

Includes Magnetic Card

HEWLETT-PACKARD

HP-75

USERS' LIBRARY SOLUTIONS
I/O Utilities



NOTICE

The program material contained herein is supplied without representation or warranty of any kind. Hewlett-Packard Company therefore assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

TABLE OF CONTENTS

	INTRODUCTION	1
I.	BRIEF REVIEW OF HP-IL	3
II.	NOTE ON SYNTAX USED IN THIS SOLUTIONS BOOK	7
III.	COPYING THE UTILITY INTO YOUR HP-75	7
IV.	SENDING INFORMATION TO HP-IL DEVICES	9
	The SENDIO Instruction	9
	Assigning Device Addresses	11
	Resuming Data Transmission	11
	SENDIO Restrictions	13
V.	RECEIVING INFORMATION FROM THE LOOP	15
	The ENTIO\$ Instruction	15
	Defining Logical End-of-Record	17
	Enhanced Printing Control	19
	ENTIO\$ Restrictions	19
VI.	SAMPLE PROGRAMS	21
	TALK: HP-75/HP-85 Interface	23
	LOAD: Load HP-75 Data to HP-85 Tape	27
	MODEM: HP-75/Modem Interface	31
	VMETER: Obtain Readings From Multimeter	35
	APPENDIX A: ERRORS AND WARNINGS	39
	APPENDIX B: HP-IL COMMAND SUMMARY	43
	APPENDIX C: EXTENDED HP-IL COMMAND CAPABILITY	47

INTRODUCTION

The I/O Utility gives the HP-75 the capability to communicate with any Hewlett-Packard Interface Loop (HP-IL) talker or listener device. With the utility, the HP-75 can be programmed to send low-level HP-IL commands to any device and receive data from any device. This solutions book is for programmers who are experienced with the HP-75 and with HP-IL. Familiarity with HP-75 and HP-IL commands is assumed. Information on specific HP-IL commands can be found in the owner's manuals for HP-IL devices, and also in THE HP-IL SYSTEM: An Introductory Guide to the Hewlett-Packard Interface Loop, by Gerry Kane, Steve Harper, and Dave Ushima, published by OSBORNE/McGraw-Hill, Berkeley, California, 1982.

The solutions book contains instructions for copying the utility into HP-75 memory as well as instructions for using the utility. One chapter is devoted to transmitting information, and one is devoted to receiving information. Sample programs are included to illustrate how the utility can be used to perform specific functions. Quick reference information on HP-IL is included in appendices.

NOTES

I. BRIEF REVIEW OF HP-IL

Before going further in this solutions book, you may find it helpful to review the fundamentals of HP-IL. This review covers the material necessary to understand the rest of this solutions book. Previous exposure to HP-IL is assumed. Users who feel sufficiently comfortable with HP-IL may skip this chapter.

HP-IL is a system in which devices are configured in a circular loop structure. Devices communicate with each other by sending messages around the loop. When a device sends or "sources" a message, it is examined by each device on the loop and passed on to the next device. Messages travel around the loop in one direction only, until they return to the original sender.

HP-IL operates on a master-slave principle. One of the devices on the loop functions as loop controller. This device has responsibility of transmitting all commands to other devices on the loop. The HP-75 is a device which can function as loop controller. A device which can send data, but not commands, to other devices on the loop is called a talker device. Although a device has talker capability, it will not actually send its data until commanded to do so by the controller. Listeners are devices with the capability to receive data from the loop. Once again, listeners will not receive data until commanded to do so by the controller.

Controller, talker and listener are the three basic capabilities of HP-IL devices. A device may have one or more of these capabilities. There can be any number of devices on the loop with controller, talker, or listener capabilities. Only one controller may be active at a time, and only one talker may be active at a time, but there may be more than one active listener. The controller device that was active when the system was turned on is called the system controller, and is in charge of the whole system.

The system controller assigns an address to each device on the loop. It can direct commands to specific devices by using the device address. The address is a number from 1 to 30 or, with extended addressing, from 1 to 960.

