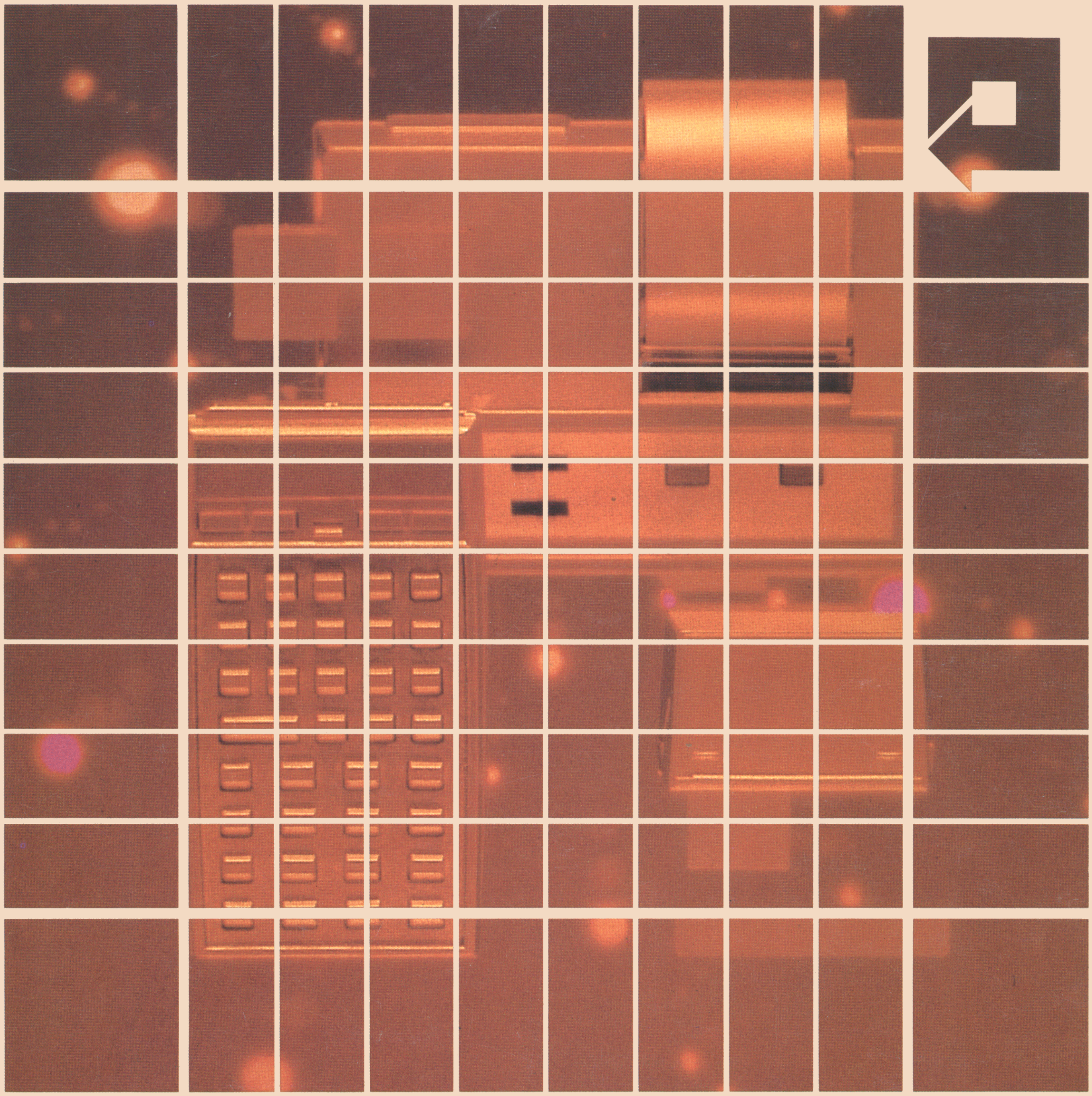
HEWLETT-PACKARD

HP 82183A

Extended I/O Module

OWNER'S MANUAL

For Use With the HP-41



HEWLETT PACKARD

HP 82183A
Extended I/O Module

Owner's Manual

For Use With the HP-41

December 1982

82183-90001

Printed in U.S.A.

Preface

This manual describes how to install the I/O module and how to use its functions in typical applications. Where you need assistance to understand how specific HP-IL peripheral devices respond to HP-IL module and I/O module functions, refer to the owner's manuals for those devices.

The use of many I/O module functions, particularly the advanced control functions described in section 5, requires you to understand the principles of HP-IL operation. Although this manual provides basic instructions for the I/O functions, and examples of their use, a complete discussion of HP-IL principles is beyond its scope. The HP-IL System: An Introductory Guide To The Hewlett-Packard Interface Loop, by Kane, Harper, and Ushijima (Osborne-McGraw Hill, Berkeley, 1982) is a sufficient introduction to HP-IL for users of the I/O module. Complete HP-IL definition and protocol are described in the HP-IL Interface Specification (HP part number 82166-90017).

Contents

Getting Started

Contents

Introduction

The HP 82183A Extended I/O Module provides 59 functions that enhance your HP-41 computer's control of the HP-IL system by expanding I/O (input/output) capabilities of the HP 82160A HP-IL Module. The I/O module's functions give you powerful control over a wide variety of HP-IL devices and allow HP-41 interaction with non-HP-IL instruments through HP-IL interfaces. In addition, the I/O module provides specific functions that interact with the file system established by the HP-IL module on standard mass storage devices such as the HP 82161A Cassette Drive.

Note: This manual assumes that you are familiar with the terminology and operation of the HP-41 and the HP-IL Module. For information about these topics, refer to the owner's manuals that describe these devices.

Installing and Removing the I/O Module

CAUTION

Be sure the HP-41 is turned off before installing or removing any modules. If this is not done, the HP-41 may be damaged or the system's operation may be disrupted.

The HP 82183A Extended I/O Module plugs into any HP-41 port. (If any HP 82106A Memory Modules are plugged in, the I/O module must be in a higher-numbered port than the memory modules.) Push in the module until it snaps into place. When you remove the module, remember to place a port cap over the unused port.

When the I/O module is installed in the HP-41, all of its functions become available. However, most of these functions operate on the HP-IL system, and therefore require that an HP-IL system be connected to the HP-41 and HP 82160A HP-IL Module. Executing an I/O module HP-IL function without the HP-IL module plugged into the HP-41 results in the error message NO HPIL.

Extended I/O Module Configuration and Characteristics

The I/O module functions are divided into four basic groups: mass storage functions (section 2), character manipulation functions (section 3), HP-IL control functions (section 4), and advanced control functions (section 5).

The functions contained in this module are grouped under four headers in the [CATALOG] 2 listing:

1. -X MASS 1A
2. -X EXT FCN
3. -X CTL FNS
4. -ADV CTL FN

The X on the first three headers is a reminder that the functions are extensions (or, in some cases, duplicates) of similar functions in the HP-IL module and the HP 82180A Extended Functions/Memory Module. (The 1A on the first header is the revision number for the I/O module.)

The mass storage functions operate with "standard" mass storage devices, such as the HP 82161A Digital Cassette Drive. A standard mass storage device is one whose HP-IL accessory ID is 16. (To determine a device's accessory ID, refer to the Send Accessory ID message in the owner's manual for that device.)

The character manipulation functions are designed to give you control over the contents of the ALPHA register when you use this register as an I/O buffer.

The HP-IL control functions provide convenient operation of the HP-IL system and operate with any type of HP-IL device. The tasks of loop addressing, designating devices as talkers or listeners, error checking HP-IL frames, etc., are carried out automatically by these functions. The description of each function includes the most significant HP-IL message used by that function. (Refer to the owner's manual for each device to determine the details of its response to that message.)

The advanced control functions provide the capability of transmitting any HP-IL command messages without any automatic loop operations. Proper use of these functions requires you to be familiar with the details of HP-IL protocol as described in the references given in the Preface on page 3.

Using This Manual

For simplicity, extended I/O module functions (and any other functions not on the standard HP-41 keyboard) are represented by single, colored keys--such as [DIRX]. When you want to execute a function, or to key it into a program, you can do it in either of two ways:

- Using [XEQ] [ALPHA] name [ALPHA].
- Assigning the function to a key using [ASN] and pressing that key on the User keyboard.

The description of each function is preceded by a summary of the information required for input and/or returned as output. (For some HP-IL control functions, the summary also includes the principal HP-IL message used by the function.) This provides a quick, visual reference for executing each function. For example:

[FINDAID] (Find Accessory ID)		HP-IL: Send Accessory ID (SAI)
Input : X	accessory ID	
Output: X	address	

This indicates that:

- The name of the function is Find Accessory ID.
- To find the address of a device on the HP-IL, you must place the device's accessory ID number in the X-register and execute [FINDAID].
- The key HP-IL message used by [FINDAID] is Send Accessory ID.
- The function's output, which is the specified device's address, is displayed in the X-register.

A function that requires an integer input (such as a character code or an HP-IL address) uses the integer portion of the input number and ignores any fractional part. If the sign of a number is not used as a parameter, the function ignores the sign of the input.

Examples of how to use most of the I/O module functions are provided in the main text and/or in appendix C.

I/O Module Functions and the HP-41 System

When a function returns a numeric value to the X-register, the effect on the stack is the same as for HP-41 functions. That is, if the function requires an input from the X-register prior to execution, the HP-41 replaces that input with the result of executing the function and places a copy of the input in the LAST X register. If the function requires no input from the X-register, the returned value is entered in the X-register (and the stack lifts--unless stack lift was disabled prior to execution). All I/O module functions enable stack lift after their execution.

The I/O module functions [ALENGIO] and [X<>FIO] are identical to the [ALENG] and [X<>F] functions, respectively, in the HP 82180A Extended Functions/Memory module. The suffix "IO" simply indicates the I/O module versions of these functions.

The "R" on the I/O module's [XTOAR] (X-to-ALPHA-Right) function (which is identical to the [XTOA] function in the extended functions/memory module) identifies it as the right-side counterpart of the I/O module's [XTOAL] (X-to-ALPHA-Left) function.

If at any time an error message is displayed by the HP-41 following an unsuccessful attempt to execute an I/O module function, refer to appendix A for an explanation of the error's cause.

> **Note:** When there is a transmission failure on the loop, up to approximately 40 seconds may elapse before the HP-41 displays an error message. This is because the computer waits for a response from the loop before displaying the error message. During this delay period the HP-41's keyboard may not respond to keystrokes.

Bar Code for Program Examples

Most of the programs in this manual are short enough to allow you to quickly enter them from the keyboard. However, if you use an HP 82153A Optical Wand in your HP-41 system, you may wish to use the bar code in appendix D (page 83) when you are ready to enter the program used in the example on page 12 or any of the program examples provided in appendix C.

Mass Storage Operations

Contents

Introduction

The I/O module provides several functions for manipulation of HP-41 files stored on standard mass storage devices; particularly the HP 82161A Digital Cassette Drive. These functions add to the mass storage capabilities already provided by the HP-IL module.*

Copying Files

The [COPYFL], [MCOPY], and [MCOPYPV] functions enable you to copy individual HP-41 files or the entire contents of a medium. <u>A minimum of two mass storage devices must be present in the HP-IL system for these three functions to operate.</u> You may wish to review the portion of section 4 in the HP-IL module owner's manual that describes the use of multiple mass storage devices on the loop.+

---

* The functions described in this section are designed to work only on HP-41 files created using HP-IL module functions. HP-IL module and I/O module mass storage functions will not work reliably on files created by other controllers, such as the HP-75.

The HP-IL control functions described in section 4, "HP-IL Control Functions," and section 5, "Advanced Control Functions," provide additional control over mass storage devices. However, the HP-IL module operates as if it has exclusive control over the mass storage device. Thus, if you use I/O module HP-IL control functions on a mass storage device, the device could inadvertantly be left in a state that may cause data loss when subsequent mass storage functions are executed.

+ Refer to the information provided under the following consecutive headings: 1) "Operation of the HP Interface Loop," 2) "Selecting an HP-IL Device," and 3) "Auto and Manual Modes."

This manual uses the term master to identify the medium containing a file or files to be copied and the term duplicate to identify a copied file or a medium on which a new file or files will be copied. The file copying functions follow the conventions established by the HP-IL module mass storage functions. That is, in Manual mode the master medium must be in the primary device--the device whose HP- IL address was specified in the most recent execution of the [SELECT] function. In Auto mode the master medium must be in either the primary device or the first mass storage device following the primary device.

Copying Individual Files

[COPYFL] (Copy File)

Input : X | address | ALPHA | filename |

Output: None

[COPYFL] copies a nonprivate file from the master device to the device with the HP-IL address specified in the X-register. The name of the file to be copied is taken from the ALPHA register. If the duplicate medium already contains a file having the same name as the source file, [COPYFL] halts and displays the DUP FL NAME error message.

Example of a Routine For Copying Files Between Mass Storage Devices. The following routine copies a file named MASTER from a mass storage device at HP-IL address 1 to a mass storage device at address 5.

```
01 1            Specifies address of master.
02 SELECT
03 5            Specifies address of duplicate.
04 "MASTER"     Specifies name of file to be copied.
05 COPYFL       Copies file MASTER from device at
                address 1 to device at address 5.
```

Duplicating an Entire Medium

[MCOPY] (Mass Copy)

[MCOPYPV] (Mass Copy-Private)

Input : -None-

Output: | number of records copied |

[MCOPY] copies the contents of a master medium onto all other media (termed the duplicate media) on the loop. [MCOPYPV] operates in the same way as [MCOPY], except that all HP-41 program files are made "private" as they are copied onto the duplicate media. Copying is performed on all media simultaneously, which provides maximum speed for the process of recording multiple duplicate copies.

[MCOPY] and [MCOPYPV] begin execution by formatting all of the duplicate media to match the format of the master medium. During this operation, the HP-41 displays the FORMAT message. When the actual copying begins, the COPY message is displayed. Unused records at the end of the master medium are not copied. When copying is completed, the X-register contains the number of 256-byte records actually copied--and not the number of files. The number of records is intended as the input for the [MVERIFY] function.

Before attempting to use [MCOPY] (or [MCOPYPV]), be sure that you understand the following points:

- The batteries in any HP 82161A Digital Cassette Drive should be well charged before you execute a copying function. If the power fails on any drive in the loop during the operation, some or all of the contents of the duplicate media may be invalid.

