75 ms

Pulse Dialing:
- Pulse Rate: 10-11 pulses/s
- Interdigit Delay: 700-800 ms

Ringer Equivalence: 0.8 B
Jacks: two RJ11C

Electronic Disc Specifications

Memory Size: 32 kbytes (Option 032) or 64 kbytes (Option 064)

Bar Code Specifications

The HP-75D and the bar code software is designed for use with the HP 92267A and HP 92267B Digital Bar Code Wands. However, you can use any wand that has all of the following:

- Three-pin Hypertronics D01 connector
- Operation on a single 5-volt power supply
- Digital output compatible with TTL and CMOS.
- Signal polarity: high on non-reflective, low on reflective
- Less than 50 ma peak current

Switched wands are recommended for use with the HP-75D in order to conserve battery life.

The bar code software will reliably decode bar code with these specifications:

- Minimum element width (high resolution wand) of 0.005 in (0.13 mm)
- Minimum element width (medium resolution wand) of 0.0075 in (0.19 mm)
- Ratio of wide to narrow elements between 2:1 and 3:1
- Minimum scan speed of 3 in/s (7.6 cm/s)
- Maximum scan speed of 30 in/s (76 cm/s)

POWER SUPPLY INFORMATION

Power Consumption

The power consumption of the HP-75D and the HP 82718A is shown in the following table:

Operating Condition	Power Supply Current	Modem Adds	Wand Adds
Off	120 ua	0 ma	0 ma
On, idle	24 ma	100 ma	0 ma
Running program or STANDBY ON	35 ma	100 ma	45 ma
Accessing electronic disc	45 ma	100 ma	0 ma

These numbers represent typical values for power supply current at room temperature (25 C). Your HP-75D and HP 82718A may consume more or less, depending on the individual electronic components and the operating temperature.

A fully charged pair of battery packs typically allow more than 10 days of operation using the following duty cycle: HP-75D and HP 82718A running a program for 1 hr/day, with the wand in use 20% of the time, and the modem in use 5% of the time. Note that this is based on the charge level of the batteries, room temperature (25 C), and the typical operating current requirements shown in the above table.

The HP 82718A battery packs are charged by connecting the HP-75 to the ac adapter/recharger. Turn the HP-75 off (press [SHIFT] [ATTN]) before connecting the unit to a power outlet. This will prevent unexpected voltage "spikes" from disturbing the contents of the electronic disc. When the HP-75 is connected to a power outlet, the HP 82718A uses the battery packs only as a backup power supply and normally doesn't draw any power from the batteries.

You won't damage the HP 82718A by using it without one of its two battery packs (see the caution below), but you will only be able to use it for half as long between charges.

CAUTION

The HP 82718A must have at least one battery pack installed at all times to guarantee proper operation. Be sure to keep a battery pack in the section of the battery compartment that is marked with the warning PROPER OPERATION REQUIRES BATTERY IN THIS LOCATION.

Low Battery Safeguards

The HP-75 has two low-power safeguards to protect the contents of the electronic disc.

After the first indication of low power, either recharge or replace the battery packs as soon as convenient.

- When the battery voltage drops below a predetermined level, the BATT annunciator turns on, signifying from 5 minutes to 2 hours more of operation, depending on the condition of the batteries. This time is based on running a program. If the electronic disc is being accessed, the modem is on, or the wand is being used, the time will be drastically shortened.

- If the voltage continues to drop after this grace period, the HP-75 aborts its normal operations, including any currently executing program, electronic disc access, or modem operation, and turns itself off. However, the battery packs continue to power the electronic disc circuits. If the [ATTN] key or a due appointment causes the HP-75 to turn back on, the HP-75 briefly displays the following message:

  ```
  WARNING: low batteries
  ```

  Immediately afterwards, the HP-75 turns off again. At this point, to prevent a complete reset, don't press the [ATTN] key before connecting the unit to an ac outlet or before replacing the battery packs. In this nearly discharged condition, the HP 82718A will preserve the information on the electronic disc for a few days as long as no one attempts to turn the unit back on.

If the battery pack nears total discharge, the HP-75 will reset itself and lose everything in memory. Independently, the HP 82718A will reset itself and lose everything on the electronic disc. Resets may also occur if an attempt is made to turn the unit on (by pressing [ATTN]) or to acknowledge an alarm on low batteries.

Battery Recharging

The nickel-cadmium battery packs are being charged whenever the HP-75 is connected to an ac outlet using an ac adapter/recharger. A normal charging time from a fully discharged pair of battery packs to full charge is about 8 hours. Shorter charging periods will reduce the operating time you can expect from a single battery charge. The battery packs are never in danger of becoming overcharged. However, if you recharge the battery packs outside of the recharging temperature range, you may seriously decrease the life of the battery packs.