Messages are sent around the loop as 11-bit "frames". The first 3 bits of the frame identify the type of frame. For purposes of this discussion, we will distinguish between two types of frames only: command frame and data frame. The last 8 bits are the actual content of the frame. Thus, to send a command such as "IFC", a frame would be sent out as follows: 3 bits identifying the frame as a command frame; 8 bits with the command code for "IFC" (binary 10010000).

Each frame is examined by every device on the loop. By examining the frame, devices determine whether or not any further action is required. Action is indicated in a number of circumstances. Certain commands, such as IFC, indicate action for all devices on the loop. Other commands, such as LAD and TAD, contain a device address. A device acts on the command only if

NOTES

the address in the command is the same as the address of the device. Some commands are processed only if the device is in an active state. Data frames and DDL commands are processed only by devices in an active listener state. SDA is processed only by devices that are active talkers.

An example of how all this works is as follows: Suppose the HP-75 controller wants to print a line on a printer. Assume that the printer has a device address of 2, and assume all devices on the loop are in inactive status. The controller would first send a LAD2 (listen-address device 2) around the loop. This would put device 2, the printer, in active listener status. The controller would then source the data frames. If the line to be printed were an 80-character line, there would be 80 data frames, plus a frame each for a carriage return and a line feed character. Once all data has been sent, the controller sources an UNL (Unlisten) command frame. This would deactivate all listener devices on the loop, in this case, the printer.

Appendix B contains a list of the HP-IL commands that can be used with this utility.

NOTES

II. NOTE ON SYNTAX USED IN THIS SOLUTIONS BOOK

Certain syntax conventions will be in use throughout this solutions book to describe HP-75 commands and instructions. The rules are:

- 1) Words appearing in upper case should be interpreted literally. It is not necessary to use upper case when entering the word. For example, if the solutions book instructs you to type "SENDIO", either "SENDIO" or "sendio" is acceptable.
- 2) Lower case words appearing in brackets are descriptive. Actual data should be substituted when entering the instruction into the HP-75. The item <command list> in the solutions book refers to a list of commands that is to be supplied by the programmer.
- 3) When quotation marks are required as part of the instruction, the solutions book will use single quotation marks (''). Double quotation marks (") in the solutions book should not be reproduced when entering the commands into the HP-75.

III. COPYING THE UTILITY INTO YOUR HP-75

The utility is provided on a magnetic card. In order to use the utility, the card must be copied into HP-75 memory. You may choose any valid HP-75 filename in which to store the utility. The file will be a Lexfile, type "L". To copy the utility into HP-75 memory, turn on the HP-75 and type:

COPY CARD TO <filename>

See the HP-75 Owner's Manual for details on the COPY command. When you have completed the copy command (2 tracks must be read), the utility is copied into memory and ready for use!

NOTES

IV. SENDING INFORMATION TO HP-IL DEVICES

THE SENDIO INSTRUCTION

The SENDIO instruction is used to send commands and data to HP-IL devices. SENDIO can be issued from the HP-75 keyboard or executed in a BASIC program. The syntax of the instruction is:

```
SENDIO <device code>,<command list>,<data list>
```

The three parameters are character strings. Any valid HP-75 string expression may be used. The meaning of each of the parameters is:

<device code> : A name given to a device via an HP-75 ASSIGN IO command.

<command list>: A list of HP-IL commands to be executed, separated by commas. The commands may be specified in the form of command mnemonics. A summary of mnemonics is supplied in Appendix B.

<data list> : A character string to be transmitted as data.

Most of the time, SENDIO will be used to activate a device as a listener. The device to be activated can be specified with either the <device code> parameter or the <command list>.

Use the <device code> when you know what name has been given to the device with ASSIGN IO. To do this, specify the name of the device in the <device code> parameter. When this is done, you can generate a LAD to the named device either by leaving the <command list> null, or by specifying "LAD#" in the <command list>. "LAD#" can be used in combination with any other HP-IL commands, and it may appear anywhere in the <command list>.