- If the master medium contains any "private" program files, [MCOPY] halts, displays PRIVATE, and does not attempt to format the duplicate media.

- If the cassette door on any HP 82161A Digital Cassette Drive (other than the master drive) is open when you execute a medium copying function, the tape cassette in that drive will be unaffected by the copying operation. However, opening a cassette door during execution of a medium-copying function halts execution of the function and invalidates all duplicate media.

- [MCOPY] formats all of the duplicate media to match the format of the master medium. This destroys any previous information recorded on those media, so it is very important that you ensure that your master medium is installed in the correct drive. To minimize the chance of error, you should normally use [SELECT] to make the primary device the master mass storage device. A simple way to verify this manually for HP 82161A Digital Cassette Drives is to execute [RCLSEL] [LISTEN] from the HP-41 keyboard. ([RCLSEL] is described on page 32 in section 4.) This operation causes the BUSY light on the master device to turn on if it is the primary device. You then have only to check that your master tape is in that drive. (For a demonstration of this operation, refer to the next example.)

Example Using the BUSY Light to Indicate the Master Device. The following program, adapted from the 00041-15042 Automatic Start and Cassette Duplication Module, blinks the BUSY light on the master drive (which should be at address 1) ten times prior to executing [MCOPY]. (Bar code for this program is provided on pages 83-84 in appendix D.)

```
LBL MSCOPY
1.01
SELECT
```

Selects primary device.

```
AUTOIO
CF 21
"MASTER IN"
AVIEW
PSE
"DRIVE WITH"
AVIEW
PSE
"FLASHING"
AVIEW
PSE
"BUSY LIGHT?"
AVIEW
PSE
"IF NOT PRESS"
AVIEW
PSE
"R/S"
AVIEW
PSE
```

Prompts you to halt program execution by pressing [R/S] if the master medium is not in the drive indicated by the flashing BUSY light.

```
LBL 01
RCLSEL
LISTEN
31
LISTEN
RDN
RDN
ISG X
```

Causes BUSY light on primary device to blink and removes all devices on loop from listener status.

```
GTO 01
"LAST CHANCE"
AVIEW
PSE
PSE
PSE
```

Pauses program before commencing copying operation.

```
FS? 55
SF 21
```

Restores printing output.

```
MCOPY
```

Executes copying operation (unless master tape contains any "private" files).

```
MVERIFY
```

Verifies that contents of master medium were correctly copied onto duplicate media.

```
"DONE"
BEEP
PROMPT
GTO "MSCOPY"
```

Prompts you when medium copying and verifying is completed. If you press [R/S], program re-executes.

Verifying Media

[MVERIFY] (Mass Verify)

Input : X | Number of Records |

Output: None

[MVERIFY] checks each mass storage device on the loop to verify that the specified number of records on its medium can be read without error.* (Any cassette drive with its cassette door opened prior to execution of [MVERIFY] is ignored.) [MVERIFY] is intended for use following [MCOPY] or [MCOPYPV], which place the number of records copied into the X-register. If you are uncertain of the number of records currently in use on any of your media, use 512 as the number of records so that the entire media are verified.

[MVERIFY] checks all media on the loop simultaneously. During execution, the HP-41 displays VERIFY. When [MVERIFY] finishes reading all of the media, it restores the normal X-register display if it has found no medium errors. However, if [MVERIFY] detects an error, the HP-41 displays the message DEV nn ERR, where nn is the HP-IL address of the device on which the error was found. If there are multiple errors, successive error messages will appear in the display (and be printed, if a printer is present).

Obtaining Directory Information

The file system established by the HP-IL module on a mass storage medium uses one or more medium records to store a file directory. (Refer to "The Storage Medium" in your HP-IL module owner's manual.) The directory on a medium contains the following information about each file on that medium:

- Filename
- File location.
- HP-41 file type.
- File length.

---

* [MVERIFY] is specifically designed for use with the HP 82161A Digital Cassette Drive. It can also be used with other standard mass storage devices provided such devices have an accessory ID of 16 and implement the same device dependent commands as the cassette drive. [MVERIFY] also requires that each mass storage device take a default HP-IL address of 2. Refer to the Auto Address Unconfigure message in the owner's manual for the device you are using.

The following three functions give you access to directory information needed for programs to automatically manipulate the files on the medium.

[DIRX] (Directory Entry X)

Input : X	file number		
Output: X	file number	ALPHA	filename

[DIRX] returns to the ALPHA register the name of the file whose position in the primary medium's directory is specified by the number in the X-register. The number in the X-register is unchanged unless that number exceeds the number of files actually on the medium. In this case, 0 replaces the number in the X-register, and the ALPHA register remains unchanged.

The file number in X must be nonzero (the sign and fractional part of the number are ignored.) Execution of [DIRX] when $\underline{x} = 0$ or $\underline{x} > 999$ results in the DATA ERROR message.

[FLTYPE] (File Type)

Input : None		ALPHA	filename
Output : X	file type		

[FLTYPE] places into the X-register a two-character Alpha string representing the type of the file named in the ALPHA register. The Alpha string corresponds to one of the file type abbreviations used by the HP-IL module's [DIR] function:

[DIR] Abbreviation	File Type
PR	Program
DA	Data
KE	Key Assignment
ST	Status
WA	"Write-All"
AS	ASCII
??	Unknown

If the file named in the ALPHA register does not exist on the medium, execution of [FLTYPE] causes the HP-41 to display FL NOT FOUND.

Example of a Routine That Searches for a Specified File Type. The following routine copies any data files currently on a master mass storage medium onto a duplicate medium, but ignores all other types. The master device is at HP-IL address 1 and the duplicate device is at address 2.

Program	Description
01 LBL "DACOPY"	
02 1 03 SELECT	Selects the master device.
04 1.447 05 STO 00	Stores a loop counter for the maximum directory size.
06 "DA" 07 ASTO 01	Stores the Alpha string that specifies "data" file type.
08 LBL 00	Begins loop that checks for file type.
09 RCL 00 10 DIRX	Recalls and uses loop counter to obtain file name.
11 X=0? 12 RTN	Tests for end of directory. If file number is zero, all files have been checked and program halts.
13 RCL 01	Recalls DA to X.
14 FLTYPE	Obtains file type in X and lifts DA to Y.
15 X#Y? 16 GTO 01 17 2 17 2 18 COPYFL	Tests "not a data file?". If file type in X is not DA, returns execution to [LBL] 01 to begin check of next file. If file type in X is DA, specifies duplicate device address and copies file named by string placed in ALPHA by preceding execution of [DIRX].
19 LBL 01	
20 ISG 00	Increments file counter to next file listed in master directory.
21 GTO 00 22 END	

If you omit lines 6, 7, 13 through 16, and 19, DACOPY then copies all of the files on the master medium to the duplicate medium. If the duplicate medium is newly formatted, the files will be stored in the minimum number of records possible on the new medium.

[FLLENG] (File Length)

Input : ALPHA | file name |

Output: X | file length |

[FLLENG] returns the length of the file named in the ALPHA register. The meaning of the "length" value depends on the file type:

File Type	Length
PR	Number of Bytes in the Program
DA	Number of Registers
KE	Number of Registers
ST	Number of Registers
WA	Number of Registers
AS	Undefined
??	Unknown

If the named file does not exist on the medium when you execute [FLLENG], the HP-41 displays FL NOT FOUND.

Example of Keystrokes to Use For Determining Directory Information. Suppose that a medium has the following directory, as listed by the HP-IL module's [DIR] function:

Name	Type	Regs
CODE	PR	32
FLY	PR	70
BLKBOX	WA	336
FBDATA	DA	32
NNN	KE	13
DECODE	PR	23
KOUT	AS	42

Use the following keystrokes to obtain directory information.

Keystrokes	Display	
1	1_	Obtains name of first file
[DIRX]	1.0000	listed in directory.
[ALPHA]	CODE	Displays name of first file,
[ALPHA]	1.0000	which is a program file. (Refer to preceding listing.)
[FLLENG]	221.0000	Displays number of bytes in CODE program. (Because HP-41 uses 7 bytes per register, the CODE program uses 31 registers and 4 bytes.)
4	4_	
[DIRX]	4.0000	
[ALPHA]	FBDATA	
[ALPHA]	4.0000	
[FLTYPE]	DA	Displays file type; in this case, a data file.
[FLLENG]	32.0000	Displays number of registers used in FBDATA file.
20	20_	Specifies 20th file listed in
[DIRX]	0.0000	directory. Displayed zero indicates fewer than 20 files on medium.

Character Manipulation Functions

Contents

Introduction

The functions described in this section are designed for interchanging characters or bytes between the X-register and the ALPHA register. This gives you the ability to construct arbitrary strings in ALPHA to be sent to HP-IL devices, and to decipher strings read into the ALPHA register from these devices.

The ALPHA Register as an I/O Buffer

The ALPHA register is a specially reserved portion of the HP-41 memory that was designed for displaying and manipulating Alpha strings and User prompts. The ALPHA register's size (24 characters) and the availability of HP-41 Alpha functions make this register suitable for use as an "I/O buffer" for HP-IL operations. That is,the ALPHA register can serve as a temporary storage location for strings of bytes you want to transmit or receive on HP-IL. However, in order to use the ALPHA register successfully as an I/O buffer, you need to understand the effects of some ALPHA register features that affect its behavior for I/O applications.

Each character in the ALPHA register is represented within the HP-41 by a character code (numbered from 0 through 255), which is based on the ASCII (American Standard Code for Information Interchange) convention. Not all of the 256 character codes have unique HP-41 display characters. The table below lists the display characters and their character codes. All character codes not listed in the table are represented by the display character ▩.

Displayable Characters and Their Equivalent Codes

Char.	Code	Char.	Code	Char.	Code
¯	0	3	51	N	78
[illegible]	1	4	52	O	79
[illegible]	4	5	53	P	80
[illegible]	5	6	54	Q	81
[illegible]	6	7	55	R	82
μ	12	8	56	S	83
∡	13	9	57	T	84
≠	29	:	58	U	85
space	32	;	59	V	86
!	33	<	60	W	87
"	34	=	61	X	88
#	35	>	62	Y	89
$	36	?	63	Z	90
%	37	@	64	[	91
&	38	A	65	\	92
'	39	B	66	]	93
(	40	C	67	↑	94
)	41	D	68	_	95
*	42	E	69	⊤	96
+	43	F	70	a	97
,	44	G	71	b	98
−	45	H	72	c	99
.	46	I	73	d	100
/	47	J	74	e	101
0	48	K	75	Σ	126
1	49	L	76	⊦	127
2	50	M	77		

The ALPHA-to-X transfer functions described later in this section illustrate the relationship between Alpha display characters and their character codes. For example, the keystroke sequence [CLA] 65 [XTOAR] places an A (character code 65) in the ALPHA register--it equates 65 in the X-register with the A in the ALPHA register.

The **null** character (character code 0) plays a special role in Alpha displays. In order to understand its behavior, visualize the ALPHA register as a fixed register having 24 character positions. The Alpha **display**, however, is a moving "window" through which you can view up to 12 character positions in the ALPHA register.

Executing [CLA] fills the ALPHA register with 24 nulls. When you subsequently enter an Alpha string, each character in the sequence enters the ALPHA register at the rightmost character position, thus pushing to the left all preceding characters in the string:

	ALPHA Register	ALPHA Display
Register Cleared	------------------------	
Register After Alpha Entry	-----------------STR¯ING	STR¯ING

As illustrated above, the display shows the contents of the ALPHA register, beginning with the leftmost non-null character. The beginning of the string is left-justified in the display (or scrolled left if it is followed by more than 11 characters). If nulls occupy one or more of the leftmost positions in the register, those **leading** nulls are ignored in the Alpha display. When a null is preceded on the left by any non-null characters, it is displayed as the overbar (¯) character.

Functions in the HP-41, the HP 82180A Extended Functions/Memory Module, and the HP 82160A HP-IL Module that use the ALPHA register also assume that the "useful" content of the ALPHA register starts with the first non-null character. For example, the [ALENG] function ignores leading nulls in determining the length of the string in ALPHA (if it didn't, it would always return 24). This can lead to some unexpected results when nulls are interspersed with non-null characters. When, for example, you use [APPEND] to add a character at the end of the ALPHA register, the HP-41 decides what to display by looking at the last character. If that character is a null, the HP-41 decides that the register is empty and shows a blank display even though the actual contents of the register remain intact. But if you execute [APPEND] or [ARCL] and then delete characters from the end of the Alpha string, the HP-41 assumes that the register is empty whenever the character you delete is a null (because it appears to have reached the leading nulls). Then the HP-41 clears the entire register.

When an Alpha string is stored in a data or stack register, embedded nulls are stored correctly, but the nulls are not shown in the display of the register obtained with [RCL] or [VIEW].

Finally, embedded nulls may cause unexpected results with functions built into HP-41 extensions. For example, nulls embedded in a filename in the ALPHA register are ignored by functions using that filename. As another example, the [OUTA] function will not transmit nulls on HP-IL--they're skipped.

Identifying leading nulls in Alpha strings is not possible because the HP-41 cannot distinguish between any nulls you might wish to include as the leading part of a string to be transmitted on the loop, and those nulls that are used to fill the empty positions in the ALPHA register.

Therefore, the functions described in section 4 that send Alpha strings out on HP-IL are intentionally designed to ignore not only leading nulls in the ALPHA register, but also the first non-null character. This allows you to place a "dummy" character at the beginning of the string you want to send. When you execute a function that sends the string, the dummy character is not sent--instead it defines the beginning of the string that you do wish to transmit. Similarly, functions that read bytes into the ALPHA register from HP-IL place a dummy byte (a D or an S) in front of the input string, which enables you to identify any leading nulls read from the loop.

ALPHA-X Transfer

You can transfer a character (a byte of information) back and forth between the ALPHA register and the X-register using any of the next six functions. Remember that when the byte is in the ALPHA register, it is displayed as one of the HP-41 display characters; when it is in the X-register, it is represented by a decimal number from 0 to 255. Numbers from -255 through 255 are allowable inputs for X-to-ALPHA transfers. (The signs and fractional parts of the numbers are ignored.) If the numbers are outside this range, the DATA ERROR message is displayed.

[XTOAR] (X-to-ALPHA Right)

Input :	X: character code	ALPHA	string
Output:		ALPHA	modified string

[ATOXR] (ALPHA-to-X Right)

Input :		ALPHA	string
Output:	X: character code	ALPHA	modified string

These two functions affect the right end of a string in the ALPHA register.

