NOTICE

Hewlett-Packard Company makes no express or implied warranty with regard to the program material offered or the merchantability or the fitness of the program material for any particular purpose. The program material is made available solely on an "as is" basis, and the entire risk as to its quality and performance is with the user. Should the program material prove defective, the user (and not Hewlett-Packard Company nor any other party) shall bear the entire cost of all necessary correction and all incidental or consequential damages. Hewlett-Packard Company shall not be liable for any incidental or consequential damages in connection with or arising out of the furnishing, use, or performance of the program material.
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

Section 1: Getting Started .......................................................... 6  
Contents--Introduction--Installing and Removing the I/O Module-- 
Extended I/O Module Configuration and Characteristics--Using 
This Manual--I/O Module Functions and the HP-41 System--Bar Code 
for Program Examples

Section 2: Mass Storage Operations .............................................. 10 
Contents--Introduction--Copying Files--Obtaining Directory 
Information

Section 3: Character Manipulation Functions ............................... 19 
Contents--Introduction--The ALPHA Register As an I/O Buffer-- 
ALPHA-X Transfer--Interpreting Alpha Strings

Section 4: HP-IL Control Functions .............................................. 30 
Contents--Introduction--Loop Configuration Functions--Device 
Control Functions--Data Transfer Functions

Section 5: Advanced Control Functions ....................................... 56 
Contents--Introduction--I/O Module HP-IL Device Addressing 
Modes--Sending HP-IL Command Messages

Appendix A: Error Messages ...................................................... 67
Appendix B: Care, Warranty, and Service Information ...................... 69
Appendix C: Examples ............................................................... 75
Appendix D: Bar Code ............................................................... 83
Function Index and XROM Listing ............................................... 88
Subject Index ................................................................. 91
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.
Section 2

Mass Storage Operations

Contents

Introduction .................................................. 10
Copying Files .................................................. 10
  Copying Individual Files ([COPYFL]) ..................... 11
  Duplicating an Entire Medium ([MCOPY], [MCOPYPV]) .... 11
  Verifying Media ([MVERIFY]) ............................. 14
Obtaining Directory Information ([DIRX], [FLTYPE],
  [FLENG]) ..................................................... 14

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. A minimum of two mass storage devices must be present in the HP-IL system for these three functions to operate. 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 inadvertently 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 Specifies address of duplicate.
03 5 Specifies name of file to be copied.
04 "MASTER" Copies file MASTER from device at address 1 to device at address 5.
05 COPYFL

Duplicating an Entire Medium

[MCOPY] (Mass Copy)

[MCOPYPV] (Mass Copy-Private)

Input: -None-

Output: number of records copied
Section 2: Mass Storage Operations

[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.)
Section 2: Mass Storage Operations

01 LBL MSCOPY
02 1.01
03 SELECT
04 AUTOIO
05 CF 21
06 "MASTER IN"
07 AVIEW
08 PSE
09 "DRIVE WITH"
10 AVIEW
11 PSE
12 "FLASHING"
13 AVIEW
14 PSE
15 "BUSY LIGHT?"
16 AVIEW
17 PSE
18 "IF NOT PRESS"
19 AVIEW
20 PSE
21 "R/S"
22 AVIEW
23 PSE
24 LBL 01
25 RCLSEL
26 LISTEN
27 31
28 LISTEN
29 RDN
30 RDN
31 ISG X
32 GTO 01
33 "LAST CHANCE"
34 AVIEW
35 PSE
36 PSE
37 PSE
38 FS? 55
39 SF 21
40 MCOPY
41 MVERIFY
42 "DONE"
43 BEEP
44 PROMPT
45 GTO "MSCOPY"

Selects primary device.

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.

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

Pauses program before commencing copying operation.

Restores printing output.

Executes copying operation (unless master tape contains any "private" files). Verifies that contents of master medium were correctly copied onto duplicate media.