Use the <command list> when the device's HP-IL address is known. To do this, specify "LADn", where "n" is the HP-IL address of the device. This will cause a LADn to be sent to device "n" regardless of what appears in the <device code>. You may have any number of "LADn" commands within a single SENDIO instruction, and you may have both "LADn" and "LAD#" in the same SENDIO. Here is an example of SENDIO that sends the word "HELLO" to a device named ":D1" and also to the devices with addresses 3 and 4:

```
SENDIO ':D1','LAD#,LAD3,LAD4','HELLO'
```

It is not necessary to supply values for all three parameters. If you wish to omit a parameter, you must specify a null string. Here is an example of a SENDIO which sends no commands, but which sends the word "DATA" to any devices on the loop that are already in active listener status:

```
SENDIO '',','DATA'
```

NOTES

The SENDIO instruction processes parameters from left to right. Processing proceeds as follows:

- 1) If the <device code> parameter has been specified, SENDIO determines the device's HP-IL address. This device address is used when processing the <command list>. If the <device code> field is null, then no action is taken in this step.
- 2) The <command list> is processed. Commands are sent one at a time through the loop. RFC (Ready For Command) frames are sent automatically after each command.
- 3) After all commands are sent, the data specified in the <data list> is sent around the loop, one character at a time. If a listener device sends an NRD frame (Not Ready for Data), a warning is issued and the transmission of data is terminated. See the subheading "Resuming Data Transmission" for details on recovering from this condition.
- 4) After all commands and data have been sent, Untalk (UNT) and Unlisten (UNL) commands are sent around the loop, deactivating all talker and listener devices. If you want talker and listener devices to remain active, you can suppress the automatic UNT/UNL by including a "TL+" anywhere in the <command list>.

ASSIGNING DEVICE ADDRESSES

The HP-75 assigns addresses to devices on the loop with the ASSIGN IO statement. To assign addresses without using the ASSIGN IO statement, the SENDIO command should be used. The statement:

```
SENDIO '', 'AAU, AAD1', ''
```

will first clear the loop address structure and then the AAD1 will cause the loop to automatically readdress the loop starting with address 1. No other commands should appear in the <command list>.

RESUMING DATA TRANSMISSION

If a device on the loop sends an NRD frame while SENDIO is transmitting data, transmission terminates. To enable you to resume transmission from the point of interruption, the "SEND?" function has been provided.

SEND? is a function that takes no parameters. It returns an integer value. The value returned is the position in the <data list> of the character that was unsuccessfully sourced in the last SENDIO instruction. If the <data list> in the last SENDIO was null, or if the last SENDIO completed successfully, then SEND? returns a 0.

Here is an example of how to use SEND?. The following instructions will send the characters "I love my HP-75" to the fourth device on the loop:

NOTES

```
A$ = 'I love my HP-75'
SENDIO '', 'LAD4', A$
```

Suppose an NRD frame is received after SENDIO attempts to send the "m" from "my". SENDIO will stop transmitting at this point. The function SEND? would return a value of 8, since the "m" was the eighth character in the data list. Transmission could be resumed with the following instructions:

```
B$ = A$[SEND?]
SENDIO '', 'LAD4', B$
```

B\$ in this example will be a substring of A\$ starting at the eighth position. Its value will be "my HP-75".

SENDIO RESTRICTIONS

SENDIO causes the HP-75 to become active as a talker. Therefore, although it is possible to issue TAD commands with SENDIO, doing so will cause more than one talker to become active on the loop. This will result in a dead-lock condition. Therefore, it is recommended not to use SENDIO to talk-address devices.

If DISPLAY IS or PRINTER IS devices have been assigned on the HP-75, the talkers will automatically be deactivated even if "TL+" is specified in the <command list>. Although "TL+" will stop SENDIO from automatically deactivating listeners, HP-75 I/O operations not related to SENDIO may cause deactivation when DISPLAY IS or PRINTER IS devices are in use.

NOTES

V. RECEIVING INFORMATION FROM THE LOOP

THE ENTIO\$ INSTRUCTION

The ENTIO\$ function is used to receive data from other HP-IL devices. In contrast to SENDIO, which is an instruction, ENTIO\$ is a function, and it returns a character string value. The string returned is the data transmitted by an HP-IL device. The syntax of ENTIO\$ is:

ENTIO\$(<device code>,<command list>)

The parameters are character strings. Any valid HP-75 string expression may be used. You may specify the null string for either of the two parameters, but not both. The meaning of the parameters are:

<device code> : The name given to a device via an HP-75 ASSIGN IO command.