[XTOAR] appends the character corresponding to the code in the X-register to the end of the current Alpha string, shifting the previous contents of the ALPHA register one position to the left. (If the initial string is 24 characters in length, the first character in the string is lost when you execute [XTOAR].)

[ATOXR] is the reverse of [XTOAR]--it places the code of the last character of the Alpha string into the X-register and deletes the character from the string. If the ALPHA register is empty, [ATOXR] returns -1 to the X-register.

[XTOAL] (X-to ALPHA Left)

Input : X | character code | ALPHA | string |

Output: ALPHA | modified string |

[ATOXL] (ALPHA-to-X Left)

Input : ALPHA | string |

Output: X | character code | ALPHA | modified string |

[ATOXL] and [XTOAL] affect the left end of the string in the ALPHA register.

[XTOAL] adds a character corresponding to the code in the X-register immediately to the left of the first non-null character in the ALPHA register. If the original string in ALPHA already contains 24 characters when you execute [XTOAL], the rightmost character is lost and the string is shifted one position to the right to create space for the new character.

[ATOXL] removes the first character from the ALPHA string and places its code into the X-register. Notice that if the first character is followed by one or more consecutive nulls, those nulls become leading nulls and disappear from the remaining string. [ATOXL] returns -1 to the X-register if the ALPHA register is empty.

Example. This routine rotates a string of non-null characters in the ALPHA register to the right by the number of positions specified in the X-register.

```
01 LBL "AROR"          Rotates the ALPHA register.
02 INT
03 STO L
04 LBL 01
05 ATOXR               Takes the character from the right...
06 XTOAL               ...and puts it on the left side of ALPHA
07 DSE L               Done?
08 GTO 01              Branches to rotate again.
09 RTN
```

[ATOXX] (ALPHA-to-X by X)

Input : X	position	ALPHA	string
Output: X	character code	ALPHA	string

[YTOAX] (Y-to-ALPHA by X)

Input : Y	character code	ALPHA	string
X	position		
Output:		ALPHA	modified string

This pair of functions enables you to address any character position in the ALPHA register. Both functions require that you specify a

character position in the X-register. A positive character position indicates a position within the current ALPHA string, beginning with the leftmost non-null character as position 0, and numbering the remaining characters consecutively to the right. (This convention is consistent with that used by the [POSA] function in the extended functions/memory module.) A negative character position specifies an absolute position in the ALPHA register, independent of the current contents of the register. The positions are counted from the right, starting with -1 as the rightmost position, and going up to -24 as the leftmost position. The following table summarizes how [ATOXX] and [YTOAX] interpret character position.

Character Position	Character
n > 24 or n >= length of string	Not Valid (DATA ERROR)
0 <= n < length of string	Nth Character After Leftmost Character in String
n = 0	Leftmost Character in String
-24 <= n < 0	Nth Character From Right End of ALPHA Register
n < -24	Not Valid (DATA ERROR)

[ATOXX] finds the character in the position defined by the number in the X-register and returns its character code to the X-register. The string in the ALPHA register is unchanged.

The reverse procedure is performed by [YTOAX]--the character corresponding to the value in Y is placed into the ALPHA register at the position indicated by the number in X. The new character replaces the original character at that position.

A few examples should make the operations of these functions clear to you.

Keystrokes	Contents of ALPHA Register*
[ALPHA] STRING [ALPHA]	------------------STRING
85 [ENTER↑] 3 [YTOAX]	------------------STRUNG

* These and the following two examples represent the contents of the ALPHA register. The nulls to the left of the first non-null character would not appear in a display of the ALPHA register.

Keystrokes	Contents of ALPHA Register
79 [ENTER↑] 3 [CHS] [YTOAX]	------------------STRONG
88 [ENTER↑] 15 [CHS] [YTOAX]	---------X---------STRONG

And, to continue from the preceding keystroke series:

Keystrokes	Display	
0 [ATOXX]	88.0000	Character code for "S".
11 [ATOXX]	82.0000	Character code for "R".
20 [ATOXX]	DATA ERROR	Not enough characters in string.
1 [CHS] [ATOXX]	71.0000	Character code for "G".
10 [CHS] [ATOXX]	0.0000	Null.

Interpreting Alpha Strings

The [ALENGIO], [ANUMDEL], and [X<>FIO] functions are designed to help you decipher information contained in Alpha strings.

[ALENGIO] (Alpha Length)

Input : ALPHA [string]

Output: X [string length]

[ALENGIO] places into the X-register the length of the current string in the ALPHA register. Leading null characters are skipped. An empty ALPHA register corresponds to a length value of 0. [ALENGIO] duplicates the [ALENG] function in the Extended Functions/Memory module.

Example. The following routine capitalizes any lowercase letters found in an ALPHA register string. It uses [ALENGIO] to provide a loop counter equal initially to the number of characters in the string (which must not contain any null characters).

```
01 LBL "CAP"
02 ALENGIO          Counts characters in ALPHA register.
03 LBL 00
04 ATOXL            Places code of leading character into X.
05 97          }    The character codes for lower case letters
                    are in the range 97 ≤ c ≤ 122.
06 X>Y?
07 GTO 01           If not lower case character, go to [LBL] 01.
08 CLX
09 122
10 X<Y?
11 GTO 01           If not lower case character, go to [LBL] 01.
12 CLX         }    The character codes for upper case letters
13 32          }    are determined by subtracting 32 from their
14 -           }    lower case counterparts.
15 R↑
16 LBL 01
17 RDN
18 XTOAR            Restores capitalized letter to ALPHA.
19 RDN
20 DSE X
21 GTO 00           Branches if there are characters remaining.
22 END
```

[ANUMDEL] (ALPHA-to-Number, Delete)

Input : ALPHA [string]

Output: X [value from string] ALPHA [modified string]

[ANUMDEL] searches the current string in the ALPHA register for a number (represented in numerals) and returns to the X-register the value of the number. It also deletes all characters in the string from the start of the string through the last numerical character used. If the Alpha string contains more than one number separated by non-numeric characters, [ANUMDEL] uses only the first number. [ANUMDEL] is identical in operation to the [ANUM] function in the extended functions/memory module, except that [ANUM] does not alter the string.

The HP-41 considers execution of [ANUMDEL] to be a numeric entry, and sets flag 22 if a number is returned to the X-register. If the Alpha string contains no numeric characters, [ANUMDEL] clears the ALPHA register but has no effect on the stack.

The characters "+", "-", ",", ".", and "E" (for exponent) are interpreted by [ANUMDEL] as numeric or non-numeric characters according to their context in the Alpha string. An isolated "+", for example, is not treated as a numeric character. A "+" or "-" symbol immediately preceding, embedded in, or following a sequence of number digits will be interpreted exactly as if the symbols and numbers had been keyed into the X-register (with [CHS] representing "-" and [CHS] [CHS] representing "+".) For example, [ANUMDEL] returns the value -3425 if executed when the ALPHA register contains the string "34-2+5".

The status of the numeric display control flags (flags 28 and 29) determines how the Alpha string is interpreted by [ANUMDEL]. For example, if flag 28 and flag 29 are both set, commas are treated as digit separators. But commas are considered as non-numeric if flag 28 is set and flag 29 is clear. Suppose that the ALPHA register contains the string "PRICE: $1,234.5XYZ". The following table shows how the settings of flags 28 and 29 affect the results of executing [ANUMDEL]:

Flag 28	Flag 29	X-Register	Modified Alpha String
set	set	1,234.5000	XYZ
set	clear	1.0000	,234.5XYZ
clear	set	1,2345	XYZ
clear	clear	1,2340	.5XYZ

Example. The HP 7470A Graphics Plotter can send on HP-IL an ASCII character string that describes the current pen position. The string contains three integer numbers separated by commas: X,Y,P. X is the pen's x-coordinate; Y is the pen's y-coordinate; P has a value of 1 if the pen is down, or 0 if the pen is up. Suppose that the plotter has sent the string "123,456,1" to the HP-41's ALPHA register. You could use the following keystrokes to decipher the string:

Keystrokes	Display	
[SF] 28		Ensures that a comma is not interpreted as a radix.
[ANUMDEL]	123.0000	Displays X-coordinate.
[ANUMDEL]	456.0000	Displays Y-coordinate.
[ANUMDEL]	1.0000	Pen is down.

[X<>FIO] (X Exchange Flags)

Input : X | new flags value |

Output: X | old flags value |

[X<>FIO] exchanges the current number in the X-register with the decimal equivalent of the binary number represented by flags 00 through 07, which enables you to test individual bits in an eight-bit byte. Flags 00 through 07 are reset according to the value in the X-register when you execute [X<>FIO]. ([X<>FIO] is identical to [X<>F] in the Extended Functions/Memory module.) Each of the 8 flags can represent one binary bit--flag 07 is the most significant bit, flag 0 the least significant bit.

Flag	07	06	05	04	03	02	01	00
Value	128	64	32	16	8	4	2	1

For example, pressing 169 [X<>FIO] sets flags 7, 5, 3, and 0, and clears flags 6, 4, 2, and 1. This is because

$$169 = 128 + 32 + 8 + 1$$

The integer part of the number in the X-register when you execute [X<>FIO] must be in the range -255 <= $\underline{x}$ <= 255. (The sign is ignored.)

Example. The following routine uses [X<>FIO] to determine whether an HP 82162A Thermal Printer is out of paper. The printer is at address 1 on the loop. The function [STAT], which reads the printer status, is described in section 4.

```
01 1
03 SELECT          Selects the printer.
04 STAT            Reads 2-byte printer status into ALPHA.
05 1
06 ATOXX           Puts the first status byte into X.
07 X<>FIO          Stores the byte into the flags.
08 FS? 03          Bit 3 shows "out of paper."
```

If the result of the test in line 08 is true, the printer is indicating that it is out of paper (or has a jammed carriage). Line 09 should be a branch to a subroutine that is designed to handle the out of paper condition. You might wish to follow this sequence with another [X<>FIO], to restore the flags to their original states.

Section 4

HP-IL Control Functions

Contents

Introduction

The following three kinds of HP-IL control functions give the HP-41 the ability to control operations on HP-IL:

- Loop Configuration Functions: Enable the HP-41 to determine the current loop configuration, including the type, address, and status of each device.
- Device Control Functions: Implement several HP-IL commands that prepare loop devices for subsequent interaction.
- Data Transfer Functions: Control the transfer of data between devices, including input and output of data and programs from the HP-41 itself.

All operations on HP-IL involve the transmission around the loop of HP-IL messages (also called "frames"), each of which consists of 11 bits sent as a unit. When the messages carry data from one device to another, the transmitting device is known as a "talker"; the receiving device is a "listener." The "controller" device has the responsibility for assigning the talker and listener roles to loop devices, and also for sending the commands to start or terminate transmissions. In all operations described in this manual, the HP-41 is always the controller. The HP-IL module operates with the assumption that there are no other devices acting as controllers on the loop.

The functions described in this section operate in the same general manner as similar functions in the HP-IL module. That is, although each function is associated with a key HP-IL message, execution of the function causes transmission on the loop of a number of additional messages that precede and follow the key message. These "overhead" messages allow you to execute loop operations without concern for loop addressing or designating talkers or listeners. (These operations are performed automatically by the appropriate functions.*) Furthermore, the functions are designed to interact with the HP-IL module so that their execution can be single-stepped and traced using an HP-IL printer.

The key HP-IL message associated with each function is listed in the function summary preceding each function description. To determine the precise response of a particular device to the message, refer to the owner's manual for that device.

The control functions that interact with a single device always operate with the primary device, as specified by the HP-IL module's [SELECT] function. The selected address is stored in a special memory register in the HP-IL module. If you have not executed [SELECT] since installing the HP-IL module, the selected address is unpredictable. If there is no device with the selected address, the control functions use the device with address 1 as a default primary device. The operations of the control functions are unaffected by whether you are working in the HP-IL module's Auto or Manual I/O modes. That is, the functions do not search for a particular device type.

Loop Configuration Functions

The loop configuration functions enable the HP-41 to determine the current organization of the interface loop, including the number of devices on the loop, the address of the primary device, and the address, type, and status of each device.

---

* If you execute the [ADROFF] function, the automatic loop message sequences associated with the I/O module data transfer functions are disabled. You can restore normal operation using the [ADRON] function. For further information about these message sequences, refer to "I/O Module HP-IL Device Addressing Modes" on page 56 in Section 5.

[NLOOP] (Number on Loop) HP-IL: Auto Address (AAD)

Input : None

Output: X | number of devices |

[NLOOP] places into the X-register the number of devices currently connected on HP-IL, not counting the HP-41 itself. The number will be an integer from 0 to 30. Devices that do not respond to the Auto-Address message are not counted by [NLOOP]. You can use the number returned by [NLOOP] to control iterations in any program application that searches the loop for one or more devices.

[RCLSEL] Recall Selected Address

Input : None

Output: X | address |

[RCLSEL] returns the HP-IL address specified by the most recent execution of [SELECT]. This address is that of the primary device if:

1. [SELECT] has been executed since the HP-IL module was installed.

2. The loop contains a device having the selected address.

If the number returned by [RCLSEL] is greater than the number of devices on the loop, the primary device is the device at address 1.

Identifying Devices

HP-IL protocol provides two ways to identify a device on the loop:

- Device ID
- Accessory ID

The **device ID** is a string of data messages that may include the device model number and any other information that the device manufacturer wishes to provide. A Hewlett-Packard device ID is an ASCII-coded character string with the letters "HP" followed by a 5-character model number and a revision letter, and terminated by a carriage return/line feed. For example, the device ID for the HP 82095A printer is the string "HP82095A".

The **accessory ID** is a numeric code that identifies a device by its general type. The code is one eight-bit byte (decimal value from 0 to 255). Each group of 16 accessory ID numbers forms a device **class**, whose number (the first four bits of the accessory ID) can be obtained by dividing the accessory ID by 16 and taking the integer part. Each entry in a class is specified by the accessory ID modulo 16 (the last four bits of the ID).