If you operate the HP-75 from ac voltage continuously for long periods, the nickel-cadmium batteries in the rechargeable battery packs will eventually reach the point at which they will no longer recharge to their full capacity. Consequently, it's advisable every few months to disconnect the ac adapter/recharger and let the rechargeable batteries discharge through normal use of the HP-75 and the HP 82718A until the BATT annunciator lights. Then plug in the ac adapter/recharger to recharge the nickel-cadmium batteries to their full potential.

Replacing the Battery Packs

CAUTION

When the HP 82718A battery packs are removed, the electronic disc circuits are protected against loss for a minimum of 30 s. You may wish to copy critical files to a mass storage medium before removing the battery packs to avoid the possibility of losing important information if a reset occurs.

Here is how to replace the battery packs:

1) Turn off the HP-75. Press [SHIFT][ATTN] or type BYE [RTN].

2) Connect the ac adapter/recharger to the HP-75.

3) Turn the HP 82718A upside down and lay it on a flat, smooth surface.

4) Unlatch the HP 82718A battery door by sliding both ribbed latches away from the edge of the case, and remove the door.

5) Remove each battery pack by pressing down on the edge of the pack nearest the latches, and lifting up on the opposite end.

6) Align the contacts of the replacement battery packs with the correct spring contacts and set them into the battery compartment.

7) Replace the compartment door and secure its two latches.

The nickel-cadmium battery packs use the two parallel contacts in each section of the battery compartment. If the battery packs are inserted backwards, neither the HP-75 nor the HP 82718A will be damaged, although the HP-75 and the electronic disc will reset if turned on in this condition.

Rechargeable Battery Care

You will know that the battery packs are recharging if the battery door on the HP 82718A is warm to the touch. After 8 or more hours of recharging, the battery packs should be fully charged. If rechargeable battery packs seem to discharge very quickly, the problem may be that the battery pack has been charged repeatedly at shallow levels due to the low current drain of the HP 82718A. In that case, disconnect the unit from its power outlet, type STANDBY ON [RTN] in EDIT mode, and leave the HP-75 and HP 82718A turned on until the battery voltage drops to the point at which the HP-75 suspends operations and turns itself off (as described in "Low Battery Safeguards"). Then reconnect the unit to a power outlet for at least 10 hours to recharge the battery pack fully.

WARNING

Do not attempt to incinerate or mutilate battery packs—the packs may burst or release toxic materials.

Do not connect together or otherwise short-circuit the battery pack terminals—the packs may melt or cause serious burns.

The battery packs are warranted for 1 year and will be replaced if they don't work properly during the warranty period. Return defective packs to Hewlett-Packard according to the shipping instructions later in this section. (If you are in doubt about the cause of the problem, return the complete HP-75 and HP 82718A along with the battery packs and the ac adapter/recharger.) If the battery packs are out of warranty, see your nearest dealer to order replacements.

GENERAL CLEANING INFORMATION

The HP 82718A can be cleaned with a soft cloth dampened either in clean water or in water containing a mild detergent. Don't use an excessively wet cloth or allow water inside the HP 82718A. Avoid abrasive cleaners.

VERIFYING PROPER OPERATION

If you suspect that your HP 82718A is not operating properly and may require service, troubleshoot your unit this way:

1) Turn the HP-75 off. Press [SHIFT] [ATTN] or type BYE [RTN].

2) Plug the unit into a power outlet using the ac adapter/recharger.

3) Turn the unit back on by pressing [ATTN]. The display should show a blinking cursor and either the BASIC prompt (>) or the text prompt (:). If MEMIC didn't select appears, the HP 82718A has a hardware failure and requires service.

4) To check the operation of the modem, type: INITIALIZE ":MODM" @ MODEM ON @ WAIT 10 @ DISP MODIN$ [RTN]. Approximately 30 s later, you should see %OK appear in th display. If this operation takes significantly longer than 30 s, the modem has failed its self-test,

and the HP 82718A requires service. Pressing [ATTN] will halt the execution of the command. Also, if you see %FAIL, the modem has failed its self-test, and the HP 82718A requires service.

5) To check the operation of the electronic disc, type: CAT ALL ":XMEM" [RTN]. (If ERROR: XMEM not INITIALIZEd appears, type INITIALIZE ":XMEM", then reexecute CAT ALL ":XMEM".) Press the [↓] key until you have seen the catalog information for every file on the electronic disc. Then type MAX XMEM [RTN] The result displayed should be either 32768 (HP 82718A Option 032) or 65536 (HP 82718A Option 064).

6) If the HP 82718A fails to perform a particular operation, such as copying a file from main memory to the electronic disc, or repeatedly displays an error message, such as invalid filespec, then carefully reread the information in this manual regarding that operation; you may be specifying the operation improperly.