Prompts you when medium copying and verifying is completed. If you press [R/S], program re-executes.
Section 2: Mass Storage Operations

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 \( x = 0 \) or \( x > 999 \) 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:
Section 2: Mass Storage Operations

<table>
<thead>
<tr>
<th>[DIR]</th>
<th>Abbreviation</th>
<th>File Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR</td>
<td>Program</td>
<td></td>
</tr>
<tr>
<td>DA</td>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>KE</td>
<td>Key Assignment</td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>WA</td>
<td>&quot;Write-All&quot;</td>
<td></td>
</tr>
<tr>
<td>AS</td>
<td>ASCII</td>
<td></td>
</tr>
<tr>
<td>??</td>
<td>Unknown</td>
<td></td>
</tr>
</tbody>
</table>

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.

01 LBL "DACOPY"          \ Selects the master device.
02 1                      \}
03 SELECT                 \}
04 1,447                  \} Stores a loop counter for the maximum
05 STO 00                 \} directory size.
06 "DA"                   \} Stores the Alpha string that specifies
07 ASTO 01                \} "data" file type.
08 LBL 00                 \} Begins loop that checks for file type.
09 RCL 00                 \} Recalls and uses loop counter to obtain
10 DIRX                   \} file name.
11 X=0?                   \} Tests for end of directory. If file number
12 RTN                    \} is zero, all files have been checked and
13 RCL 01                 \} program halts.
14 FLTYPE                 \} Recalls DA to X.
15 X#Y?                   \} Obtains file type in X and lifts DA to Y.
16 GTO 01                 \} Tests "not a data file?". If file type in
17 2                      \} X is not DA, returns execution to [LBL] 01
18 COPYFL                 \} to begin check of next file.
19 LBL 01                 \} If file type in X is DA, specifies duplicate
20 ISG 00                 \} device address and copies file named by
21 GTO 00                 \} string placed in ALPHA by preceding
22 END                     \} execution of [DIRX].

Increments file counter to next file listed in master directory.
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.

[FILLENG] (File Length)

Input: ALPHA file name

Output: X file length

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

<table>
<thead>
<tr>
<th>File Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR</td>
<td>Number of Bytes in the Program</td>
</tr>
<tr>
<td>DA</td>
<td>Number of Registers</td>
</tr>
<tr>
<td>KE</td>
<td>Undefined</td>
</tr>
<tr>
<td>ST</td>
<td>Unknown</td>
</tr>
<tr>
<td>WA</td>
<td></td>
</tr>
<tr>
<td>AS</td>
<td></td>
</tr>
<tr>
<td>??</td>
<td></td>
</tr>
</tbody>
</table>

If the named file does not exist on the medium when you execute [FILLENG], 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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Regs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>PR</td>
<td>32</td>
</tr>
<tr>
<td>FLY</td>
<td>PR</td>
<td>70</td>
</tr>
<tr>
<td>BLKBOX</td>
<td>WA</td>
<td>336</td>
</tr>
<tr>
<td>FBDATA</td>
<td>DA</td>
<td>32</td>
</tr>
<tr>
<td>NNN</td>
<td>KE</td>
<td>13</td>
</tr>
<tr>
<td>DECODE</td>
<td>PR</td>
<td>23</td>
</tr>
<tr>
<td>KOUT</td>
<td>AS</td>
<td>42</td>
</tr>
</tbody>
</table>

Use the following keystrokes to obtain directory information.
## Section 2: Mass Storage Operations