Accessory ID		Device Class	
Decimal	Hexadecimal		
0-15	0-F	0	Controller
16-31	10-1F	1	Mass storage
32-47	20-2F	2	Printer
48-63	30-3F	3	Display
64-79	40-4F	4	Interface
80-95	50-5F	5	Instrument
96-111	60-6F	6	Graphics
112-239	70-EF	7-14	Not defined
240-255	F0-FF	15	Future extensions

For example, the accessory ID of the HP 82161A Digital Cassette Drive is 16, which makes it entry 0 in the mass storage device class. That is:

Accessory ID = 00010000
= 16_{10}

and

16 MOD 16 = 0

[ID] (Device ID) HP-IL: Send Device ID (SDI)

Input : None

Output: ALPHA | device ID |

[ID] returns to the ALPHA register a string containing the device ID of the primary device on HP-IL. If the device does not respond to the Send Device ID message (refer to the device owner's manual), the ALPHA register is cleared, and the HP- 41 display shows NO RESPONSE.

[ID] reads a maximum of 24 characters from the primary device. If the device ID has fewer than 24 characters, the HP-41 terminates input from the device when it receives an End-of-Transmission message. Carriage returns (character code 13) and line feeds (code 10) are deleted from the ID string placed in the ALPHA register.

[ID] is related to the HP-IL module function [FINDID], which finds the HP-IL address of a device that matches the device ID you specify in the ALPHA register. Notice that [FINDID] uses only the first seven characters of the string you enter.

Example. This routine tests whether the primary device is an HP 3468 voltmeter. If it is, the routine returns to a main program. Otherwise, execution halts and the HP-41 displays NOT A VOLTMETER.

```
01 LBL "VOLT?"            Is primary device a voltmeter?
02 SF 25
03 ID                     Get device ID.
04 FC?C 25
05 GTO 01         }       Branches if device does not return a
                          device ID
06 ASTO X                 Put ID into X.
07 "HP3468"               Look at first 6 characters in ID.
08 ASTO Y
09 X=Y?                   Is the primary device an HP3468?
10 RTN                    Yes
11 LBL 01
12 "NOT A VOLTMETER"      Display an error.
13 AVIEW
14 STOP
```

[AID] (Accessory ID) — HP-IL: Send Accessory ID (SAI)

Input : None

Output: X | accessory ID |

[AID] places the accessory ID of the primary device into the X-register. If the device does not respond to the Send Accessory ID message, the X-register is not changed and the error message NO RESPONSE is displayed.

[FINDAID] (Find Accessory ID)	HP-IL: Send Accessory ID
Input : X — Accessory ID	
Output: X — address	

[FINDAID] allows the HP-41 to locate a device of a specific class or type which you specify with a number in the X-register. If the number is positive, [FINDAID] searches the loop, starting with the primary device, until it finds a device whose accessory ID matches the number (integer part) in the X-register. If such a device is present, its HP-IL address is returned in the X-register.

If the number you enter in the X-register is negative, [FINDAID] searches the loop in the same way as described above. However, the value returned is the address of the first device found in the device class corresponding to the absolute value of the number in the X-register. Recall from the discussion under "Identifying Devices" (page 32) that

device class = INT[ABS(n/16)].

Thus, when n is negative, [FINDAID] matches only the first four bits of the accessory ID with the first (leftmost) four bits of the number in the X-register.

If there are no devices of the specified type or class present, 0 is placed in the X-register. DATA ERROR is displayed if [FINDAID] is executed with an input outside the range $-255 \leq n \leq 255$.

Example. FINDP returns the address of the first graphics device, starting from the primary device.

```
01 LBL "FINDP"        Find a plotter
02 -96                Look for a graphics device
03 FINDAID
04 RTN
```

Determining Device Status

The next six functions provide methods that enable you to determine whether a device on HP-IL needs service--that is, whether or not the device requires attention from the controller. A printer that is out of paper and a modem having an incoming transmission are examples of devices requiring service.

The HP-IL Identify message (refer to the discussion of HP-IL messages in an HP-IL device owner's manual) provides an HP-IL controller with a way to determine not only the existence of a service request on the loop, but also which device is signalling the request. Nine of the Identify message's bits can be set by devices to indicate service needs. The [POLL] function described on page 38 sends the Identify message with the nine bits initially cleared to zero.

One bit is termed the Service Request bit. Any HP-IL device needing service can set this bit as it passes the Identify message along the loop. The [SRQ?] function described next allows your HP-41 to read the Service Request bit. However, when the Service Request bit is set by any device on the loop, the controller has no information regarding which device is requesting service. The controller must then interrogate each device in turn to find the one that initiated the actual request. This technique is termed serial polling.

Some HP-IL devices implement parallel polling, which can greatly speed up the process of locating a service request. The remaining eight bits of the Identify message are used for parallel polling. A device with parallel poll capability has a special memory register in which it can remember a bit number (from 0 through 7) and also whether it is to respond to a parallel poll when it does want service, or when it doesn't want service. Using the parallel poll function [POLLE] (described on page 37), you can tell a device which of the eight Identify message bits you are assigning to it. Then when you execute the [POLL] function, the device uses its assigned bit to indicate its service needs. (Refer to the owner's manual for a particular device to determine how it responds to the Identify message.)

[SRQ?] (Service Request) HP-IL: Identify (IDY)

[SRQ?] is an HP-41 conditional operation* that tests the loop for the presence of a service request by sending the HP-IL Identify message. [SRQ?] follows the "do if true" rule. If any device signals a service request, the condition is true, and the next program line is executed; otherwise the next line is skipped. When executed from the keyboard, [SRQ?] displays either YES or NO.

Example. You can use [SRQ?] to make a program wait until a service request is detected, then branch to a subroutine that handles the request.

---

* Refer to the discussions of various conditional operations described in the HP-41 owner's manual.

```
01 LBL 00        Begins "wait" loop.
03 SRQ?          Checks for service request.
04 GTO 01        Branches to service request routine.
05 GTO 00        Branches for no service request.
06 LBL 01        Begins service request routine.
    .
    .
```

This routine provides a means for the HP-41 to prepare, for example, to read data from an HP 82168A Acoustic Coupler (which can request service when it has data available in its buffer). Notice that [SRQ?] provides no information about which device on the loop is requesting service.

[POLLE] (Parallel Poll Enable) HP-IL: Parallel Poll Enable (PPE)

Input : X | enable number 0-15 |

Output: None

[POLLE] enables the primary device to respond to a parallel poll, assigns one of the eight parallel poll bits to the device, and determines the sense in which the assigned bit is used by the device. The last two operations are determined by the enable number you specify in the X-register. The enable number must have a value between 0 and 15. (The sign does not affect the result.) The use of the enable number is shown in the following table:

Enable Number	Designates Bit Number	Significance of Bit When Set By Device
0	0	Service is not requested.
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	0	Service is requested.
9	1	
10	2	
11	3	
12	4	
13	5	
14	6	
15	7	

Enable numbers from 0 to 7 instruct the device to set the designated bit to 1 when it receives the Identify message and it does not require service.

Enable numbers from 8 to 15 instruct the device to set its designated bit to 1 when it receives the Identify message and it does require service.

Because the [POLLE] function does not return any information from the primary device, the HP-41 cannot know whether the device responds. Therefore, if the device does not implement parallel poll capability, no error message is displayed when you execute [POLLE]. Notice that you can assign the same identity bit to more than one device.

[POLL] (Parallel Poll) — HP-IL: Identify (IDY)

Input : None

Output: X | response |

[POLL] sends an Identify message around the loop and returns to the X-register a number from 0 to 255 representing the the loop's response. The response number is the decimal equivalent of the eight parallel poll bits. [POLL] initially sets those bits in the Identify message to "0". If any loop device has been enabled previously by [POLLE], it can change its assigned bit to 1 according to its service request needs, and the number returned to X may differ from 0. If you execute [X<>FIO] immediately after executing [POLL], HP-41 flags 00-07 are set or cleared according to the states of the corresponding Identify message bits.

[POLLD] (Parallel Poll Disable) — HP-IL: Parallel Poll Disable (PPD)

Input : None

Output: None

[POLLD] disables the parallel poll response of the primary device, thus cancelling the effect of a previous [POLLE] directed to that device. (After [POLLD] is executed, the primary device does not respond to subsequent executions of [POLL].)

[POLLUNC] (Parallel Poll Unconfigure) — HP-IL: Parallel Poll Unconfigure (PPU)

[POLLUNC] disables the parallel poll responses of all devices on HP-IL.

Example. The following routine assigns identity bit 0 to an acoustic coupler on HP-IL. (Notice that an HP 82168A Acoustic Coupler must be already configured to enable parallel poll capability--refer to the Acoustic Coupler owner's manual.)

```
01 LBL "CCON"
02 65                 Coupler's accessory ID.
03 FINDAID
04 STO 00             Saves address in R00.
05 SELECT             Selects the coupler.
06 8         }
07 POLLE     }        Enables bit 0 of the poll byte.
08 RTN
```

If the coupler is configured by the preceding routine, you can use the next program when a service request is detected.

```
01 LBL "CCHK"         Begins coupler check program.
02 POLL               Polls loop.
03 X<>FIO    }        Tests whether modem is requesting
04 FC? 00    }        service.
05 GTO "OTHER"        Branches to a routine to handle
                      other devices.
06 RCL 00    }
07 SELECT    }        Selects the coupler.
.  .
.  .                  Process coupler request.
.  .
```

[STAT] (Status) — HP-IL: Send Status (SST)

Input : None

Output: ALPHA [S + status string]

[STAT] reads up to 23 bytes of status from the primary device and stores the bytes as a character string in the ALPHA register. The string will always be preceded by a dummy character S (for "status"). The dummy character is necessary because the status string may contain one or more leading null characters that would otherwise be undetected in the ALPHA register.

[STAT] is a variation of the HP-IL module function [INSTAT]. [INSTAT] reads one byte of status, placing the value of that byte into the X-register and setting flags 00-07. You can use [INSTAT] if you are interested in only the first byte of a device's status. To access all of the multibyte status used in some devices, you must use [STAT].

It is important to realize that many devices that set status bits to indicate a device condition may change those bits after the device's status is read. Therefore, whenever your program reads a device's status, it must be prepared to respond to any condition that the status indicates as needing service. For example, when you open, then close, the cassette door on the HP 82161A Digital Cassette Drive, the drive sets its status byte to indicate that a new tape has been installed but not positioned. If you execute [STAT] (or [INSTAT]) with the cassette drive as the primary device, the new tape indication is cleared. Then, when you execute an HP-IL module or I/O module cassette function, the new tape will not be positioned correctly, which results in a possible loss of data from the tape.

Device Control Functions

Device control functions perform HP-IL operations that send instructions to HP-IL devices to affect their internal operating modes. The functions do not transmit data to or from the devices, but rather prepare the devices for sending or receiving data or for other loop processes. The precise response of any HP-IL device to these functions depends on the design of the device. [LOCAL], [PWRDN], [PWRUP], [REMOTE], and [TRIGGER] are device control functions provided in the HP-IL module.

Clearing Devices

[CLRDEV] (Clear Device)	HP-IL: Selected Device Clear (SDC)

The [CLRDEV] function "clears" the primary device, that is, instructs the device to return to a specific initial state. The actual response of the device depends upon its design. Typically, the clear state is the same as the initial power-on state. For example, [CLRDEV] causes the HP 82162A Thermal Printer to position its carriage to the right, clear its print buffer (allowing you to remove unwanted characters from the buffer without printing them), and set escape, single-wide, left-justify, and nonparse modes.

Example. The following program clears all display-class devices on HP-IL.

```
01 LBL "CLRD"          Program to clear displays.
02 1                 }
03 SELECT            }  Starts with first loop device.
04 LBL 01
05 -48               }
06 FINDAID           }  Finds next display.
07 RCLSEL
08 X>Y?
09 RTN                  Returns if there are no more displays.
10 RDN
11 SELECT               Selects the next display.
12 CLRDEV               Clears the display.
13 1                 }
14 +                 }  Prepares to search for the next display.
15 SELECT            }
16 GTO 01
```

[CLRLOOP] (Clear Loop)	HP-IL: Device Clear (DCL)

[CLRLOOP] clears all devices on the loop simultaneously.

Using Remote and Local Modes

Two I/O module functions, [NOTREM] and [LOCK], use HP-IL messages that control the responses of devices that must function in two modes: Remote mode and Local mode.

There are two distinctions between Remote mode and Local mode. The first is the manner in which devices respond to data messages transmitted on HP-IL. In Local mode, data messages (such as strings of ASCII characters) are treated as simple data by the devices. However, when in Remote mode, a device may treat as instructions to itself the data it receives from the controller. For example, an HP 82168A Acoustic Coupler set to Local mode loads character strings received from the loop into its output buffer for subsequent transmission through the telephone. In Remote mode, the coupler interprets ASCII strings as instructions. For instance, the modem executes a self-test when it receives the ASCII character "T".

The second difference between the modes is that in Local mode a device responds to its own manual controls, such as built-in switches or pushbuttons. In Remote mode the manual controls are inoperative (with the possible exception of a remote override switch).

The HP-IL module's [REMOTE] function places the primary device into Remote mode (if the device implements the two modes). [REMOTE] first sends the Remote Enable (REN) message, which puts each device having remote capability into the "remote-enabled" state. [REMOTE] then temporarily makes the primary device a listener so that it switches to Remote mode. (A remote-enabled device switches to Remote mode when it is made a listener.)

The HP-IL module also provides the [LOCAL] function, which uses the HP-IL Go To Local message (GTL) to return the primary device to local mode (after the device has received any remote instructions and is ready to resume ordinary data reception). However, the device remains in a remote-enabled state (as do all other devices on the loop), and it returns to Remote mode from Local mode any time it is made a listener by the controller. This occurs automatically when you execute functions that send data to the device.

[NOTREM] (Not Remote)	HP-IL: Not Remote Enable (NRE)
Input : None	
Output: None	

[NOTREM] returns all loop devices to Local mode and disables them from switching to Remote mode. [NOTREM] affects all devices on the loop that use Local and Remote modes.

For an example of [NOTREM] usage, refer to the ANS and ORIG programs on pages 79 and 80 in appendix C.

[LOCK] (Lock Out)	HP-IL: Local Lockout (LLO)

[LOCK] disables the remote override switch that allows you to manually force an instrument into Local mode. For [LOCK] to have an effect on a device, the device must already be in Remote mode. Once disabled, the override switch remains disabled until [NOTREM] is executed.