<command list>: One or more HP-IL commands, separated by commas. The commands may be specified in the form of command mnemonics. A summary of mnemonics is supplied in Appendix B.

Most of the time, ENTIO\$ will be used to activate a device as a talker. The device to be activated can be specified with either the <device code> parameter or the <command list>.

Use the <device code> when you know what name has been given to the device with ASSIGN IO. To do this, specify the name of the device in the <device code> parameter. You can generate a "TAD#,SDA" to the named device by leaving the <command list> null, or you can talk or listen address the named device by specifying "TAD#" or "LAD#" in the <command list>. "TAD#" and "LAD#" can be used in combination with any other HP-IL commands, and may appear anywhere in the <command list>.

Use the <command list> when the device's HP-IL address is known. To do this, specify "TADn" or "LADn", where "n" is the HP-IL address of the device. This will send a TADn or a LADn to device "n" regardless of what appears in the <device code>. Both "TADn" and "LADn" may be used in conjunction with other HP-IL commands within a single ENTIO\$ instruction. You may also combine "TADn" or "TAD#" with "LADn" and "LAD#" in the same ENTIO\$.

With ENTIO\$, either the <device code> or the <command list> may be null, but not both. If null strings are specified for both parameters, an error results (see Appendix A).

Similar to SENDIO, ENTIO\$ processes parameters from left to right. Note, however, that ENTIO\$ does not have a data field. This is because ENTIO\$ causes the HP-75 to become active as a controller and a listener only; it can transmit commands and receive data, but it cannot send data. Processing proceeds as follows:

NOTES

- 1) If the <device code> parameter has been specified, ENTIO\$ determines the device's address on the loop. This device address is used when processing the <command list>. If the <device code> field is null, then no action is taken in this step.
- 2) The <command list> is processed. Commands are sent one at a time through the loop. RFC (Ready For Command) frames are automatically sent after each command.
- 3) Data is collected from the loop. The value returned by the function ENTIO\$ will be the data collected in this step. Data collection terminates when one of the following three conditions is met:
 - a) An EOT frame is received.
 - b) The number of data frames received exceeds 256 bytes. If this occurs, the HP-75 sources an NRD frame.
 - c) A logical end-of-record character is received. If this occurs, the HP-75 sources an NRD frame. See the subheading "Defining Logical End-of-Record" for more details.

End-of-line sequences, such as carriage return or line feed characters are treated as data by ENTIO\$. If these characters are received, they are included in the string returned by the ENTIO\$ function.

- 4) Untalk and Unlisten commands are sent around the loop to deactivate all talker and listener devices. If you wish talkers and listeners to remain active, you can suppress the automatic Untalk/Unlisten by coding "TL+" in the <command list>.

DEFINING LOGICAL END-OF-RECORD

You can define a character to serve as a logical end-of-record during transmission. When the logical end-of-record is received, transmission will be terminated, and the data that has been collected up to this point will be returned by ENTIO\$. To specify character "XX" as the logical end-of-record character, code "TR:XX" anywhere in the <command list>, where "XX" is the ASCII representation of the character in hexadecimal.

Here is an example of how to use logical end-of-record: Suppose that the data you are receiving consists of lines of text with a line feed character separating each line. Rather than having ENTIO\$ return 256-character strings with embedded line feed characters, you may wish to treat each text line as a logical record. To accomplish this, you would simply code "TR:0A" within the command list. This will cause the line feed character to be treated as the logical end-of-record. Each time ENTIO\$ is invoked, it will return a string containing exactly one line of text. The line feed character will be included in the string.