<table>
<thead>
<tr>
<th>Keystrokes</th>
<th>Display</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 [DIRX]</td>
<td>1.0000</td>
<td>Obtains name of first file listed in directory.</td>
</tr>
<tr>
<td>[ALPHA]</td>
<td>CODE</td>
<td>Displays name of first file, which is a program file. (Refer to preceding listing.)</td>
</tr>
<tr>
<td>[ALPHA]</td>
<td>1.0000</td>
<td>Displays number of bytes in CODE program. (Because HP-41 uses 7 bytes per register, the CODE program uses 31 registers and 4 bytes.)</td>
</tr>
<tr>
<td>[FLLENG]</td>
<td>221.0000</td>
<td>Displays number of bytes in CODE program. (Because HP-41 uses 7 bytes per register, the CODE program uses 31 registers and 4 bytes.)</td>
</tr>
<tr>
<td>4 [DIRX]</td>
<td>4.0000</td>
<td>Displays file type; in this case, a data file.</td>
</tr>
<tr>
<td>[ALPHA]</td>
<td>FBDATA</td>
<td>Displays number of registers used in FBDATA file.</td>
</tr>
<tr>
<td>[ALPHA]</td>
<td>4.0000</td>
<td>Displays number of registers used in FBDATA file.</td>
</tr>
<tr>
<td>[FLTYPE]</td>
<td>DA</td>
<td>Displays file type; in this case, a data file.</td>
</tr>
<tr>
<td>[FLLENG]</td>
<td>32.0000</td>
<td>Displays number of registers used in FBDATA file.</td>
</tr>
<tr>
<td>20 [DIRX]</td>
<td>0.0000</td>
<td>Specifies 20th file listed in directory. Displayed zero indicates fewer than 20 files on medium.</td>
</tr>
</tbody>
</table>
Section 3

Character Manipulation Functions

Contents

Introduction .................................................. 19
The ALPHA Register as an I/O Buffer ..................... 19
ALPHA-X Transfer ([XTOAL], [ATOXL], [XTOAR], [ATOXR],
[ATOXX], [YTOAX]) ........................................... 22
Interpreting Alpha Strings ([ALENGIO], [ANUMDEL], [X<>FIO]) .. 26

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 8.

19
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:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>%</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>~</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>;</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>m</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>d</td>
<td>13</td>
<td>9</td>
</tr>
<tr>
<td>#</td>
<td>29</td>
<td>:</td>
</tr>
<tr>
<td>space</td>
<td>32</td>
<td>;</td>
</tr>
<tr>
<td>?</td>
<td>33</td>
<td>&lt;</td>
</tr>
<tr>
<td>&amp;</td>
<td>34</td>
<td>=</td>
</tr>
<tr>
<td>'</td>
<td>35</td>
<td>&gt;</td>
</tr>
<tr>
<td>@</td>
<td>36</td>
<td>?</td>
</tr>
<tr>
<td>A</td>
<td>37</td>
<td>@</td>
</tr>
<tr>
<td>B</td>
<td>38</td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>39</td>
<td>B</td>
</tr>
<tr>
<td>D</td>
<td>40</td>
<td>C</td>
</tr>
<tr>
<td>E</td>
<td>41</td>
<td>D</td>
</tr>
<tr>
<td>F</td>
<td>42</td>
<td>E</td>
</tr>
<tr>
<td>G</td>
<td>43</td>
<td>F</td>
</tr>
<tr>
<td>H</td>
<td>44</td>
<td>G</td>
</tr>
<tr>
<td>I</td>
<td>45</td>
<td>H</td>
</tr>
<tr>
<td>J</td>
<td>46</td>
<td>I</td>
</tr>
<tr>
<td>K</td>
<td>47</td>
<td>J</td>
</tr>
<tr>
<td>L</td>
<td>48</td>
<td>K</td>
</tr>
<tr>
<td>M</td>
<td>49</td>
<td>L</td>
</tr>
<tr>
<td>N</td>
<td>50</td>
<td>M</td>
</tr>
</tbody>
</table>
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 (\(\overline{\text{-}}\)) 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]  
**Output:** ALPHA [modified string]

**[ATOXR] (ALPHA-to-X Right)**

**Input:**  
**Output:** X [character code]  ALPHA [modified string]
Section 3: Character Manipulation Functions

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].)