Sending Device-Dependent Commands

Device-dependent command messages are HP-IL messages directed to individual devices. The response to a device-dependent command is determined by the device. For example, device-dependent Listener commands may change the manner in which a device interprets subsequent HP- IL data messages. Device-dependent Talker commands may similarly prepare a device to transmit data to the controller. Either type of command can be used to execute internal device operations. Both types of messages contain a number from 0 to 31, enabling each device to distinguish 32 different talker and listener commands.

Examples of devices that use device-dependent commands are the HP 82161A Digital Cassette Drive and the HP 82165A HP-IL/GPIO Interface. The cassette drive, for example, rewinds its tape when it receives a Device Dependent Listener 7 message. A Device Dependent Talker 0 message prepares the HP-IL/GPIO interface to transmit the 19 data bytes contained in its control buffer.

[DEVL] (Device-Dependent Listener) — HP-IL: Device-Dependent Listener (DDL)

Input : X | command number |

Output: None

[DEVT] (Device-Dependent Talker) — HP-IL: Device-Dependent Talker (DDT)

Input : X | command number |

Output: None

[DEVL] and [DEVT] send the Device-Dependent Listener and Device-Dependent Talker commands, respectively, according to the number in the X-register. The commands are sent to the primary device. The command number must be in the range $-31 \le n \le 31$. Numbers outside the allowed range will result in the DATA ERROR display.

Example. The following sequence clears the transfer buffer on an HP 82165A HP-IL/GPIO Interface:

```
01 64
02 FINDAID                 Returns interface address.
03 SELECT                  Makes interface the primary device.
04 2          }            Sends device-dependent command 2 to
05 DEVL       }            clear buffer.
```

Data Transfer Functions

An HP-IL data transmission consists of a transaction between a single talker and one or more listeners, either of which type can be the controller itself. The I/O module can control three types of data transmissions:

- Data output from the HP-41 to an HP-IL device (the OUT functions).
- Data input to the HP-41 from a device (the IN functions).
- Data transferred from device to device without being stored in the HP-41 (the XFER functions).

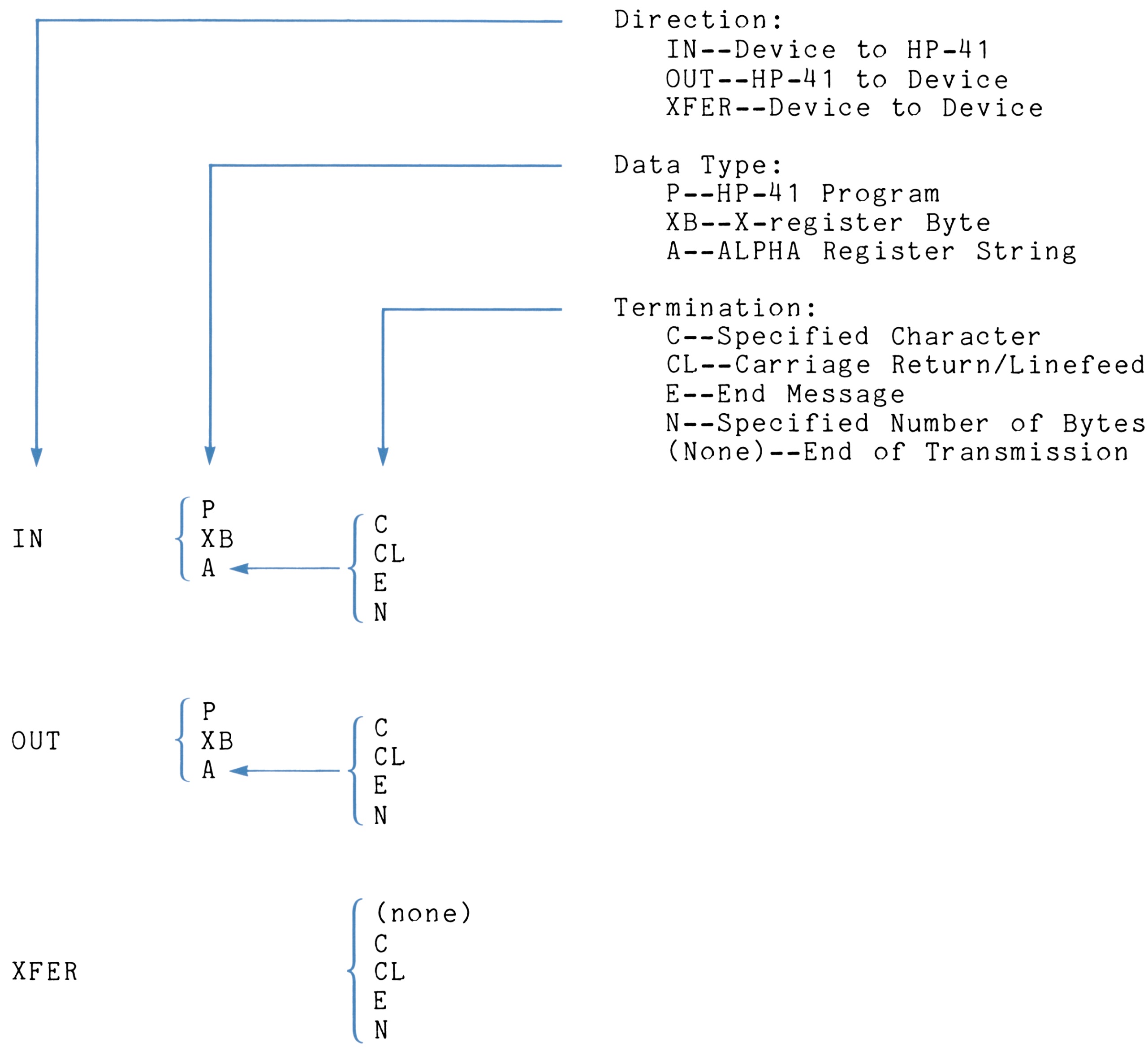
In any of these operations, data is sent around the loop one byte at a time, with each byte represented by a single HP-IL data message.

Once a transmission is begun, there are several ways that it can be terminated normally (that is, other than with some kind of transmission error). In the following material you will see that the data transfer functions provide five methods of terminating a transmission. During execution of an I/O module data transfer function, if a talker sends an End of Transmission message to indicate, for example, that its Send buffer is empty, the HP-41 continues to send the Send Data message until the termination condition is satisfied. (The one exception is the [XFER] function, which is designed to terminate on the End of Transmission message.) When the termination condition is satisfied, the HP-41 stops the transfer by sending the Not Ready for Data message (NRD).

If there is a delay of approximately 40 seconds during the execution of any data transfer function (that is, no messages are transmitted on HP-IL for 40 seconds), the function halts and displays TRANSMIT ERROR.

The name of each of the data transmission functions consists of one or more parts. Identifying these parts can help you remember the behavior of each function.

Names of Data Transfer Functions



Thus, for example, you can translate [INACL] into IN + A + CL, to see that [INACL] reads data bytes from HP-IL into the ALPHA register and continues to add data bytes to the string in the ALPHA register until a carriage return/line feed is received.

Functions that input data to the ALPHA register stop data transfer with a Not Ready for Data message (NRD) when the ALPHA register is full. The functions use flag 17 to indicate whether the data transfer is complete--that is, whether the termination condition has been met. Flag 17 is cleared if the condition has been met. Flag 17 is set if the ALPHA register fills up first. (Flag 17 is unaffected if data transfer is stopped by a transmit error.)

The data transfer functions ([INAC], [OUTAC], [XFERC], [INXB], [OUTXB]) that use a character code in the X-register or return a character code to the X-register make a conversion between an 8-bit byte transmitted on HP-IL and a decimal number stored in the X-register. Thus, a value returned to the X-register always lies in the range between 0 and 255. A number taken from the X-register to be sent on HP-IL must have a value in the same range, ignoring the sign and fractional part, or the DATA ERROR message is displayed.

The functions that use the ALPHA register as an input/output buffer use a dummy character at the start of the ALPHA string to define the beginning of the string. (This allows you to use strings having leading null characters.) The OUT functions delete the leading byte of the string before it is transmitted. You must therefore add an extra non-null character to the beginning of the string you wish to send. (These functions return the DATA ERROR message if the ALPHA register is empty.) The IN functions place a D character at the beginning of any string input from HP-IL. The D serves only to identify the beginning of the string; it does not represent data actually received from the primary device.

Example. To illustrate the significance of the various termination conditions, suppose that the primary device on HP-IL is ready to send the following block of data (where "cr/lf" indicates carriage return/linefeed):

```
12.3 cr/lf
45.6 cr/lf
78.9 cr/lf
```

The final linefeed is to be sent as an End message to signal the end of the block of data. Then the following keystrokes produce the results shown:

Keystrokes	Display of Alpha Register	
[INACL]	D12.3	Sends data until linefeed encountered.
2 [INAN]	D45	Sends two characters.
13 [INAC]	D.6	Character 13 = carriage return (cr).
[INACL]	D	Sends until linefeed encountered.
[INAE]	D78.9⊾▦	Sends until End message encountered. (Places carriage return/linefeed--cr/lf--characters in ALPHA register.)

Transmitting a Single Byte

[INXB] (Input Byte to X)

Input : None

Output: X | byte value |

[INXB] directs the primary device to send one byte of data to the HP-41. The decimal equivalent of the byte, from 0 to 255, is placed in the X-register.

[OUTXB] (Output Byte From X)

Input : X | byte value |

Output: None

[OUTXB] converts a number in the X-register into an equivalent 8-bit byte and sends it to the primary device.

As an example of [OUTXB] operation, you can use the following two steps to change the HP 82162A Thermal Printer from Escape mode to Eight-Bit mode:

1. Select the printer.
2. Execute 27 [OUTXB] 124 [OUTXB].

To switch back to Escape mode, execute 252 [OUTXB].

Transmissions That Stop at a Specific Character

[INAC] (Input to ALPHA Until Character)

Input : X | character code |

Output: ALPHA | string |

[INAC] clears the ALPHA register and reads a sequence of data bytes from the primary device into the ALPHA register, preceded by a dummy character (D). Characters are added to the ALPHA string until a character specified in the X-register is received, or until 23 characters are received (not counting the dummy D). The specified character is not included in the ALPHA string. Flag 17 is cleared if the specified character is received; flag 17 is set if the ALPHA register fills without receiving the character.

For example, suppose an HP-IL device has a data buffer containing the character string "HELLO?XYZ". Assume that the device has been prepared to transmit the buffer by a device-dependent command or an ASCII-coded command string. If 63 is placed in the X-register, [INAC] would place the string "DHELLO" into the ALPHA register. Since 63 is the character code for "?", the HP-41 stops the transmission when it receives the "?" character. You can remove the D by executing [ATOXL].

[OUTAC] (Output ALPHA and Character)

Input : X [character code] ALPHA [string]

Output: None

[OUTAC] sends the string in the ALPHA register--minus the first non-null character--to the primary device. The string is followed by a terminating character corresponding to the character code specified in the X-register.

For example, if you place the string DHELLO in the ALPHA register and the number 63 in the X-register, executing [OUTAC] causes six data bytes to be transmitted to the primary device. These data bytes then represent the characters HELLO? because [OUTAC] removes the leading D, and adds the trailing ? (character code 63).

[XFERC] (Transfer Data Until Character)

Input : Y [character code] X [receive address]

Output: None

[XFERC] transfers data from the primary device to a second device whose HP-IL address is specified in the X-register. The HP-41 stops the transmission when the character specified in the Y-register is sent. The destination device will receive the specified terminating character.

Transferring a Specified Number of Bytes

[INAN] (Input to ALPHA N Characters)		
Input : X	number of characters	
Output:		ALPHA string

[INAN] clears the ALPHA register, reads a specified number of bytes from the primary device, and places them into the ALPHA register following an initial D character. The number of characters to be input, from 0 to 23, is specified in the X-register. For programming convenience, the sequence 0 [INAN] is equivalent to 23 [INAN]. [INAN] does not affect flag 17.

[OUTAN] (Output ALPHA N Characters)		
Input : X	number of characters	ALPHA string
Output: None		

[OUTAN] transmits to the primary device up to 23 characters from the ALPHA register, where the number of characters to send is specified in the X-register. The characters transmitted are those following the first non-null character in the ALPHA string (the leading character is not sent), up to the number of characters specified. If the string in ALPHA has fewer characters than specified, then the entire string is sent (except for the leading character). [OUTAN] with 0 or 23 in the X-register always sends the entire ALPHA string, regardless of its length.

[XFERN] (transfer N bytes)	
Input : Y	number of bytes
X	receive address
Output: None	

[XFERN] instructs the primary device to send data bytes to the device at the address specified in the X-register. The number of bytes to be transferred is specified in the Y-register.

Transfers That Stop at an End Message

Data is usually transferred on HP-IL using Data messages. However, a talker may indicate that a byte is the last in a logical block of data by sending it as an End message. For example, a device may indicate the end of each line of data by sending the last character as an End message. The listener has the option of treating that character in a special way. Notice that an End message doesn't necessarily signal the end of data transfer.

[INAE] (Input to ALPHA Until End Message)

Input : None

Output: ALPHA | string |

[INAE] reads data bytes from the primary device until an End message is received. [INAE] clears the ALPHA register and places the data bytes, including the end byte, into the ALPHA register as a character string preceded by a D character. Flag 17 is cleared if the End message is received. The flag is set if the ALPHA register is full before the End message is received.

[OUTAE] (Output ALPHA With an End Message)

Input : ALPHA | string |

Output: None

[OUTAE] sends the string in the ALPHA register--minus the first non-null character--to the primary device. The final character is sent as an End message.

If the primary device does not distinguish End messages from other Data messages, you can use [OUTAE] as a shorter equivalent to the sequence 0 [OUTAN]. Either method sends the entire ALPHA string without regard to its length.

[XFERE] (Transfer Data Until End Message)

Input : X | receive address |

Output: None

[XFERE] causes the primary device to send data to the device whose address is specified in the X-register. The HP-41 terminates the transmission when the primary device sends an End message. (The listener will receive the End message.)

Transfers That Stop at a Carriage Return/Linefeed

Many devices that send ASCII-coded data terminate data transmissions with a carriage return (character code 13) followed by a linefeed (character code 10). Each of the following three functions terminates data transmissions when a linefeed character is detected.