7) If the display remains blank when [ATTN] is pressed, remove the HP-75 from the HP 82718A, and verify the operation of just the HP-75 (see the "Verifying Proper Operation" procedure on page 278 of the *HP-75 Owner's Manual*). If the HP-75 works properly by itself, but not when installed in the HP 82718A, the HP 82718A needs service.

If you cannot determine the cause of difficulty, write or telephone Hewlett-Packard at an address or phone number listed below under service.

LIMITED ONE-YEAR WARRANTY

What We Will Do

The HP 82718A Expansion Pod is warranted by Hewlett-Packard against defects in materials and workmanship affecting electronic and mechanical performance, but not software content, for one year from the date of original purchase. If you sell your unit or give it as a gift, the warranty is transferred to the new owner and remains in effect for the original one-year period. During the warranty period, we will repair or, at our option, replace at no charge a product that proves to be defective, provided you return the product, shipping prepaid, to a Hewlett-Packard service center.

What Is Not Covered

This warranty does not apply if the product has been damaged by accident or misuse or as the result of service or modification by other than an authorized Hewlett-Packard service center.

No other express warranty is given. The repair or replacement of a product is your exclusive remedy. **ANY OTHER IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS IS LIMITED TO THE ONE-YEAR DURATION OF THIS WRITTEN WARRANTY.** Some states, provinces, or countries do not allow limitation on how long an implied warranty lasts, so the above limitation may not apply to you. **IN NO EVENT SHALL HEWLETT-PACKARD COMPANY BE LIABLE FOR CONSEQUENTIAL DAMAGES.** Some states, provinces, or countries do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

This warranty gives you specific legal rights, and you may also have other rights which vary from state to state, province to province, or country to country.

Warranty for Consumer Transactions in the United Kingdom

This warranty shall not apply to consumer transactions and shall not affect the statutory rights of a consumer. In relation to such transactions, the rights and obligations of Seller and Buyer shall be determined by statute.

Obligation to Make Changes

Products are sold on the basis of specifications applicable at the time of manufacture. Hewlett-Packard shall have no obligation to modify or update products once sold.

Warranty Information

If you have any questions concerning this warranty, please contact an authorized Hewlett-Packard dealer or a Hewlett-Packard sales and service office. Should you be unable to contact them, please contact Hewlett-Packard at one of the three locations listed below.

- In the United States:

Hewlett-Packard Company
Personal Computer Group
Customer Communications
11000 Wolfe Road
Cupertino, CA 95014

Toll-Free Number: (800) FOR-HPPC (800 367-4772)

- In Europe:

Hewlett-Packard S.A.
150, route du Nant-d'Avril
P.O. Box CH-1217 Meyrin 2
Geneva
Switzerland
Telephone: (022) 83 81 11

Note: Do *not* send units to this address for repair

- In other countries:

Hewlett-Packard Intercontinental
3495 Deer Creek Rd.
Palo Alto, CA 94304
U.S.A.
Telephone: (415) 857-1501

Note: Do *not* send units to this address for repair

SERVICE

Hewlett-Packard maintains service centers in most major countries throughout the world. You may have your unit repaired at a Hewlett-Packard service center any time it needs service, whether the unit is under warranty or not. There is a charge for repairs after the one-year warranty period.

Hewlett-Packard products are normally repaired and reshipped within five (5) working days of receipt at any service center. This is an average time and could vary depending on the time of year and the work load at the service center. The total time you are without the unit will depend largely on the shipping time.

Obtaining Repair Service in the United States

The Hewlett-Packard United States Service Center for battery-powered computational products is located in Corvallis, Oregon:

Hewlett-Packard Company
Service Department

P.O. Box 999
Corvallis, Oregon 97339, U.S.A
or
1030 N.E. Circle Blvd.
Corvallis, Oregon 97330, U.S.A

Telephone: (503) 757-2000

Obtaining Repair Service in Europe

Service centers are maintained at the following locations. For countries not listed, contact the dealer where you purchased your unit.

AUSTRIA
HEWLETT-PACKARD Ges.m.b.H.
Kleinrechner-Service
Wagramerstrasse-Lieblgasse 1
A-1220 Wien (Vienna)
Telephone: (0222) 23 65 11

BELGIUM
HEWLETT-PACKARD BELGIUM SA/NV
Woluwedal 100
B-1200 Brussels
Telephone: (02) 762 32 00

DENMARK
HEWLETT-PACKARD A/S
Datavej 52
DK-3460 Birkerod (Copenhagen)
Telephone: (02) 81 66 40

EASTERN EUROPE
Refer to the address for Austria.