[AOTOXR] 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, [AOTOXR] returns -1 to the X-register.

[XTOAL] (X-to ALPHA Left)

Input: X character code
Output: ALPHA modified string

[AOTOXL] (ALPHA-to-X Left)

Input: ALPHA string
Output: X character code

[AOTOXL] 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.
Section 3: Character Manipulation Functions

[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 \ldots 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 \hspace{1cm} position \hspace{1cm} ALPHA \hspace{1cm} string

Output: X \hspace{1cm} character code \hspace{1cm} ALPHA \hspace{1cm} string

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

Input: Y \hspace{1cm} character code \hspace{1cm} ALPHA \hspace{1cm} string

X \hspace{1cm} position

Output: \hspace{1cm} ALPHA \hspace{1cm} 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.

<table>
<thead>
<tr>
<th>Character Position</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>n &gt; 24 or n &gt;= length of string</td>
<td>Not Valid (DATA ERROR)</td>
</tr>
<tr>
<td>0 &lt;= n &lt; length of string</td>
<td>Nth Character After Leftmost Character in String</td>
</tr>
<tr>
<td>n = 0</td>
<td>Leftmost Character in String</td>
</tr>
<tr>
<td>-24 &lt;= n &lt; 0</td>
<td>Nth Character From Right End of ALPHA Register</td>
</tr>
<tr>
<td>n &lt; -24</td>
<td>Not Valid (DATA ERROR)</td>
</tr>
</tbody>
</table>

[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.

<table>
<thead>
<tr>
<th>Keystrokes</th>
<th>Contents of ALPHA Register*</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ALPHA] STRING [ALPHA]</td>
<td>-------------------------- STRING</td>
</tr>
<tr>
<td>85 [ENTER†] 3 [YTOAX]</td>
<td>-------------------------- STRUNG</td>
</tr>
</tbody>
</table>

* 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.
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: \(\text{ALPHA} \text{ string}\)

Output: \(X \text{ string length}\)

[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 \([\text{ALENGIO}]\) to provide a loop counter equal initially to the number of characters in the string (which must not contain any null characters).

\[
\begin{align*}
01 & \text{LBL "CAP"} \\
02 & \text{ALENGIO} \\
03 & \text{LBL 00} \\
04 & \text{ATOXL} \\
05 & 97 \\
06 & \text{X>Y?} \\
07 & \text{GTO 01} \\
08 & \text{CLX} \\
09 & 122 \\
10 & \text{X<Y?} \\
11 & \text{GTO 01} \\
12 & \text{CLX} \\
13 & 32 \\
14 & - \\
15 & \text{RT} \\
16 & \text{LBL 01} \\
17 & \text{RDN} \\
18 & \text{XTOAR} \\
19 & \text{RDN} \\
20 & \text{DSE X} \\
21 & \text{GTO 00} \\
22 & \text{END}
\end{align*}
\]

\([\text{ALENGIO}]\) counts characters in the ALPHA register.
\([\text{ATOXL}]\) places the leading character's code into X. The character codes for lower case letters are in the range \(97 \leq c \leq 122\).

\([\text{X>Y?}]\) if not lower case character, go to \([\text{LBL}]\) 01.

\([\text{CLX}]\) if not lower case character, go to \([\text{LBL}]\) 01.

\([\text{CLX}]\) the character codes for upper case letters are determined by subtracting 32 from their lower case counterparts.

\([\text{XTOAR}]\) restores capitalized letter to ALPHA.

\([\text{RDN}]\) branches if there are characters remaining.