[INACL] (Input to ALPHA Until Line Feed)

Input : None

Output: ALPHA | string |

[INACL] clears the ALPHA register, reads a string of data bytes from the primary device and places them into the ALPHA register following a leading D character. Input stops when a linefeed (character code 10) is received (flag 17 cleared). Carriage returns (character code 13) are deleted from the string placed into the ALPHA register, as is the terminating linefeed. If 23 characters (not counting carriage returns) are input before a linefeed is detected, [INACL] stops the transmission and sets flag 17.

[OUTACL] (Output ALPHA and Carriage Return/Linefeed)

Input : ALPHA | string |

Output: None

[OUTACL] sends the string in the ALPHA register--minus the first non-null character--as data bytes to the primary device. A carriage return character and a linefeed character are sent following the ALPHA string characters.

[XFERCL] (Transfer Data Until Linefeed)

Input : X | receive address |

Output: None

[XFERCL] initiates data transfer from the primary device to the device specified by the HP-IL address in the X-register. The HP-41 stops the transmission when a linefeed character is sent. Carriage return characters are transferred with no special treatment.

Example. The program listed below instructs an HP 3468A Multimeter to make five measurements and to display the results on an HP-IL display device. The data is not recorded in the HP-41.

```
01*LBL "DM5"
02 "HP3468A"          Specifies multimeter device ID.
03 FINDID             Selects multimeter.
04 SELECT
05 REMOTE             Puts multimeter in Single Trigger mode.
06 "DT2"
07 OUTACL
08 NOTREM
09 48                 Finds display.
10 FINDAID
11 5                  Enters loop counter for five measurements.
12 X<>Y
13*LBL 01
14 TRIGGER            Instructs multimeter to take a measurement.
15 XFERCL             Transfers data to display.
16 DSE Y              Decrements loop counter.
17 GTO 01             Branches for next measurement.
18 END
```

Transfers That are Stopped By an End of Transmission Message

[XFER] (Transfer Data Until End of Transmission)

Input : X | receive address |

Output: None

[XFER] instructs the primary device to send data to the device whose HP-IL address is specified in the X-register. This function can be used with devices that send an HP-IL End of Transmission (EOT) message when they have finished transmission. The HP-41 uses the End of Transmission message to determine when to resume program execution after the data transfer is complete.

Transferring HP-41 Programs

The functions [INP] and [OUTP] give your HP-41 the ability to exchange programs directly between HP-41 memory and HP-IL devices. Programs are transmitted as a series of hexadecimal-coded ASCII data bytes. This means that each HP-41 program byte is encoded as two hexadecimal digits, each of which is sent as an ASCII character 0 through 9 or A through F.* This type of coding avoids sending data bytes that might be interpreted by certain devices as special instructions. (For example, modems may use certain characters for handshake signals. If one of these characters were transmitted as part of a program sequence, the modem transmission could be interrupted.)

[INP] (Input a Program)

Input : None

Output: None

[INP] instructs the primary device to send a series of data bytes which the HP-41 can translate into a program. This program replaces the last program in memory.

The first four bytes received must indicate the length in bytes of the program. After the four bytes are received, [INP] checks the HP-41

---

* The program bytes are preceded in a transmission by four data bytes that represent (in hexadecimal digits) the length of the program in bytes. The last byte of the program is followed by two bytes representing a program checksum.

to see if the program will fit into memory. If there is enough space in memory, the transfer continues and the bytes are added to program memory. If there is insufficient room, the HP-41 stops the transmission and:

- If [INP] was executed in a program, displays NO ROOM.
- If [INP] was executed from the keyboard, displays PACKING and TRY AGAIN.

The last two bytes received are the program checksum. If this checksum does not agree with the program bytes previously read, the new program in memory is cleared and the READ ERR message is displayed. READ ERR is also be displayed if there is a loop failure or an improper data byte (characters other than ASCII-coded hexadecimal digits) detected during execution of [INP].

> **Note:** A program executing [INP] halts when a read error is detected, even if flag 25 is set.

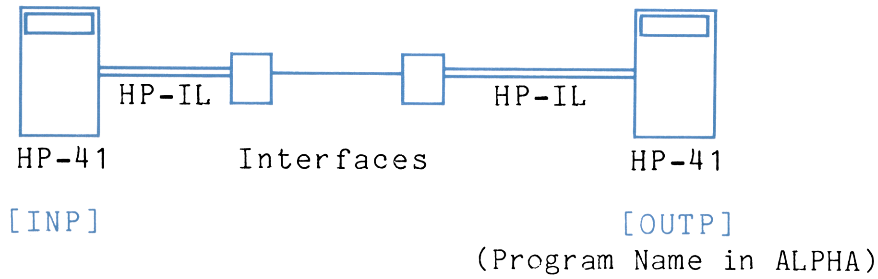
If [INP] is executed from the current last program in memory, the new program replaces the current program and program execution continues at the first line of the new program. If [INP] is executed from any program other than the last program, program execution continues in the current program at the line following [INP]. ([INP] is identical in these respects to [READP]. (Refer to the discussion of [READP] under "Executing Mass Storage Functions in Programs" in section 3 of the HP-IL module owner's manual.)

[OUTP] (Output Program)		
Input :	ALPHA	program name
Output: None		

[OUTP] finds the global label named in the ALPHA register and sends a sequence of data bytes representing the program containing that label to the primary device. The actual program bytes are preceded by four bytes representing the length of the program in bytes and are followed by two checksum bytes.

If the ALPHA register is empty, [OUTP] sends the current program. [OUTP] will not send a private program.

Suppose that you have two HP-41Cs connected to each other through an interface. You could move a program from one to the other by executing an [INP] in one HP-41 and an [OUTP] in the other.



Each interface must be the primary device in its loop and must be properly initialized. The initialization and timing of the transfer depends on the interfaces. An example of program transfer using the HP 82168A Acoustic Coupler is provided on page 77 in appendix C.

Advanced Control Functions

Contents

Introduction

Advanced control functions are designed to to enable the HP-41 to control certain HP-IL data transfers that are not possible using the data transfer functions in the manner described in section 4. In order to use the advanced control functions properly, you must understand HP-IL operation in detail, since you will be using the functions to replace loop operations that would otherwise be performed automatically during execution of data transfer functions. Advanced control functions are necessary only for a limited set of HP-IL operations. Thus, for most programmers, an understanding of the functions described in this section is optional.

I/O Module HP-IL Device Addressing Modes

All of the HP-IL operations described in the previous sections of this manual and in the HP-IL module owner's manual are executed by the modules' functions in a manner that emphasizes user convenience and programming simplicity. This means that before each function transmits its principal HP-IL messages, it also carries out some standard loop set-up procedures, including sending an Identify message to test loop continuity, assigning device addresses, checking the loop for the presence of a printer or display for tracing program execution, and designating devices as talkers or listeners as appropriate. At the end of execution, such functions remove all devices from talker or listener status. This behavior makes HP-IL devices, especially printers and standard mass storage devices, as easy to use as non HP-41 peripherals.

Some HP-IL devices and many HP-IB devices that can be connected to HP-IL through the HP 82169A HP-IL/HP-IB Interface cannot be fully operated in this manner. These devices change internal modes when they are designated as a talker or a listener. This can prevent complete data transfers, if the transfer requires execution of either more than one data transfer function or of a data transfer function plus a control function. For example, a user might employ the following sequence in an attempt to read the contents of the control registers of an HP 82165A HP-IL/GPIO Interface into the ALPHA register. (The interface is assumed to be the primary device.)

01 0	
02 DEVT	Instructs interface to send contents of its 19 control registers.
03 19	
04 INAN	Inputs 19 bytes to ALPHA register.

However, this sequence fails because [DEVT] clears the talker status of the interface after sending the device-dependent command message. [INAN] then makes the interface a talker again. This cancels the effect of the device-dependent command 0 (Send Control Registers), so that the interface sends 19 bytes from its data transfer buffer rather than from the control registers. (A successful version of the preceding routine is provided at the end of this section.)

To overcome such HP-IL control limitations (that is, limitations that can result from the user-convenience features of I/O module data transfer functions), the functions are actually designed to work in two modes (controlled by flag 34): Addressing-Off mode and Addressing-On mode. Addressing-On mode (flag 34 cleared), which you select by executing [ADRON], is the mode used implicitly in Section 4. This mode causes the functions that take care of loop addressing automatically. When you execute [ADROFF] (flag 34 set), the I/O module is placed in addressing-off mode. In this mode, data transfer functions will not auto-address the loop nor change the talker or listener status of loop devices. Other I/O module functions and HP-IL module functions (including [INA] and [OUTA] are not affected by flag 34.

The HP-IL Module has an additional HP-41 programming feature that complicates loop control. If there is a printer or display device on the loop and the loop is otherwise idle, the module continuously sends Identify messages (so that it can respond to the PRINT key on an HP-82162A Thermal Printer). The automatic transmission ceases when program execution begins, and resumes when program execution is completed. Furthermore, if the print function switch on the HP-IL module is set to ENABLE, then either at the beginning of program execution or when a program line is single-stepped, the module sends a series of messages to locate an HP 82162A printer and to determine if it is in NORMAL or TRACE mode. If the HP 82162A printer is in either of these modes, (or if another type of printer is present and flag 15 or 16 is set), additional HP-IL messages are sent to the printer during program execution.

[ADRON] (Addressing ON)

Input : None

Output: Clears flag 34.

[ADRON] enables the automatic loop addressing and talker/listener commands used by data transfer functions.

Flag 34 is the indicator used by the I/O module to control the Addressing-On/Off modes. [ADRON] clears flag 34 to indicate that Addressing-On mode is active. Flag 34 is cleared by a MEMORY LOST condition, but it is not affected by turning off the HP-41.

[ADROFF] (Addressing OFF)

Input : None

Output: Sets Flag 34

[ADROFF] sets flag 34, which places the I/O module in Addressing-Off mode. In this mode no automatic loop addressing or talker/listener commands are performed by I/O module data transfer functions (the IN, OUT, and XFER functions). Thus, in Addressing-Off mode, the data transfer functions cause only Send Data, Data or End, and Ready for Command messages to be transmitted on HP-IL.

The HP-IL module functions [INA] and [OUTA] are not affected by Addressing-On or Addressing-Off modes.

The HP-IL message sequences listed below illustrate the differences between Addressing-On and Addressing-Off modes. The following program sends the character string "ABCDE" (followed by a carriage return/linefeed) from the ALPHA register to a device at address 2:

```
01   ADRON
02   2
03   SELECT
04   "DABCDE"
05   OUTACL
```

During execution of [OUTACL], the HP-41 sends the following messages on HP-IL:

IDY 00	Checks loop continuity
AAU	Unconfigures the loop
RFC	
AAD 1	Auto-addresses the loop
LAD 2	Makes the primary device a listener.
RFC	
DAB 65	Sends data character "A".
DAB 66	" " " "B".
DAB 67	" " " "C".
DAB 68	" " " "D".
DAB 69	" " " "E".
DAB 13	Sends carriage return.
DAB 10	Sends linefeed.
UNL	Unlistens the loop.
RFC	

If however, you substitute [ADROFF] for the [ADRON] in line 01 of the preceding routine, [OUTACL] will send only these messages:

DAB 65	"A"
DAB 66	"B"
DAB 67	"C"
DAB 68	"D"
DAB 69	"E"
DAB 13	Carriage return
DAB 10	Line feed

Only the data bytes from the ALPHA register are transmitted.

When Addressing-Off mode is active, you (or your program) will have to specify the loop and device addressing for data transfer that is performed automatically in Addressing-On mode. The [SEND] function, which can transmit any HP-IL command message, provides this capability, with the exception of auto-addressing. The Auto-Address (AAD) message is a "Ready class" message, which [SEND] cannot transmit. However, execution of any of the control functions other than the data transfer functions assigns HP-IL addresses to the loop devices. Therefore, before your program attempts operation in Addressing-Off mode, it should execute a control function to carry out loop addressing. [NLOOP] is a good choice for this purpose, since it sends only the Auto-Address Unconfigure (AAU) and Auto-Address (AAD) messages.

The following is the general procedure for data transfers in Address-Off mode:

1. If there is a printer on the loop:

   - For HP 82162A printers, set mode switch to MAN or NORM.

   - For other printers, clear flags 15 & 16.

2. Be sure that a control function has been executed to assign loop addresses at least once since the loop was connected. Data transfer functions may be used for this purpose only if they are executed in Addressing-On mode.

3. Execute [ADROFF].

4. Use [SEND], or [TAD] and/or [LAD] to enable the desired devices as talkers and/or listeners. (These three functions are described later in this section, on pages 61 through 65.)

5. Execute one or more data transfer functions.

6. Use [SEND], or [UNL] and/or [UNT] to return addressed talker or listener devices to their idle states.

7. After you complete all data transfers, it is recommended that you restore the Addressing On mode by executing [ADRON].

Sending HP-IL Command Messages

The remaining functions in this section enable the HP-41 to send HP-IL command messages individually. They are necessary for use with data transfer functions executed in Addressing-Off mode. [SEND] can transmit any of the 256 possible command messages. The remaining six functions are special cases of [SEND] that are provided to save program bytes and enhance program legibility.