FINLAND
HEWLETT-PACKARD OY

FRANCE
HEWLETT-PACKARD FRANCE

Revontulentie 7
SF-02100 Espoo 10 (Helsinki)
Telephone: (90) 455 02 11

Division Informatique Personnelle
S.A.V. Calculateurs de Poche
F-91947 Les Ulis Cedex
Telephone: (6) 907 78 25

GERMANY
HEWLETT-PACKARD GmbH
Kleinrechner-Service
Vertriebszentrale
Berner Strasse 117
Postfach 560 140
D-6000 Frankfurt 56
Telephone: (611) 50041

ITALY
HEWLETT-PACKARD ITALIANA S.P.A.
Casella postale 3645 (Milano)
Via G. Di Vittorio, 9
1-20063 Cernusco Sul Naviglio (Milan)
Telephone: (2) 90 36 91

NETHERLANDS
HEWLETT-PACKARD NEDERLAND B.V.
Van Heuven Goedhartlaan 121
NL-1181 KK Amstelveen (Amsterdam)
P.O. Box 667
Telephone: (020) 472021

NORWAY
HEWLETT-PACKARD NORGE A/S
P.O. Box 34
Oesterndalen 18
N-1345 Oesteraas (Oslo)
Telephone: (2) 17 11 80

SPAIN
HEWLETT-PACKARD ESPANOLA S.A.
Calle Jerez 3
E-Madrid 16
Telephone: (1) 458 2600

SWEDEN
HEWLETT-PACKARD SVERIGE AB
Skalholtsgatan 9, Kista
Box 19
S-163 93 Spanga (Stockholm)
Telephone: (08) 750 2000

SWITZERLAND
HEWLETT-PACKARD (SCHWEIZ) AG
Kleinrechner-Service
Allmend 2
CH-8967 Widen
Telephone: (057) 31 21 11

UNITED KINGDOM
HEWLETT-PACKARD Ltd.
King Street Lane
GB-Winnersh, Wokingham
Berkshire RG11 5AR
Telephone: (0734) 784 774

International Service Information

Not all Hewlett-Packard service centers offer service for all models of HP products. However, if you bought your product from an authorized Hewlett-Packard dealer, you can be sure that service is available in the country where you bought it.

If you happen to be outside of the country where you bought your unit, you can contact the local Hewlett-Packard service center to see if service is available for it. If service is unavailable, please ship the unit to the address listed above under "Obtaining Repair Service in the United States." A list of service centers for other countries can be obtained by writing to that address.

All shipping, reimportation arrangements, and customs costs are your responsibility.

Service Repair Charge

There is a standard repair charge for out-of-warranty repairs. The repair charges include all labor and materials. In the United States, the full charge is subject to the customer's local sales tax. In European countries, the full charge is subject to Value Added Tax (VAT) and similar taxes wherever applicable. All such taxes will appear as separate items on invoiced amounts.

Computer products damaged by accident or misuse are not covered by the fixed repair charges. In these situations, repair charges will be individually determined based on time and materials.

Service Warranty

Any out-of-warranty repairs are warranted against defects in materials and workmanship for a period of 90 days from date of service.

Shipping Instructions

Should your unit require service, return it with the following items:

- A completed Service Card, including a description of the problem.
- A sales receipt or other proof of purchase date if the one-year warranty has not expired.

The product, the Service Card, a brief description of the problem, and (if required) the proof of purchase date should be packaged in adequate protective packaging to prevent in-transit damage. Such damage is not covered by the one-year limited warranty; Hewlett-Packard suggests that you insure the shipment to the service center. The packaged unit should be shipped to the nearest Hewlett-Packard designated collection point or service center. Contact your dealer for assistance. (If you are not in the country where you originally purchased the unit, refer to "International Service Information," above.)

Whether the unit is under warranty or not, it is your responsibility to pay shipping charges for delivery to the Hewlett-Packard service center.

After warranty repairs are completed, the service center returns the unit with postage prepaid. On out-of-warranty repairs in the United States and some other countries, the unit is returned C.O.D. (covering the shipping costs and the service charge).

Further Information

Service contracts are not available. Circuitry and designs are proprietary to Hewlett-Packard, and service manuals are not available to customers. Should other problems or questions arise regarding repairs, please call your nearest Hewlett-Packard service center.

POTENTIAL FOR RADIO/TELEVISION INTERFERENCE (U.S.A. ONLY)

The HP 82718A Expansion Pod generates and uses radio frequency energy and, if not installed and used properly—that is, in strict accordance with the instructions in this manual—may cause interference with radio and television reception. It has been tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC rules, which are

designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. In the unlikely event that your HP 82718A does cause interference to radio or television reception (which can be determined by removing all power to the HP 82718A and then reconnecting the power and turning it on), you are encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna.
- Relocate the HP 82718A with respect to the receiver.
- Move the HP 82718A away from the receiver.
- Plug the ac adapter/recharger into a different ac outlet so that the HP 82718A and the receiver are on different branch circuits.

If necessary, you should consult your dealer or an experienced radio/television technician for additional suggestions. you may find the following booklet, prepared by the Federal Communications Commission, helpful: *How to Identify and Resolve Radio-TV Interference Problems.* This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402, Stock Number 004-000-00345-4.