\([\text{ANUMDEL}]\) (ALPHA-to-Number, Delete)

**Input:**

ALPHA string

**Output:**

\(X\) value from string

ALPHA modified string

\([\text{ANUMDEL}]\) 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, \([\text{ANUMDEL}]\) uses only the first number. \([\text{ANUMDEL}]\) is identical in operation to the \([\text{ANUM}]\) function in the extended functions/memory module, except that \([\text{ANUM}]\) 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]:

<table>
<thead>
<tr>
<th>Flag 28</th>
<th>Flag 29</th>
<th>X-Register</th>
<th>Modified Alpha String</th>
</tr>
</thead>
<tbody>
<tr>
<td>set</td>
<td>set</td>
<td>1,234.5000</td>
<td>XYZ</td>
</tr>
<tr>
<td>set</td>
<td>clear</td>
<td>1.0000</td>
<td>,234.5XYZ</td>
</tr>
<tr>
<td>clear</td>
<td>set</td>
<td>1,2345</td>
<td>XYZ</td>
</tr>
<tr>
<td>clear</td>
<td>clear</td>
<td>1,2340</td>
<td>.5XYZ</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Keystrokes</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SF] 28</td>
<td></td>
</tr>
<tr>
<td>[ANUMDEL]</td>
<td>123.0000</td>
</tr>
<tr>
<td>[ANUMDEL]</td>
<td>456.0000</td>
</tr>
<tr>
<td>[ANUMDEL]</td>
<td>1.0000</td>
</tr>
</tbody>
</table>
[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.

<table>
<thead>
<tr>
<th>Flag</th>
<th>07</th>
<th>06</th>
<th>05</th>
<th>04</th>
<th>03</th>
<th>02</th>
<th>01</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

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 <= 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.
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).

<table>
<thead>
<tr>
<th>Accessory ID</th>
<th>Device Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal</td>
<td>Hexadecimal</td>
</tr>
<tr>
<td>0-15</td>
<td>0-F</td>
</tr>
<tr>
<td>16-31</td>
<td>10-1F</td>
</tr>
<tr>
<td>32-47</td>
<td>20-2F</td>
</tr>
<tr>
<td>48-63</td>
<td>30-3F</td>
</tr>
<tr>
<td>64-79</td>
<td>40-4F</td>
</tr>
<tr>
<td>80-95</td>
<td>50-5F</td>
</tr>
<tr>
<td>96-111</td>
<td>60-6F</td>
</tr>
<tr>
<td>112-239</td>
<td>70-EF</td>
</tr>
<tr>
<td>240-255</td>
<td>F0-FF</td>
</tr>
</tbody>
</table>

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:

\[
\text{Accessory ID} = 00010000 \\
= \text{16}_{10}
\]

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
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] 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

$$\text{device class} = \text{INT}[\text{ABS}(n/16)].$$

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.
Section 4: HP-IL Control Functions

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:

<table>
<thead>
<tr>
<th>Enable Number</th>
<th>Designates Bit Number</th>
<th>Significance of Bit When Set By Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Service is not requested.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>Service is requested.</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>
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<>F10] 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 service.
04 FC? 00
05 GTO "OTHER" Branches to a routine to handle other devices.
06 RCL 00
07 SELECT } Selects the coupler.
: : : : :
```

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

Input : None

Output:  ALPHA S + status string
40  Section 4: HP-IL Control Functions

[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, \texttt{[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.

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

\texttt{[CLRLOOP]} (Clear Loop) \texttt{HP-IL: Device Clear (DCL)}

\texttt{[CLRLOOP]} clears all devices on the loop simultaneously.

Using Remote and Local Modes

Two I/O module functions, \texttt{[NOTREM]} and \texttt{[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 \(-31 \leq n \leq 31\). 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 clear buffer.
05 DEVL } }
```

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

Direction:
- IN--Device to HP-41
- OUT--HP-41 to Device
- XFER--Device to Device

Data Type:
- P--HP-41 Program
- XB--X-register Byte
- A--ALPHA Register String

Termination:
- C--Specified Character
- CL--Carriage Return/Linefeed
- E--End Message
- N--Specified Number of Bytes
- (None)--End of Transmission

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:

<table>
<thead>
<tr>
<th>Keystrokes</th>
<th>Display of Alpha Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[INACL]</td>
<td>D12.3</td>
<td>Sends data until linefeed encountered.</td>
</tr>
<tr>
<td>2 [INAN]</td>
<td>D45</td>
<td>Sends two characters.</td>
</tr>
<tr>
<td>[INACL]</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>[INAE]</td>
<td>D78.9Δ</td>
<td>Sends until End message encountered. (Places carriage return/linefeed--cr/lf--characters in ALPHA register.)</td>
</tr>
</tbody>
</table>
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
Section 4: HP-IL Control Functions

[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
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 O [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"
03 FINDID
04 SELECT
05 REMOTE
06 "DT2"
07 OUTACL
08 NOTREM
09 48
10 FINDAID
11 5
12 X<>Y
13 *LBL 01
14 TRIGGER
15 XFERCL
16 DSE Y
17 GTO 01
18 END
```

- **01*LBL "DM5"**  
  Specifies multimeter device ID.
- **02 "HP3468A"**  
  Selects multimeter.
- **03 FINDID**  
  Puts multimeter in Single Trigger mode.
- **04 SELECT**  
  Finds display.
- **05 REMOTE**  
  Enters loop counter for five measurements.
- **06 "DT2"**  
  Instructs multimeter to take a measurement.
- **07 OUTACL**  
  Transfers data to display.
- **08 NOTREM**  
  Decrements loop counter.
- **09 48**  
  Branches for next measurement.
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 :

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.
Section 5

Advanced Control Functions

Contents

Introduction ................................................. 56
I/O Module HP-IL Device Addressing Modes ([ADRON], [ADROFF]) 56
Sending HP-IL Command Messages ([SEND], [LAD], [TAD],
[UNL], [UNT], [DDT], [DDL]) .............................. 60

Introduction

Advanced control functions are designed 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.)

<table>
<thead>
<tr>
<th>No.</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>DEVT</td>
<td>Instructs interface to send contents of its 19 control registers.</td>
</tr>
<tr>
<td>03</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>INAN</td>
<td>Inputs 19 bytes to ALPHA register.</td>
</tr>
</tbody>
</table>

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.
Section 5: Advanced Control Functions

[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**: Auto-addresses the loop
- **LAD 2**: Makes the primary device a listener.
- **RFC**: 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.

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.
Section 5: Advanced Control Functions 61

SEND] (send command)

Input: X [command number]  HP-IL: CMD

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

<table>
<thead>
<tr>
<th>+</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NUL</td>
<td>GTL</td>
<td>-</td>
<td>-</td>
<td>SDC</td>
<td>PPD</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>GET</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ELN</td>
</tr>
<tr>
<td>32</td>
<td>LAD0</td>
<td>LAD1</td>
<td>LAD2</td>
<td>LAD3</td>
<td>LAD4</td>
<td>LAD5</td>
<td>LAD6</td>
<td>LAD7</td>
</tr>
<tr>
<td>40</td>
<td>LAD8</td>
<td>LAD9</td>
<td>LAD10</td>
<td>LAD11</td>
<td>LAD12</td>
<td>LAD13</td>
<td>LAD14</td>