[SEND] (send command) HP-IL: CMD

Input : X | command number |

Output: None

[SEND] transmits the HP-IL command message specified in the X-register, followed by a Ready-for-Command (RFC) message. The command number (from 0 to 255) selects one of 256 possible command messages, as listed in the following table:

HP-IL Command Message Table

+	0	1	2	3	4	5	6	7
0	NUL	GTL	-	-	SDC	PPD	-	-
8	GET	-	-	-	-	-	-	ELN
16	NOP	LLO	-	-	DCL	PPU	-	-
24	EAR	-	-	-	-	-	-	-
32	LAD0	LAD1	LAD2	LAD3	LAD4	LAD5	LAD6	LAD7
40	LAD8	LAD9	LAD10	LAD11	LAD12	LAD13	LAD14	LAD15
48	LAD16	LAD17	LAD18	LAD19	LAD20	LAD21	LAD22	LAD23
56	LAD24	LAD25	LAD26	LAD27	LAD28	LAD29	LAD30	UNL
64	TAD0	TAD1	TAD2	TAD3	TAD4	TAD5	TAD6	TAD7
72	TAD8	TAD9	TAD10	TAD11	TAD12	TAD13	TAD14	TAD15
80	TAD16	TAD17	TAD18	TAD19	TAD20	TAD21	TAD22	TAD23
88	TAD24	TAD25	TAD26	TAD27	TAD28	TAD29	TAD30	UNT
96	SAD0	SAD1	SAD2	SAD3	SAD4	SAD5	SAD6	SAD7
104	SAD8	SAD9	SAD10	SAD11	SAD12	SAD13	SAD14	SAD15
112	SAD16	SAD17	SAD18	SAD19	SAD20	SAD21	SAD22	SAD23
120	SAD24	SAD25	SAD26	SAD27	SAD28	SAD29	SAD30	-
128	PPE0	PPE1	PPE2	PPE3	PPE4	PPE5	PPE6	PPE7
136	PPE8	PPE9	PPE10	PPE11	PPE12	PPE13	PPE14	PPE15
144	IFC	-	REN	NRE	-	-	-	-
152	-	-	AAU	LPD	-	-	-	-
160	DDL0	DDL1	DDL2	DLL3	DDL4	DDL5	DDL6	DDL7
168	DDL8	DDL9	DDL10	DDL11	DDL12	DDL13	DDL14	DDL15

HP-IL Command Message Table (Continued)

+	0	1	2	3	4	5	6	7
176	DDL16	DDL17	DDL18	DDL19	DDL20	DDL21	DDL22	DDL23
184	DDL24	DDL25	DDL26	DDL27	DDL28	DDL29	DDL30	DDL31
192	DDT0	DDT1	DDT2	DDT3	DDT4	DDT5	DDT6	DDT7
200	DDT8	DDT9	DDT10	DDT11	DDT12	DDT13	DDT14	DDT15
208	DDT16	DDT17	DDT18	DDT19	DDT20	DDT21	DDT22	DDT23
216	DDT24	DDT25	DDT26	DDT27	DDT28	DDT29	DDT30	DDT31
224	-	-	-	-	-	-	-	-
232	-	-	-	-	-	-	-	-
240	-	-	-	-	-	-	-	-
248	-	-	-	-	-	-	-	-

How to Use the Command Table. To determine the command number corresponding to a specific command message, find the row and column containing the abbreviation of the desired command. The command number is the sum of the number at the left of the row and the number at the top of the column.

Abbreviation	Message
AAU	Auto Address Unconfigure
DCL	Device Clear
DDL	Device Dependent Listener Command
DDT	Device Dependent Talker Command
ELN	Enable Listener Not Ready for Data
EAR	Enable Asyncronous Request
GET	Group Execute Trigger
GTL	Go To Local
IFC	Interface Clear
LAD	Listener Address
LLO	Local Lock Out
LPD	Loop Power Down
NOP	No Operation
NRE	Not Remote Enable
NUL	Null
PPD	Parallel Poll Disable
PPE	Parallel Poll Enable
PPU	Parallel Poll Unconfigure
REN	Remote Enable
SAD	Secondary Address
SDC	Selected Device Clear
TAD	Talker Address
UNL	Unlisten
UNT	Untalk
-	Undefined

Example Using [SEND]. The following program causes the device at address 4 to be placed into Local mode.

```
01 36
02 SEND                  Listener Address 4 (LAD4).
03 1
04 SEND                  Go To Local (GTL).
```

This routine differs from the sequence

```
01 4
02 SELECT
03 LOCAL
```

in that the [LOCAL] function auto-addresses the loop prior to execution and sends the Unlisten command message afterwards.

[LAD] (Listener Address) HP-IL: Listener Address

Input : X | address |

Output: None

[TAD] (Talker Address) HP-IL: Talker Address

Input : X | address |

Output: None

[LAD] transmits a Listener Address message (followed by RFC) to make the device at the address specified in the X-register a listener. The keystroke sequences 1 [LAD], 2 [LAD], ..., 30 [LAD] are equivalent to 33 [SEND], 34 [SEND], ..., 62 [SEND], respectively.

[TAD] operates in the same manner as [LAD], sending a Talker Address message. The device at the address specified in the X-register is made a talker. The sequences 1 [TAD], 2 [TAD], ..., 30 [TAD] are equivalent to 65 [SEND], 66 [SEND], ..., 94 [SEND], respectively.

The number in the X-register prior to executing [LAD] or [TAD] must have a value between 1 and 30 (ignoring the sign and fractional part). If it is not in this range, the HP-41 displays the ADR ERR (Address Error) message.

[DDL] (Device-Dependent Listener) — HP-IL: Device-Dependent Listener (DDL)

Input : X | command number |

Output: None

[DDT] (Device-Dependent Talker) — HP-IL: Device-Dependent Talker (DDT)

Input : X | command number |

Output: None

[DDL] sends on HP-IL the Device-Dependent Listener command (from 0 to 31) message specified in the X-register. The message is received by all active listeners.

[DDT] sends the Device-Dependent Talker command (from 0 to 31) specified in the X-register. Only the current active talker responds to the message.

The DEVT* routine listed below illustrates the use of advanced control functions to perform loop operations that are carried out automatically by other control functions. DEVT* sends a Device-Dependent Talker command (specified in the X-register) to the current primary device. "DEVT*" is a near equivalent to the function [DEVT], with the important difference that the auto-addressing and final unlisten performed by [DEVT] are not carried out by DEVT*. (Before you execute DEVT*, you must execute an auto-addressing function such as [NLOOP] to assign addresses to the devices on the loop.)

01 LBL "DEVT*"		Assumes a DDT number dd in the X-register.
02 RCLSEL		Finds the primary address nn.
03 TAD	TADnn, RFC	Makes the primary device a talker.
04 X<>Y		Returns dd to the X-register.
05 DDT	DDTdd, RFC	Sends DDT.

In contrast to the four HP-IL messages shown in the preceding listing, the [DEVT] function sends the following sequence of ten messages:

```
IDY
AAU
RFC
AAD
TADnn
RFC
DDTdd
RFC
UNT
RFC
```

[UNL] (Unlisten)	HP-IL: Unlisten (UNL)
Input : None	
Output: None	

[UNL] sends the Unlisten message on HP-IL, removing all currently addressed listeners from listener status. [UNL] is equivalent to the sequence 63 [SEND].

[UNT] (Untalk)	HP-IL: Unlisten (UNL)
Input : None	
Output: None	

[UNT] removes the current talker on HP-IL from talker status by sending the Untalk message (followed by RFC). [UNT] is equivalent to the sequence 95 [SEND].

The following program illustrates the use of advanced control functions to control devices in Addressing-Off mode. The program causes an HP 82165A HP-IL/GPIO Interface to send the contents of its 19 control registers to the HP-41 ALPHA register (compare this routine with the incorrect version on page 57).

```
01*LBL "GPIOREG"
02 64                      Interface accessory ID.
03 FINDAID                 Returns interface address (and
                           assigns loop addresses).
04 ADROFF                  Sets Addressing-Off mode.
05 TAD                     Makes interface a talker.
```

06 0	DDT 0 instructs interface to send its control registers.
07 DDT	Sends device-dependent Talker message.
08 19 09 INAN	} Inputs 19 bytes to ALPHA.
10 UNT	Removes interface from Talker status.
11 END	

Error Messages

Display	Functions	Meaning
ADR ERR	[COPYFL] [LAD] [TAD] [XFER...]	Specified address is less than 1 or greater than 30.
ALPHA DATA	-all-	Alpha characters are in a register where a number is expected.
DATA ERROR	-all-	Data is out of range or of an inappropriate type.
DEV nn ERR	[MCOPY] [MCOPYPV] [MVERIFY]	The mass storage device at nn address has an error. Its medium may not be valid.
DIR FULL	[COPYFL]	A medium's directory has no room for further file entries.
DUP FL NAME	[COPYFL]	The named file already exists on the medium.
FL NOT FOUND	[COPYFL] [FLLENG] [FLTYPE]	The specified file does not exist on the medium.
MEDM ERR	[COPYFL] [DIRX] [FLLENG] [FLTYPE] [MCOPY] [MCOPYPV] [MVERIFY]	There is an error on the medium.
MEDM FULL	[COPYFL]	The medium is full.

NAME ERR	[OUTP]	The global label specified in the ALPHA register doesn't exist.
NO DRIVE	[COPYFL] [DIRX] [FLLENG] [FLTYPE] [MCOPY] [MCOPYPV] [MVERIFY]	There is no standard mass storage device on the loop.
NO HPIL	-all-	The HP 82160A HP-IL Module is not plugged into the HP-41.
NO RESPONSE	[AID] [ID] [STAT] [IN...] [XFER...]	The primary device does not respond to a particular HP-IL message.
NO ROOM	[INP]	Available HP-41 memory is too small to store the program.
PACKING TRY AGAIN	[INP]	There is no room to finish storing the program. Memory is packed.
PRIVATE	[COPYFL] [MCOPY] [MCOPYPV] [OUTP]	A file to be copied is private.
ROM	[OUTP]	Specified program is in ROM (read-only memory).
TRANSMIT ERR	all loop functions	The HP-41 has received an invalid HP-IL message or no message at all.

Appendix B

Care, Warranty, and Service Information

Contents

Module Care

CAUTION

Always turn off the HP-41 before connecting or disconnecting any module or peripheral. Failure to do so could result in damage to the HP-41 or disruption of the system's operation.

- Keep the contact area of the module free of obstructions. Should the contacts become dirty, carefully brush or blow the dirt out of the contact area. Do not use any liquid to clean the contacts.

- Store the module in a clean, dry place.

- Always turn off the computer before installing or removing any module or peripheral.

- Observe the following temperature specifications:

  Operating: 0° C to 45° C (32° F to 113° F)

  Storage: -40° C to 75° C (-40° F to 167° F)

Limited One-Year Warranty

What We Will Do

The HP 82183A Extended I/O Module 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 limitations 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:

- In the United States:

Hewlett-Packard
Corvallis Division
1000 N.E. Circle Blvd.
Corvallis, OR 97330
Telephone: (503) 758-1010
Toll-Free Number: (800) 547-3400 (except in
Oregon, Hawaii, and Alaska)

- In Europe:

Hewlett-Packard S.A.
7, rue du Bois-du-lan
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, California 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 upon the time of year and the work load at the service center. The total time you are without your 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
Corvallis Division Service Department
P.O. Box 999/1000 N.E. Circle Blvd.
Corvallis, Oregon 97339/97330, 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 GmBH
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 listed under Austria.

FINLAND

HEWLETT-PACKARD OY
Revontulentie 7
SF-02100 ESPOO 10 (Helsinki)
Telephone: (90) 455 02 11

FRANCE

HEWLETT-PACKARD FRANCE
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.
Cassella postale 3645 (Milano)
Via G. DiVittorio, 9
I-20063 CERNUSCO SUL NAVIGLIO (Milan)
Telephone: (2) 90 36 91

NETHERLANDS

HEWLETT-PACKARD NEDERLAND B.V.
Van Heuven Goedhartlaan 121
N-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 20 00

SWITZERLAND

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

UNITED KINGDOM

HEWLET-PACKARD LTD
King Street Lane
GB-Winnersh, Wokingham
BERKSHIRE RG11 5AR
Telephone: (734) 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.

Technical Assistance

The keystroke procedures and program material in this manual are supplied with the assumption that the user has a working knowledge of the concepts and terminology used. Hewlett-Packard's technical support is limited to an explanation of operating procedures used in the manual and verification of answers given in the examples. If you have technical problems involving this manual, consult your HP-41 owner's manual. Should you require further assistance, write to:

Hewlett-Packard
Corvallis Division Customer Support
1000 N.E. Circle Blvd.
Corvallis, OR 97330

Dealer and Product Information

For U.S.A, dealer locations, product information, and prices, please call (800) 547-3400. In Oregon, Alaska, or Hawaii, call (503) 758-1010.

Examples

Contents

Introduction

The program examples listed in this appendix demonstrate the use of many I/O module functions. Even if you have no direct application for the programs, you may wish to study the listings for programming suggestions that you can use in your own HP-IL applications.

Printing an HP-IL Device Directory (The LCAT Program)

The LCAT (Loop Catalog) program listed below illustrates the use of loop configuration functions. LCAT prints a directory of the current HP-IL configuration, showing the number of devices and the primary address, and listing for each device on the loop, the device ID, accessory ID, and device class. The directory is printed on the first printer or display found on the loop, starting with the primary device. Notice that LCAT uses data storage registers R00 and R01.

	01♦LBL "LCAT"
	02 AUTOIO
	03 CF 29
	04 FIX 0
Finds number of devices on HP-IL.	05 NLOOP
Prints number of devices.	06 CLA 07 ARCL X 08 "⊦ DEVICES"
	09 PRA
	10 1 E3
	11 /
	12 1
	13 +
Stores a loop counter.	14 STO 00
Returns primary address.	15 RCLSEL
	16 STO 01

Prints primary address.	17 CLA 18 "PRIMARY ADR. " 19 ARCL X
	20 PRA
Prints directory heads.	21 "ADR ID AID" 22 "⊦ CLASS " 23 PRA
	24♦LBL 01 25 CLA 26 10
Recalls loop counter.	27 RCL 00
Ensures that address is two digits.	28 X<Y? 29 "0"
Accumulates address into print buffer.	30 ARCL X 31 "⊦ " 32 ACA
Selects device having **address** equal to **loop counter.**	33 SELECT
	34 SF 25
Returns device ID.	35 ID
Substitutes dashes if device sends no device ID.	36 FC?C 25 37 "--------"
	38 8
Finds length of ID string.	39 ALENGIO
	40 - 41 X=0? 42 GTO 04
Fills out device ID string to eight characters.	43♦LBL 03 44 "⊦ " 45 DSE X
	46 GTO 03 47♦LBL 04 48 "⊦ " 49 100 50 SF 25 51 AID
Reselects original primary device.	52 RCL 01 53 SELECT
	54 RDN
Branches if device did not send accessory ID.	55 FC?C 25 56 GTO 05
Appends a space if accessory ID < 100.	57 X<Y? 58 "⊦ "
	59 ARCL X
Adds ALPHA register string to print buffer.	60 ACA
Determine device class...	61 16 62 / 63 INT 64 16 65 *
Places device class string into ALPHA.	66 XEQ IND X
	67♦LBL 05 68 ACA
Prints printer buffer.	69 PRBUF

Increments loop counter.
Branches for additional devices.