NOTES

ENHANCED PRINTING CONTROL

You can have a carriage return and a line feed (ASCII 0COA) inserted into the data string automatically each time an END frame is received from the talker. This is accomplished by coding "CL+" anywhere in the <command list>. An example of this would be when receiving readings from a voltmeter and printing them on a printer. The voltmeter transmits readings (data frames) followed by END frames. If these transmissions were printed as is, the readings would all be on one line with no spacing. Specifying "CL+" will cause carriage return and line feed to be inserted after each reading, thus allowing each reading to be printed on a separate line.

ENTIO\$ RESTRICTIONS

The ENTIO\$ function will return the null string unless either SDA, SST, SDI, or SAI appears as the last command in the <command list>. These commands should not appear in the <command list> except as the last command. SDA, SST, SDI, or SAI occurring as other than the last command will cause the transmission to begin, but the transmission will be terminated after one frame is sent.

If DISPLAY IS or PRINTER IS devices have been assigned on the HP-75, the talker will automatically be deactivated even if "TL+" is specified in the <command list>. Although "TL+" will stop ENTIO\$ from automatically deactivating listeners, HP-75 I/O operations not related to ENTIO\$ may cause deactivation when DISPLAY IS or PRINTER IS devices are in use.

NOTES

VI. SAMPLE PROGRAMS

The programs in this section are provided to illustrate the power and flexibility of SENDIO and ENTIO\$. Although you may find them useful in practical applications, the programs exist primarily for demonstration purposes. If you want to use these programs, you will probably have to change them in order to adapt them to your particular system configuration. The owner's manuals for the specific devices used in these programs contain background material necessary for understanding these examples.

NOTES

TALK: HP-75/HP-85 Interface

An interface between the HP-75 and the HP-85, this program can send mail or messages between the two computers using the interface link. The HP-85 must have the HP-IL ROM, and must be set in non-controller mode. There are two programs involved: one for the HP-75 and one for the HP-85. The programs assume that the HP-85 is assigned a device name of ":8X".

INSTRUCTIONS

- 1) Key in each program to the appropriate machine.
- 2) Run the programs concurrently.
- 3) The HP-75 starts out as the talker, the HP-85 as the listener.
- 4) The prompt "MESSAGE:" will appear on the display of the talker.
- 5) Key in the message to be sent and press the return key. The message will appear on the display of the listener.
- 6) To swap talker/listener, precede message with a "*".
- 7) To stop the programs, precede the message with a "/".
- 8) Unless the last message began with a "/", go to step 4.

NOTES

PROGRAM LISTING

HP-75 program

```

10 DIM A$(256)
20 INPUT 'Message : ';A$
30 SENDIO '8X', 'LAD#', A$&CHR$(10)
40 IF A$(1,1)='*' THEN 70
50 IF A$(1,1)='\ ' THEN END
60 GOTO 20
70 A$=ENTIO$( '8X', 'TAD#', SDA')
80 IF A$='' THEN 70
90 DISP USING 130 ;A$
100 IF A$(1,1)='*' THEN 20
110 IF A$(1,1)='\ ' THEN END
120 GOTO 70
130 IMAGE 'HP-85-->HP-75 : ',K

```

```

! dimension the string
! input message
! send
! change talkers?
! terminate communication?

! enter message

! display message
! change talkers?
! terminate communication?

```

HP-85 program

```

10 DIM A$(256)
20 ENTER 9;A$
30 DISP USING 130 ;A$
40 IF A$(1,1)='*' THEN 70
50 IF A$(1,1)='\ ' THEN END
60 GOTO 20
70 DISP "Message : ";
80 INPUT A$
90 OUTPUT 9;A$
100 IF A$(1,1)='*' THEN 20
110 IF A$(1,1)='\ ' THEN END
120 GOTO 70
130 IMAGE "HP-75-->HP-85 : ",K
140 END

```

```

! dimension the string
! enter message
! display message
! change talkers?
! end communication?

! input message
! send
! change talkers?
! end communication?

```

NOTES

LOAD: Load HP-75 Data to HP-85 Tape

This program loads data from the HP-75 to tape on the HP-85. The HP-85 must have the HP-IL ROM and be in non-controller mode. There are two programs: one for the HP-75 and one for the HP-85. The programs assume that the HP-85 has a device name of ":8X".