<td>LAD15</td>
</tr>
<tr>
<td>48</td>
<td>LAD16</td>
<td>LAD17</td>
<td>LAD18</td>
<td>LAD19</td>
<td>LAD20</td>
<td>LAD21</td>
<td>LAD22</td>
<td>LAD23</td>
</tr>
<tr>
<td>56</td>
<td>LAD24</td>
<td>LAD25</td>
<td>LAD26</td>
<td>LAD27</td>
<td>LAD28</td>
<td>LAD29</td>
<td>LAD30</td>
<td>UNL</td>
</tr>
<tr>
<td>64</td>
<td>TAD0</td>
<td>TAD1</td>
<td>TAD2</td>
<td>TAD3</td>
<td>TAD4</td>
<td>TAD5</td>
<td>TAD6</td>
<td>TAD7</td>
</tr>
<tr>
<td>72</td>
<td>TAD8</td>
<td>TAD9</td>
<td>TAD10</td>
<td>TAD11</td>
<td>TAD12</td>
<td>TAD13</td>
<td>TAD14</td>
<td>TAD15</td>
</tr>
<tr>
<td>80</td>
<td>TAD16</td>
<td>TAD17</td>
<td>TAD18</td>
<td>TAD19</td>
<td>TAD20</td>
<td>TAD21</td>
<td>TAD22</td>
<td>TAD23</td>
</tr>
<tr>
<td>88</td>
<td>TAD24</td>
<td>TAD25</td>
<td>TAD26</td>
<td>TAD27</td>
<td>TAD28</td>
<td>TAD29</td>
<td>TAD30</td>
<td>UNL</td>
</tr>
<tr>
<td>96</td>
<td>SAD0</td>
<td>SAD1</td>
<td>SAD2</td>
<td>SAD3</td>
<td>SAD4</td>
<td>SAD5</td>
<td>SAD6</td>
<td>SAD7</td>
</tr>
<tr>
<td>104</td>
<td>SAD8</td>
<td>SAD9</td>
<td>SAD10</td>
<td>SAD11</td>
<td>SAD12</td>
<td>SAD13</td>
<td>SAD14</td>
<td>SAD15</td>
</tr>
<tr>
<td>112</td>
<td>SAD16</td>
<td>SAD17</td>
<td>SAD18</td>
<td>SAD19</td>
<td>SAD20</td>
<td>SAD21</td>
<td>SAD22</td>
<td>SAD23</td>
</tr>
<tr>
<td>120</td>
<td>SAD24</td>
<td>SAD25</td>
<td>SAD26</td>
<td>SAD27</td>
<td>SAD28</td>
<td>SAD29</td>
<td>SAD30</td>
<td>-</td>
</tr>
<tr>
<td>128</td>
<td>PPE0</td>
<td>PPE1</td>
<td>PPE2</td>
<td>PPE3</td>
<td>PPE4</td>
<td>PPE5</td>
<td>PPE6</td>
<td>PPE7</td>
</tr>
<tr>
<td>136</td>
<td>PPE8</td>
<td>PPE9</td>
<td>PPE10</td>
<td>PPE11</td>
<td>PPE12</td>
<td>PPE13</td>
<td>PPE14</td>
<td>PPE15</td>
</tr>
<tr>
<td>144</td>
<td>IFC</td>
<td>-</td>
<td>REN</td>
<td>NRE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>152</td>
<td>-</td>
<td>-</td>
<td>AAU</td>
<td>LPD</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>160</td>
<td>DDL0</td>
<td>DDL1</td>
<td>DDL2</td>
<td>DDL3</td>
<td>DDL4</td>
<td>DDL5</td>
<td>DDL6</td>
<td>DDL7</td>
</tr>
<tr>
<td>168</td>
<td>DDL8</td>
<td>DDL9</td>
<td>DDL10</td>
<td>DDL11</td>
<td>DDL12</td>
<td>DDL13</td>
<td>DDL14</td>
<td>DDL15</td>
</tr>
</tbody>
</table>
Section 5: Advanced Control Functions

HP-IL Command Message Table (Continued)

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

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.

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

<table>
<thead>
<tr>
<th>Input</th>
<th>X address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>None</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Input</th>
<th>X address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>None</td>
</tr>
</tbody>
</table>

[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)  
Input: X  command number  
Output: None

[DDT] (Device-Dependent Talker)  
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 [DEV], with the important difference that the auto-addressing and final unlisten performed by [DEV] 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.
```
DDT 0 instructs interface to send its control registers.
Sends device-dependent Talker message.
Inputs 19 bytes to ALPHA.
Removes interface from Talker status.
### Appendix A