WHEN YOU NEED HELP

Hewlett-Packard is committed to providing after-sale support of its customers. To this end, our customer support department has established phone numbers that you can call if you have questions about this product.

Product Information. For information about Hewlett-Packard dealers, products, and prices, call the toll-free number below:

(800) FOR-HPPC
(800 367-4772)

Technical Assistance. For technical assistance with your product, call the number below:

(503) 754-6666

For either product information or technical assistance, you can also write to:

Hewlett-Packard
Personal Computer Group
Customer Communications
11000 Wolfe Road
Cupertino, CA 95014

ACCESSORIES INCLUDED WITH THE HP 82718A

APPENDIX B

Your HP 82718A Expansion Pod comes with each of the following:

- *HP 82718A Expansion Pod Reference Manual.*
- Two Clips for Carrying Strap.
- Rechargeable Nickel-Cadmium Battery Pack.
- Service Card.

For information about accessories for your HP-75, see your nearest Hewlett-Packard dealer. If you are outside the U.S., please contact the Hewlett-Packard Sales Office nearest you. Availability of all accessories, standard or optional, is subject to change without notice.

KEYWORD TAKEOVER AND DUPLICATE KEYWORDS

APPENDIX C

KEYWORD TAKEOVER BY THE HP 82718A

It is important that the final form of software being distributed is compatible with the intended hardware configuration. One issue related to this is keyword takeover. HP-75 BASIC programs are stored in a compact internal form. If the program includes keywords from plug-in ROMs, the ROM ID of the ROM is used as part of the internal representation of that keyword in the program. If the ROM has been removed when the program is run, the program will not be able to find the ROM and execute the keyword. The program will then halt and display ERROR: ROM missing at the line containing the missing command.

Seven of the keywords in the HP 82718A have the same names as keywords in the HP-75 operating system: CAT, CAT$,* CAT ALL, COPY, INITIALIZE, PURGE, and RENAME. The keywords in the HP 82718A will always override those in the HP-75, hence the term *keyword takeover.*

To see the effect of the HP 82718A taking over a keyword, look at an example. Type in a program containing the line PURGE "ABC" on an HP-75 installed in an HP 82718A. Copy the program onto a card, and then copy the card into a stand-alone HP-75. When you run the program in the stand-alone HP-75, you will get ERROR: ROM missing. The internal form of the program has the ROM ID of the HP 82718A PURGE command, not the PURGE command of the HP-75. Even though PURGE "ABC" does not require the electronic disc, because the program was created using an HP 82718A, its ROMs must be present for the program to run.

A similar situation occurs going the other direction. Type in a program containing the line PURGE "ABC:XMEM" on a stand-alone HP-75. (If you were to run this program on the HP-75, you would see ERROR: invalid filespec—a stand-alone HP-75 does not recognize the XMEM device code.) Copy the program onto a card, and then copy the card into an HP-75 installed in an HP 82718A. When you run the program, you will still get ERROR: invalid filespec. The internal form of the program does not have the ROM ID of the HP 82718A PURGE command, but instead has the HP-75 PURGE command. Even though PURGE "ABC:XMEM" is meant for the electronic disc, because the program was created without an HP 82718A, the program will not access the electronic disc.

You can avoid problems of keyword takeover simply by using the TRANSFORM command when moving BASIC programs from one HP-75 configuration to another. First, convert the program into a text file (TRANSFORM "*filename*" INTO TEXT). Then copy the text file from the source HP-75 to a mass storage device. Read it into the destination HP-75, and convert the text file into a BASIC program (TRANSFORM "*filename*" INTO BASIC).

ABBREVIATING KEYWORDS

The same idiosyncracies that give problems with keyword takeover will also create difficulties with abbreviated keywords. The HP-75 gives you the option of abbreviating keywords when you type them in, such as PU. for PURGE. The HP-75 recognizes abbreviated keywords as the HP-75 version of that

---

*CAT$ in the HP 82718A performs identically as CAT$ in the HP-75.

keyword only, regardless of any plug-in ROMs. If you type PU. "ABC:XMEM" on an HP-75 installed in an HP 82718A, you will see ERROR: invalid filespec. The PURGE command that is executed when you type PU. is the HP-75 PURGE, not the HP 82718A PURGE. Since the HP-75 doesn't recognize the XMEM device code, it rejects the filename.

You can use abbreviations if you really intend to specify the HP-75 version of the command. On the other hand, if you use abbreviations when you actually want the behavior of the ROM version of the command, you will not get that behavior. You will then either have to FETCH each line that was entered using an abbreviated command, and press [RTN] to reenter it without abbreviations, or TRANSFORM the program into text and then back into BASIC.