INSTRUCTIONS

- 1) Key in the programs to the appropriate machines.
- 2) Run the programs concurrently.
- 3) The HP-75 will issue the prompt: "DATAFILE :".
- 4) Make sure the HP-85 has created room for the data file on tape. The tape light should be off.
- 5) Enter the file name on the HP-75, and press the return key.
- 6) The named file will be sent to the HP-85.

NOTES

PROGRAM LISTING

HP-75 program

```

10 DIM A#[512],B#[256]
20 ON ERROR GOTO 90
30 INPUT 'DATAFILE : ';C$
40 ASSIGN #1 TO C$,TEXT
50 A$=''
60 READ #1;B$
70 A$=A$&B$
80 GOTO 60
90 SENDIO ';8X','LAD#',A$&CHR$(10)
100 ASSIGN #1 TO *
```

```

! dimension the strings
! on error at the end of file
! input file name
! assign file to be copied

! read the data

! send the data
```

HP-85 program

```

10 DIM A#[512]
20 CREATE "FILE",2
30 ASSIGN #1 TO "FILE"
40 ENTER 9;A$
50 PRINT #1;A$
60 ASSIGN #1 TO *
70 END
```

```

! dimension the string
! create the data file

! get the file
! store on tape
! close the data file
```

NOTES

MODEM: HP-75/Modem Interface

This program allows communication between the HP-75 and another mainframe, through an HP-IL modem. While the program is running, the HP-75 functions as a terminal.

INSTRUCTIONS

- 1) Turn on the modem.
- 2) On the telephone, dial the computer.
- 3) Place the phone into the modem.
- 4) When the carrier light goes on, run the MODEM program.
- 5) The HP-75 now functions as a terminal. From this point on, the action you take depends on the computer to which you are connected. Do what you would normally do to communicate with the computer from a terminal.

NOTES

PROGRAM LISTING

10 WIDTH 256	! set large width
20 CLEAR 'mo'	! clear the modems buffers
30 SENDIO 'mo','unl,ren,lad#','D;	! remote enable the modem
@ SENDIO 'mo','nre',''	
40 K\$=KEY\$ @ IF K\$#''	! get the key
THEN GOSUB 80	
50 E\$=ENTIO\$('mo','unl,TAD#,sda')	! get input from modem
60 DISP E\$;	! display input
70 GOTO 40	
80 SENDIO 'mo','unl,lad#',K\$! send the key
90 RETURN	

NOTES

VMETER: Obtain Readings From Multimeter

This program allows the HP-75 to interface with the HP 3468A Multimeter. The program illustrates the simplicity of programmatically obtaining and storing information from the multimeter. The program assumes that the multimeter has a device name of ":VM".

INSTRUCTIONS

- 1) The program will produce the following prompts. The prompts are explained in the HP 3468A Owner's Manual.

```

FUNCTION CODE :
RANGE CODE :
AUTOZERO: 0-OFF, 1-ON :
TIME INTERVAL :          (time between readings)
NUMBER OF READINGS :
STORE AS: 0-STRING, 1-NUMERIC :
FILENAME :                (name of file to store readings)
  
```

The file specified will be created automatically by the program.

- 2) The program will run. Termination occurs automatically when the specified number of readings have been taken.

NOTES

PROGRAM LISTING

```

10 SENDIO ':vm', 'unl,ren,lad#','T2'
20 INPUT 'FUNCTION CODE: ' ; F$
30 INPUT 'RANGE CODE: ' ; R$
40 INPUT 'AUTOZERO: 0-OFF, 1-ON : ' ; Z$
50 INPUT 'TIME INTERVAL : ' ; T
60 INPUT 'NUMBER OF READINGS : ' ; N
70 INPUT 'STORE AS: 0-STRING,
    1-NUMERIC' ; A
80 INPUT 'FILENAME : ' ; C$
90 ASSIGN #1 to C$
100 SENDIO ':vm', 'unl,lad#',
    'F'&F$&'R'&R$&'Z'&Z$
110 FOR I=1 TO N
120 A$=ENTIO$(':vm', 'unl,TAD#,sda')
130 IF A=1 THEN X=VAL(A$)
    @ PRINT #1,I;X @ GOTO 150
140 PRINT #1,I;A$
150 SENDIO ':vm', 'unl,lad#','T2'
160 WAIT T
170 NEXT I
180 SENDIO ':vm', 'unl,lad#,sdc',''