Device class Alpha strings...

```
70 ISG 00
71 GTO 01
72 RTN
73♦LBL 00
74 " CNTRLLR"
75 RTN
76♦LBL 16
77 " MASS ST"
78 RTN
79♦LBL 32
80 " PRINTER"
81 RTN
82♦LBL 48
83 " DISPLAY"
84 RTN
85♦LBL 64
86 " INTRFCE"
87 RTN
88♦LBL 80
89 " INSTRMT"
90 RTN
91♦LBL 96
92 " GRAPHIC"
93 END
```

HP-41 Program Transfer Via Acoustic Couplers (Modems)

HP-IL interfaces allow you to use data transfer functions to exchange data between HP-IL devices (including the HP-41) and devices external to the loop. Before the data transfer functions can be used with an interface, however, you must ensure that the interface is properly configured, and also that the devices connected to the non HP-IL side of the interface are prepared to send or receive data. The PROUT, PRIN, ANS, and ORIG programs listed on the following pages are examples of using an interface with the HP-41. Specifically, the programs control the transfer of HP-41 programs between two HP-41s via two HP 82168A Acoustic Couplers.

ORIG (Originate Mode) and ANS (Answer Mode) are general purpose subroutines designed for use with the couplers. With telephone receivers connected to both couplers, one of the two HP-41s executes ORIG while the other HP-41 executes ANS. When execution of both routines is completed, the two couplers will be configured so that data transfer between the two HP-41s can proceed. The routines are designed so that no particular timing is required of the two users--either program can be started first, and at any time prior to the other program.

PROUT and PRIN use ORIG and ANS specifically for program transfer. Prior to program execution, the two users contact each other by telephone, then place their receivers into their couplers. The user sending a program places the name of a global label (up to six characters) from the program in the ALPHA register, then executes PROUT. Meanwhile, the other user executes PRIN (which should not be the last program in memory).

Messages are displayed during program execution to show the ongoing "conversation" between the HP-41s. The sending HP-41 displays CALL... when its coupler is sending the answer mode carrier and looking for the Originate carrier. The receiving HP-41 displays ANSWER while its coupler is looking for the incoming carrier, to which it responds with the Originate carrier. When the carriers are established, the sending HP-41 displays TALK... while the receiving HP-41 displays REPLY... Next, the sending HP-41 displays SEND: followed by the name of the program being sent; the other HP-41 displays PRGM IN... When the transfer is complete, DONE appears in both HP-41 displays, and the receivers may be removed from the couplers.

The PRIN Program

PRIN is designed for execution after you place a telephone receiver in the coupler.

	01♦LBL "PRIN"
	02 XEQ "ORIG"
Shows that program transfer is active.	03 "PRGM IN..." 04 AVIEW
Reads in program.	05 INP
Sends DONE to other coupler.	06 "DONE" 07 OUTACL
Pauses to ensure that DONE is sent.	08 PSE
Resets coupler.	09 CLRDEV
Announces program completion.	10 AVIEW
	11 END

The PROUT Program

This program requires you to specify a global label in ALPHA and uses data storage register R00.

	01♦LBL "PROUT"
Saves program name.	02 ASTO 01
	03 XEQ "ANS"
Shows that program transfer is active.	04 "SEND: " 05 ARCL 01 06 AVIEW
	07 ASHF
Sends program.	08 OUTP
Reads DONE message from other coupler.	09 INACL
Resets coupler.	10 CLRDEV
Announces program completion.	11 AVIEW
	12 END

The ORIG Program

This program is the originate mode handshake subroutine.

Powers loop.	01♦LBL "ORIG" 02 PWRUP
Makes coupler the primary device.	03 65 04 FINDAID 05 SELECT
Clears input and output buffers.	06 "R" 07 XEQ 01
Puts coupler in Originate mode.	08 "O" 09 XEQ 01
Selects XON/XOFF handshake.	10 "C2" 11 XEQ 01
Display indicates originating coupler is waiting for answering coupler's carrier.	12 "ANSWER..." 13 AVIEW
Reads coupler status.	14♦LBL 02 15 STAT
Tests Carrier Received status bit.	16 ATOXR 17 X<>FIO 18 FC? 00
Branches if no carrier detected.	19 GTO 02
Clears buffers.	20 "R" 21 XEQ 01
Display indicates that carrier is received--now awaiting data.	22 "REPLY..." 23 AVIEW
	24♦LBL 03
Tests Data Available status bit.	25 STAT 26 ATOXR 27 X<>FIO 28 FC? 01
Branches if no data available yet.	29 GTO 03
Sends message to acknowledge reception.	30 OUTAE
Reads all data from input buffer up to character 1.	31 1 32♦LBL 04 33 INAC
Tests whether character 1 has been received.	34 FS? 17
Branches to continue reading.	35 GTO 04
Coupler is now configured to send or to receive.	36 RTN
Subroutine to send remote commands to coupler.	37♦LBL 01
Places dummy character at start of string.	38 1 39 XTOAL
Puts coupler in Remote mode.	40 REMOTE
Sends command string.	41 OUTACL
Returns coupler to Local mode.	42 NOTREM
	43 END

The ANS Program

This program is the answer mode handshake subroutine.

Powers loop.	01♦LBL "ANS" 02 PWRUP
Makes coupler the primary device.	03 65 04 FINDAID 05 SELECT
clears input and output buffers.	06 "R" 07 XEQ 01
Puts coupler in Answer mode.	08 "A" 09 XEQ 01
Selects XON/XOFF handshake mode.	10 "C2" 11 XEQ 01
Display indicates answering coupler is waiting for originating coupler's carrier.	12 "CALL..." 13 AVIEW
Reads coupler status.	14♦LBL 02 15 STAT
Tests Carrier Available status bit.	16 ATOXR 17 X<>FIO 18 FC? 00
Branches if no carrier detected.	19 GTO 02
Display indicates that carrier is received--now sending data and checking for acknowledgment.	20 "TALK..." 21 AVIEW
	22♦LBL 03
Clears buffers.	23 "R" 24 XEQ 01
	25♦LBL 04
Sets Error Ignore flag.	26 SF 25
Sends data to start "conversation."	27 OUTAE
Checks for a transmission error.	28 FC?C 25
Branches to clear buffers.	29 GTO 03
Checks Data Available status bit for acknowledgment data.	30 STAT 31 ATOXR 32 X<>FIO 33 FC? 01
Branches if no acknowledgment received	34 GTO 04
Sends character 1 to signal end of setup transmission.	35 1 36 OUTAC
Clears acknowledgment byte.	37 INXB
Coupler is now configured to send or receive.	38 RTN
Subroutine to send Remote mode commands.	39♦LBL 01 40 1 41 XTOAL 42 REMOTE 43 OUTACL 44 NOTREM 45 END

Data Transfer in Addressing-Off Mode

The next two programs illustrate the use of data transfer functions in Addressing-Off mode to transfer long data strings. The programs are designed to allow the HP-41 to read and write the 291-byte configuration string of an HP 1610B Logic State Analyzer (connected to HP-IL through the HP 82169A HP-IL/HP-IB Interface).

Both programs assume that the analyzer is the primary device.

The GETCNF Program

This program saves the analyzer configuration. GETCNF uses data registers R00 through R58.

	01♦LBL "GETCNF"
Puts analyzer in Remote mode.	02 REMOTE
LR1 instructs analyzer to send configuration. (The "?" is a command terminator.)	03 "DLR1?" 04 0 05 OUTAN
Sets Addressing-Off mode.	06 ADROFF
Makes analyzer a talker.	07 RCLSEL 08 TAD
Loop counter.	09 .057 10 5
	11♦LBL 01
Inputs five bytes.	12 INAN
Saves ALPHA register string.	13 ASTO IND Y
Increments loop counter.	14 ISG Y
Branches to input more data.	15 GTO 01
Inputs 291st byte.	16 1 17 INAN
	18 ASTO IND Z
Clears analyzer talker status.	19 UNT
Restores Addressing-On mode.	20 ADRON
Puts devices into Local mode.	21 NOTREM
	22 END

The WRTCNF Program

This program assumes that the analyzer configuration stored by the preceding GETCNF program is stored in data registers R00 through R58.

	01♦LBL "WRTCNF"
	02 REMOTE
	03 ADROFF
Makes analyzer a listener.	04 RCLSEL 05 LAD
Loop counter.	06 .058 07 5

	08♦LBL 02
	09 CLA
Recalls next register.	10 ARCL IND Y
Sends a five-byte string to analyzer.	11 OUTAN
Increments loop counter.	12 ISG Y
Branches for more data.	13 GTO 02
Clears listener status.	14 UNL
	15 ADRON
	16 NOTREM
	17 END

Bar Code

Contents

MSCOPY

PROGRAM REGISTERS NEEDED: 22

ROW 1 (1 : 2)

ROW 2 (2 : 6)

ROW 3 (6 : 9)

ROW 4 (9 : 12)

ROW 5 (12 : 15)

ROW 6 (15 : 18)

ROW 7 (18 : 23)

ROW 8 (24 : 31)

ROW 9 (32 : 33)

ROW 10 (33 : 41)

ROW 11 (42 : 45)

ROW 12 (45 : 46)

LCAT

PROGRAM REGISTERS NEEDED: 38

ROW 1 (1 : 4)

ROW 2 (4 : 8)

ROW 3 (8 : 15)

ROW 4 (15 : 18)

ROW 5 (18 : 21)

ROW 6 (21 : 22)

ROW 7 (22 : 26)

ROW 8 (26 : 33)

ROW 9 (33 : 37)

ROW 10 (37 : 44)

ROW 11 (44 : 50)

ROW 12 (50 : 58)

ROW 13 (58 : 65)

ROW 14 (66 : 73)

ROW 15 (74 : 77)

ROW 16 (77 : 80)

ROW 17 (80 : 83)

ROW 18 (83 : 86)

ROW 19 (86 : 89)

ROW 20 (89 : 92)

ROW 21 (92 : 93)

PRIN

PROGRAM REGISTERS NEEDED: 6

ROW 1 (1 : 2)

ROW 2 (2 : 4)

ROW 3 (5 : 10)

ROW 4 (11 : 11)

PROUT

PROGRAM REGISTERS NEEDED: 6

ROW 1 (1 : 3)

ROW 2 (3 : 6)

ROW 3 (7 : 12)

ORIG

PROGRAM REGISTERS NEEDED: 15

ROW 1 (1 : 4)

ROW 2 (4 : 9)

ROW 3 (10 : 12)

ROW 4 (12 : 18)

ROW 5 (19 : 22)

ROW 6 (22 : 28)

ROW 7 (29 : 36)

ROW 8 (37 : 43)

ANS

PROGRAM REGISTERS NEEDED: 15

ROW 1 (1 : 4)

ROW 2 (5 : 10)

ROW 3 (10 : 12)

ROW 4 (13 : 20)

ROW 5 (20 : 24)

ROW 6 (24 : 31)

ROW 7 (31 : 38)

ROW 8 (39 : 45)

GETCNF

PROGRAM REGISTERS NEEDED: 8

ROW 1 (1 : 3)

ROW 2 (3 : 8)

ROW 3 (8 : 14)

ROW 4 (14 : 21)

ROW 5 (21 : 22)

WRTCNF

PROGRAM REGISTERS NEEDED: 7

ROW 1 (1 : 3)

ROW 2 (3 : 9)

ROW 3 (10 : 16)

ROW 4 (16 : 17)

Function Index and XROM Listing

Function Index

Programmable Function XROM Numbers

The HP 82183A Extended I/O Module's functions can be entered in a program whenever the module is plugged into the HP-41. While the I/O module is plugged in, program lines containing I/O module functions are displayed and printed using the standard function names. If you later disconnect the module, these program lines are displayed and printed as XROM functions--with two identification numbers. These numbers indicate that the function belongs to a plug-in accessory. The first number identifies the accessory. (XROM accessory number 23 corresponds to the HP 82183A Extended I/O module.) The second number identifies the function for that accessory. When you remove the I/O module, its functions have the following XROM numbers.

Function	XROM Number	Function	XROM Number	Function	XROM Number
[COPYFL]	XROM 23,01	[ATOXX]	XROM 23,13	[ID]	XROM 23,25
[DIRX]	XROM 23,02	[XTOAL]	XROM 23,14	[INAC]	XROM 23,26
[FLLENG]	XROM 23,03	[XTOAR]	XROM 23,15	[INACL]	XROM 23,27
[FLTYPE]	XROM 23,04	[X<>FIO]	XROM 23,16	[INAE]	XROM 23,28
[MCOPY]	XROM 23,05	[YTOAX]	XROM 23,17	[INAN]	XROM 23,29
[MCOPYPV]	XROM 23,06	[AID]	XROM 23,19	[INXB]	XROM 23,30
[MVERIFY]	XROM 23,07	[CLRDEV]	XROM 23,20	[INP]	XROM 23,31
[ALENGIO]	XROM 23,09	[CLRLOOP]	XROM 23,21	[LOCK]	XROM 23,32
[ANUNDEL]	XROM 23,10	[DEVL]	XROM 23,22	[NLOOP]	XROM 23,33
[ATOXL]	XROM 23,11	[DEVT]	XROM 23,23	[NOTREM]	XROM 23,34
[ATOXR]	XROM 23,12	[FINDAID]	XROM 23,24	[OUTAC]	XROM 23,35

Function	XROM Number	Function	XROM Number	Function	XROM Number
[OUTACL]	XROM 23,36	[RCLSEL]	XROM 23,45	[ADRON]	XROM 23,55
[OUTAE]	XROM 23,37	[SRQ?]	XROM 23,46	[DDL]	XROM 23,56
[OUTAN]	XROM 23,38	[STAT]	XROM 23,47	[DDT]	XROM 23,57
[OUTXB]	XROM 23,39	[XFER]	XROM 23,48	[LAD]	XROM 23,58
[OUTP]	XROM 23,40	[XFERC]	XROM 23,49	[SEND]	XROM 23,59
[POLL]	XROM 23,41	[XFERCL]	XROM 23,50	[TAD]	XROM 23,60
[POLLD]	XROM 23,42	[XFERE]	XROM 23,51	[UNL]	XROM 23,61
[POLLE]	XROM 23,43	[XFERN]	XROM 23,52	[UNT]	XROM 23,62
[POLLUNC]	XROM 23,44	[ADROFF]	XROM 23,54		

Subject Index

Underlined page numbers indicate primary references. In addition to the references in this subject index, an index to the extended I/O module's functions is located on pages 88 through 89.

E

F

G-H

I

HEWLETT PACKARD

Corvallis Division
1000 N.E. Circle Blvd., Corvallis, OR 97330, U.S.A.

82183-90001

Printed in U.S.A. 12/82