SYSTEM CONFIGURATION

The point of all this discussion is to emphasize the following: Make sure that the final, distributed version of your application software is consistent with the hardware configuration in which it will be used. You may be able to develop your software using this final hardware configuration, in which case there should be no problem.

Software developers tend to have other tools (i.e., LEX files, ROMs, etc.) resident in their machines to simplify their development work. Before you distribute your software, TRANSFORM it into text, and reconfigure your system to match the users' configuration—attach or remove an HP 82718A, remove unneeded LEX files or ROMs, change key assignments, etc. Then test the program thoroughly to make sure it did not assume the presence of any of the missing hardware or software.

DUPLICATE KEYWORDS

When two plug-in ROMs have identical keywords, the ROM ID that will be used in a BASIC program depends on the order of the kwywords within each ROM, and the order of the ROMs in the three ROM ports in the HP-75. The bar code software in the HP 82718A has six keywords with the same names as the corresponding keywords in the HP 82725A Bar Code Reader Module: CDIGIT OFF, CDIGIT ON, CDV11, CDV39, CODE11$, and CODE39$. Consequently, if you develop a program using both the HP 82718A and the Bar Code Reader Module, your program may not run properly when one or the other is not plugged in to the HP-75. Either create your programs with the final hardware configuration you expect to use (HP 82718A attached or not, Bar Code Reader Module in or not), or TRANSFORM your program into text, configure your system in its final form, and TRANSFORM your program back into BASIC, as described previously.

HP 82718A ERRORS AND RESPONSES

APPENDIX D

ALPHABETICAL ERROR LISTING

This appendix lists all the BASIC errors reported by the high-level modem, electronic disc, and bar code commands, and all the responses reported by the low-level modem commands.

Message	Error Number
MEMIC didn't select.	——
MM generic error message . . .	211
modem abort.	151
modem bad.	152
modem cold start	153
modem off.	150
nested too deep.	213
XMEM INITIALIZEd	214
XMEM not INITIALIZEd	212

NUMERICAL ERROR LISTING AND DESCRIPTIONS

Error Number	Message and Condition
150	modem off CARRIER?, FRAME?, HOOK?, MODIN$, MODOUT, PARITY? MODEM ON must be performed before this command can be executed.
151	modem abort CARRIER?, FRAME?, HOOK?, INITIALIZE ":MODM", MODEM OFF, MODEM ON, MODIN$, MODOUT, PARITY? The modem operation was aborted because the [ATTN] key was pressed or because of low batteries in the HP-75.
152	modem bad MODEM ON The modem cannot be turned on, and requires service.
153	modem cold start

CARRIER?, FRAME?, HOOK?, MODIN$, MODOUT, PARITY?

The modem has reset itself to its default state, probably because of insufficient battery power in the HP 82718A.

---

211 MM generic error message

COPY, PRINT #

There may be a RAM hardware failure. The electronic disc should be initialized (INITIALIZE ":XMEM").

---

212 XMEM not INITIALIZEd

ASSIGN #, CAT, COPY, CREATE DIR, DIR IS, MAX XMEM, RESTORE #, XCAT$, XDIR$, XMEM

INITIALIZE ":XMEM" must be executed before this operation can be performed.

---

213 nested too deep

COPY, CREATE DIR, RENAME

The maximum number of directory levels is 7 (0-6).

---

214 XMEM INITIALIZEd

The electronic disc has reset itself to its default state, and all information in it has been lost.

— MEMIC didn't select

INITIALIZE ":XMEM" and when the HP-75 is turned on

If this message appears, there is a hardware problem with the HP 82718A, and it requires service.

ALPHABETICAL RESPONSE LISTING

Long Response	Short Response
ABORT	A
BAD ANS	B
BUSY	0
CONNECT	1
CONN LOST	8
FAIL	9
NO ANS	2
NO DIAL	3
OK	4
OV	OV
RING	5
RING IN	7

NUMERICAL RESPONSE LISTING AND DESCRIPTIONS

Short Response	Long Response and Condition
0	BUSY (BUSY followed by a space) D (DIAL), M (MODEM), O (ORIGINATE), R (REDIAL) The modem has detected a busy signal.
1	CONNECT A (ANSWER), D (DIAL), M (MODEM), O (ORIGINATE), R (REDIAL) • D, R: In MODEM mode, the modem has detected an answer tone; in VOICE mode, the modem has detected anything other than a busy signal. • M: A valid data connection exists. • O: The modem has detected an answer tone.
2	NO ANS A (ANSWER), D (DIAL), M (MODEM), O (ORIGINATE), R (REDIAL) • A: The modem did not detect an originate tone. • D, R: In MODEM mode only, the modem detected either no dial tone or an unsteady tone after dialing, or there was no answer after either

the maximum number of rings (ring count + 4, up to a limit of nine) or 50 s elapse after the dialing operation ends.