```

```

! set single trigger mode
! function code
! range code
! auto zero
! time interval for readings
! number of readings
! storage type

! file name
! assign data file
! set multimeter states

! collect data

! clear multimeter

```

NOTES

APPENDIX A: ERRORS AND WARNINGS

The following error numbers may be generated by SENDIO and ENTIO\$. They are listed here in numerical order. Included in the descriptions are the message number, the message text, the description of the error, and the suggested user response.

Error 63 -- Invalid filespec.

Description:

- 1) A device name was specified in the <device code> parameter of SENDIO or ENTIO\$, but no device with that name exists.
- 2) Error will occur when both parameters for ENTIO\$ are null.

User Response:

- 1) Correct the <device code> parameter to specify a device that exists, or use the ASSIGN IO command to assign the name to a device.
- 2) For ENTIO\$, make sure that at least one of the parameters is not null.

ERROR 89 -- Bad Parameter.

Description:

- 1) The <command list> of a SENDIO or an ENTIO\$ contains a command name that is not recognized as a valid HP-IL command.
- 2) The <command list> specified "LAD#" or "TAD#", but the <device code> field was null.

User response:

- 1) Correct the parameter so that it is a valid HP-IL mnemonic, or use the Extended Command capability to generate the desired frame.
- 2) Either specify a <device code>, or replace the "#" with a specific device address.

Warning 150 -- Device sent NRD.

Description:

1. An active listener has interrupted transmission during a SENDIO, by sourcing an NRD frame. The SENDIO has terminated, but program execution continues.

User response:

1. No response required. If you wish to complete the transmission, then use SEND? to find out where transmission was stopped, and resume the transmission.

NOTES

NOTE: Some HP-IL devices have a "standby" mode and can be "powered up" by the controller. Such devices normally do not retransmit the first frame they received when in the powered down state. If the HP-75 is in STANDBY OFF mode and you execute an ASSIGN IO or RESTORE IO, then the HP-75 will wait several seconds for the first frame to return, and then send out several more IDY's. It will beep and then the ASSIGN IO or RESTORE IO will be completed. If the HP-75 is in STANDBY ON mode and one or more devices are powered down, the HP-75 will wait indefinitely for the first frame to return. You will have to close the loop temporarily on itself and press [ATTN] to abort the ASSIGN IO or RESTORE IO. Then execute STANDBY OFF followed by ASSIGN IO or RESTORE IO.

Such devices are: HP 82161A Digital Cassette Drive
HP 82162A Thermal Printer/Plotter
HP 82169A Acoustic Coupler

NOTES

APPENDIX B: HP-IL COMMAND SUMMARY

This is a list of mnemonics for the HP-IL commands. If you wish to send a frame for which no mnemonic exists, use the Extended HP-IL Command Capability, described in Appendix C.

- n -- represents a one byte non-negative integral value
- AADn -- auto address the loop: address the loop starting with initial address n (0-30)
- AAU -- auto address unconfigure: reset addresses of the loop to unassigned state
- AEPn -- auto extended primary: assign primary address n(0-30) to extended address group
- AESn -- auto extended secondary: assign secondary address starting with n (0-30)
- AMPn -- auto multiple primary: assign primary addresses to all devices starting with n (0-30)
- CL+ -- carriage return, line feed delimiters: delimits each end of record sent by the active talker with CR/LF pair, only has meaning with ENTIO\$, no action taken if used in SENDIO
- DCL -- device clear: clears all devices on the loop
- DDLn -- device dependent listener: sets the listener to mode denoted by integer n (0-31)
- DDTn -- device dependent talker: sets the talker to mode denoted by integer n (0-31)
- EDN -- enable device sourcing NRD: sets device up to source its own NRD
- GET -- group execute trigger: sets listeners up to begin device operation
- GTL -- go to local: sets devices to local mode
- IAA -- illegal auto address: sent to determine if too many devices on the loop
- IEP -- illegal extended primary: basically a no-op
- IMP -- illegal multiple primary: sent to determine if too many devices on the loop
- IFC -- interface clear: clears the status of peripherals on the loop
- LL0 -- local lockout: sets devices up to not respond to local control

NOTES

LPD -- loop power down: puts devices in power down state
 LADn -- listen address: activate listener status of device at address n (0-30)
 LAD# -- listen address: activate listener status of device specified in device code
 NRD -- not ready for data: control interrupt of talker
 NRE -- not remote enable: sets devices up to respond only under local control
 NOP -- no op: basically a no op command (see HP-IL documentation)
 PPD -- parallel poll disable: causes listen addressed devices to no longer respond to PPEn
 PPEn -- parallel poll enable: enable devices to respond to parallel poll where n (0-15) sets the state of response (see HP-IL documentation)
 PPU -- parallel poll unconfigure: disable all devices from responding to PPEn
 REN -- remote enable: devices set to remote mode
 SADn -- secondary address: enable talkers or listeners with secondary address
 SDC -- selected device clear: clears the active listeners
 SAI -- send accessory ID: initiate talker to source accessory id
 SDA -- send data: initiate talker to source data
 SDI -- send device ID: initiate talker to source device id
 SST -- send status: initiate talker to source status byte(s)
 TADn -- talker address: activate talker status of device at address n (0-30)
 TAD# -- talker address: activate talker status of device specified in <device code>
 TCT -- take control: pass control to another controller on loop
 TL+ -- talker listener active: inhibits UNT and UNL at end of command
 UNL -- unlisten the loop: deactivate all listeners
 UNT -- untalk the talker: deactivate the talker
 ZES -- zero extended secondary: assign secondary addresses to devices with multiple address capability

NOTES

APPENDIX C: EXTENDED HP-IL COMMAND CAPABILITY

Extended HP-IL command capability allows the programmer to send frames for which no mnemonic exists. The capability can be used with both SENDIO and ENTIO\$. This ensures that when new HP-IL devices and functions are introduced, SENDIO and ENTIO\$ will continue to be usable.

Recall that HP-IL frames consist of 11 bits: a 3-bit prefix which identifies the type of frame, and 8 bits which contain the content of the frame. Eight possible prefixes exist, each of which has a special meaning. Extended command capability provides an easy way for the programmer to construct HP-IL frames.

Eight identifiers are supplied, one for each type of HP-IL frame. The types of frames and corresponding identifiers are listed below:

<u>Frame</u>	<u>Identifier</u>
1) Command.....	CD
2) Ready.....	RD
3) Data.....	DA
4) End.....	EN
5) Identify.....	ID
6) Data w/service request.....	DS
7) End w/service request.....	ES
8) Identify w/service request.....	IS

To send a frame, simply specify "XX:<hex value>" in the <command list>, where XX is one of the eight identifiers listed above, and <hex value> is the content of the frame in hexadecimal. To send an UNL frame using extended HP-IL command capability, you would code:

```
SENDIO '', 'CD:3F', ''
```

This would send a frame with a 3-bit prefix identifying the frame as a command frame, and then a binary "0011 1111", which is the code for UNL.

NOTES

NOTES

NOTES

NOTES

NOTES

I/O UTILITIES

BRIEF REVIEW OF HP-IL
NOTE ON SYNTAX USED IN THIS MANUAL
LOADING THE UTILITY
SENDING INFORMATION TO HP-IL DEVICES
RECEIVING DATA FROM THE LOOP
SAMPLE PROGRAMS

ALL HP-75 SOLUTIONS BOOKS ARE AVAILABLE RECORDED ON MINI-DATA CASSETTES
FROM EITHER A HEWLETT-PACKARD DEALER OR THE HP USERS' LIBRARY.

USE OF THIS PRODUCT REQUIRES AN UNDERSTANDING
OF THE HP-75 AND OF FRAME LEVEL HP-IL PROTOCOL.