- M, O: The modem detected either no dial tone or an unsteady tone after dialing, or there was no answer after either the maximum number of rings (ring count + 4, up to a limit of nine) or 50 s elapsed after the dialing operation ended.

---

3 NO DIAL

D (DIAL), M (MODEM), R (REDIAL), X (XMIT)

The modem did not detect a dial tone.

---

4 OK

(all commands and spontaneous responses)

The modem executed or aborted a command string, or completed the action for a spontaneous response.

---

5 RING

D (DIAL), O (ORIGINATE), R (REDIAL)

- D, R: In MODEM mode only, the modem detected a ring back from a remote modem.
- O: The modem detected a ring back from a remote modem.

---

7 RING IN

Spontaneous response

The modem detected an incoming ring.

---

8 CONN LOST

Spontaneous response and T (TEST)

The carrier from the remote modem has disappeared.

- T 2: There is no data communication established.

---

9 FAIL

Power up and T (TEST)

The modem has failed its self-test.

A	ABORT Spontaneous response The modem has aborted a command string without executing it because of an error or because it contained a BS (backspace—ASCII 8 or control-H) or CAN (cancel—ASCII 24 or control-X).
B	BAD ANS A (ANSWER), D (DIAL), M (MODEM), O (ORIGINATE), R (REDIAL) • A: In MODEM mode only, the modem detected a tone not in the originate band. • D, R: In MODEM mode only, the modem detected either a tone not in the answer band or a dial tone after dialing. • M, O: The modem detected either a tone not in the answer band or a dial tone after dialing.
OV	OV Spontaneous response The modem receive buffer has overflowed.

EXACT SEQUENCE OF RESPONSE CHARACTERS

Below are listed the exact sequence of characters reported by the modem when the different responses occur. The following symbols are used: CM (command character), CR (carriage return), LF (line feed), and ␣ (space). The spaces between symbols are to improve readability—any spaces reported in the response are shown with ␣.

ABORT

After all commands except T 3 through T F:

Long: \ CR LF CM ABORT CR LF CM OK CR LF
Short: \ CR LF CM A CR LF CM 4

After T 3:

Long: CR LF CM ABORT CR LF CM OK CR LF
Short: CR LF CM A CR LF CM 4

After T 4 through T F:

Long: ?\ CR LF CM ABORT CR LF CM OK CR LF
Short: ?\ CR LF CM A CR LF CM 4

BAD ANS

Long: CR LF CM BAD ANS CR LF CM OK CR LF
Short: CR LF CM B CR LF CM 4

BUSY

Single dialing operation or the last dial of a repeated dialing operation (either redialing or multiple dialing commands in the command string):

Long: CR LF CM BUSY CR LF CM OK CR LF
Short: CR LF CM 0 CR LF CM 4

All but the last dial of a multiple dialing operation:

Long: CR LF CM BUSY
Short: CR LF CM 0

CONNECT

Long: CR LF CM CONNECT CR LF CM OK CR LF
Short: CR LF CM 1 CR LF CM 4

CONN LOST

Long: CR LF CM CONN LOST CR LF CM OK CR LF
Short: CR LF CM 8 CR LF CM 4

FAIL

Long: CR LF CM FAIL CR LF
Short: CR LF CM 9

NO ANS

Long: CR LF CM NO ANS CR LF CM OK CR LF
Short: CR LF CM 2 CR LF CM 4

NO DIAL

Long: CR LF CM NO DIAL CR LF CM OK CR LF
Short: CR LF CM 3 CR LF CM 4

OK

After commands are echoed:

Long: CR LF CM OK CR LF
Short: CR LF CM 4

When the modem is turned on (this response will be read with a single MODIN#):

Long: %OK
Short: %4

After the Z (ZAP) command (this response will be read with a single MODIN#):

Long: CM OK
Short: CM 4

OV

Long: CM OV
Short: CM OV

RING

Long: CR LF CM RING
Short: CR LF CM 5

RING IN

Long: CR LF CM RING IN CR LF CM OK CR LF
Short: CR LF CM 7 CR LF CM 4

Q (QUERY)

Long: CR LF CM *query response* CR LF CR LF CM OK CR LF
Short: CR LF CM *query response* CR LF CR LF CM 4

For details on the specific characters returned by the Q command, see the "Low-Level Modem Command Dictionary" or "Decoding the Query Response" in Section 2.

Command String Echoing

Command strings are echoed in exactly the same form as they were sent to the modem. The CR that ended the command string will always be echoed as a CR LF, even if command echoing is disabled (E 1). For commands that expect a parameter, the terminating CR will not be echoed if it is received before the parameter—instead, the command string will be aborted. Neither CAN nor BS are echoed when they cause a command string to be aborted.

Dial String Echoing

If command echoing is enabled (E 0), the modem will echo the dial string when it receives the command string containing it. The dial string will be echoed in exactly the same form as it appeared in the command string. The modem will echo the dial string again each time it is dialed by any of the three dialing commands: D (DIAL), R (REDIAL), or X (XMIT). The D and X commands echo the string once (unless they are repeated in the command string—see the D command in the "Low-Level Modem Command Dictionary"). The R command echoes the dial string up to 10 times as it redials.

The dial string echoed during dialing will be in a different form than in the command string. Characters that are ignored in the dial string are not echoed. The special characters P and W will not be echoed, but will break the dial string into segments separated by spaces. Echoing will stop for the duration of the waits imposed by those characters. The special character I will not be echoed, but will cause all digits between the I and the next P or W (or the end of the dial string) to be echoed. The digits # and * will be echoed as ; and <. There will be a trailing space but no carriage return or line feed after the entire dial string has been echoed.

KEYWORD AND COMMAND SUMMARY

APPENDIX E

HIGH-LEVEL MODEM KEYWORDS

Keyword	Page	Description
AVAIL?	62	Checks if data is available from the modem
CARRIER?	62	Checks if there is a carrier
FRAME?	68	Checks if framing error from last MODIN$
HOOK?	68	Checks if modem is on-hook
INITIALIZE ":MODM"	69	Resets the modem to its default state
MODEM OFF	70	Turns modem off
MODEM ON	71	Turns modem on
MODIN$	71	Reads data from the modem
MODOUT	72	Sends a string to the modem
PARITY?	72	Checks if parity error from last MODIN$
READY?	74	Checks if modem is ready for data

LOW-LEVEL MODEM KEYWORDS

Command	Page	Description
A (ANSWER)	79	Places the modem in ANSWER mode
B (BREAK)	79	Sends a break
C (COUNT)	80	Sets the ring count
D (DIAL)	80	Dials a sequence of numbers
E (ECHO)	82	Enables and disables command echoing
F (FORMAT)	82	Sets data format (parity) and data length
G (GIVEBACK)	83	Enables and disables data echoing
H (HANGUP)	83	Hangs up modem
I (INITIALIZE)	83	Resets the modem to its default state
K (KONTROL)	84	Sets serial handshake protocol
L (LONG)	86	Specifies long or short echoing of modem responses
M (MODEM)	86	Places the modem in MODEM mode
N (NEW)	87	Specifies new command character
O (ORIGINATE)	87	Places the modem in ORIGINATE mode
P (PICKUP)	88	Takes modem off-hook and places it in VOICE mode
Q (QUERY)	88	Asks the modem for status information
R (REDIAL)	89	Redials last dialed number
T (TEST)	89	Performs modem self-tests
U (UNLISTEN)	90	Allows sending command character to modem as if it were data
V (VOICE)	90	Places the modem in VOICE mode
X (XMIT)	91	Dials a number without call progress reporting after dialing
Z (ZAP)	91	Clears modem receive buffer

ELECTRONIC DISC KEYWORDS

Keyword	Page	Description
ASSIGN #	61	Associates file number with specified file on electronic disc and opens the file
CAT	62	Catalogs files and directories on electronic disc
COPY	67	Copies files within electronic disc and between the disc, the HP-75, and HP-IL mass storage devices
CREATE DIR	67	Creates a new directory on electronic disc
DIR IS	68	Changes the current directory
INITIALIZE ":XMEM"	70	Resets the electronic disc to its default state
MAX XMEM	70	Returns total amount of memory on electronic disc
PRINT #	72	Writes data to electronic disc file
PURGE	73	Deletes file from electronic disc
PURGE DIR	73	Deletes directory and child files from electronic disc
READ #	74	Reads data from electronic disc file
REMOVE	74	Asks user before purging files
RENAME	76	Changes name and/or location of file on electronic disc
RESTORE #	76	Sets specified file pointer to indicated line number
XCAT$	77	Returns catalog of files on electronic disc
XDIR$	77	Returns catalog of current directory on electronic disc
XMEM	77	Returns amount of available memory on electronic disc

BAR CODE KEYWORDS

Keyword	Page	Description
CDIGIT OFF	64	Turns off automatic check digit verification by the decoders
CDIGIT ON	64	Turns on automatic check digit verification by the decoders
CDV11	65	Verifies the check digit for Code 11 labels entered from the keyboard
CDV39	65	Verifies the check digit for Code 39 labels entered from the keyboard
CODE11$	65	Scans a single Code 11 label
CODE39$	66	Scans a single Code 39 label

HEWLETT PACKARD

Portable Computer Division
1000 N.E. Circle Blvd., Corvallis, OR 97330, U.S.A.

European Headquarters
150, Route du Nant-D'Avril
P.O. Box, CH-1217 Meyrin 2
Geneva-Switzerland

HP-United Kingdom
(Pinewood)
GB-Nine Mile Ride, Wokinham
Berkshire RG11 3LL

82718-90001 English

Printed in U.S.A. 7/84