#### Error Messages

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

<table>
<thead>
<tr>
<th>Error</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME ERR</td>
<td>[OUTP]</td>
<td>The global label specified in the ALPHA register doesn't exist.</td>
</tr>
</tbody>
</table>
| 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     | [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 ................................................................. 69
Limited One-Year Warranty ............................................. 70
  What We Will Do .................................................... 70
  What is Not Covered .................................................. 70
  Warranty for Consumer Transactions in the United Kingdom ... 70
  Obligation to Make Changes ........................................ 70
  Warranty Information ................................................ 71
Service ............................................................... 71
  Obtaining Repair Service in the United States .................... 72
  Obtaining Repair Service in Europe ................................ 72
  International Service Information .................................. 73
Technical Assistance .................................................... 74
Dealer and Product Information ........................................ 74

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
Revontuletie 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
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.
Appendix C

Examples

Contents

Introduction ................................................................. 75
Printing an HP-IL Device Directory (The LCAT Program)........ 75
HP-41 Program Transfer Via Acoustic Couplers (Modems)....... 77
  The PRIN Program..................................................... 78
  The PROUT Program.................................................. 78
  The ORIG Program.................................................... 79
  The ANS Program...................................................... 80
Data Transfer in Addressing-Off Mode................................ 81
  The GETCNF Program................................................ 81
  The WRTCNF Program............................................ 81

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.

Finds number of devices on HP-IL.
Prints number of devices.

Stores a loop counter.
Returns primary address.

{01 LBL "LCAT"
  02 AUTO 10
  03 CF 29
  04 FIX 0
  05 NLOOP
  06 CLA
  07 ARCL X
  08 "+ DEVICES"
  09 PRA
  10 1 E3
  11 /
  12 1
  13 +
  14 STO 00
  15 RCL SEL
  16 STO 01

75
Prints primary address.

Prints directory heads.

Recalls loop counter.
Ensures that address is two digits.

Accumulates address into print buffer.

Selects device having address equal to loop counter.

Returns device ID.
Substitutes dashes if device sends no device ID.

Finds length of ID string.

Fills out device ID string to eight characters.

Reselects original primary device.

Branches if device did not send accessory ID.
Appends a space if accessory ID < 100.

Adds ALPHA register string to print buffer.

Determine device class...

Places device class string into ALPHA.

Prints printer buffer.
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"
03 "PRGM IN..."
04 AVIEW
05 IMP
06 "DONE"
07 OUTACL
08 PSE
09 CLRDEV
10 AVIEW
11 END
```

Shows that program transfer is active. 
Reads in program. 
Sends DONE to other coupler. 
Pauses to ensure that DONE is sent. 
Resets coupler. 
Announces program completion.

The PROUT Program

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

```
01 LBL "PROUT"
02 ASTO 01
03 XEQ "ANS"
04 "SEND: "
05 ARCL 01
06 AVIEW
07 ASHF
08 OUTP
09 INACL
10 CLRDEV
11 AVIEW
12 END
```

Saves program name. 
Shows that program transfer is active. 
Sends program. 
Reads DONE message from other coupler. 
Resets coupler. 
Announces program completion.
The ORIG Program

This program is the originate mode handshake subroutine.

Powers loop.

Makes coupler the primary device.

Clears input and output buffers.

Puts coupler in Originate mode.

Selects XON/XOFF handshake.

Display indicates originating coupler is waiting for answering coupler's carrier.

Reads coupler status.

Tests Carrier Received status bit.

Branches if no carrier detected.

Clears buffers.

Display indicates that carrier is received—now awaiting data.

Tests Data Available status bit.

Branches if no data available yet.

Sends message to acknowledge reception.

Reads all data from input buffer up to character 1.

Tests whether character 1 has been received.

Branches to continue reading.

Coupler is now configured to send or to receive.

Subroutine to send remote commands to coupler.

Places dummy character at start of string.

Puts coupler in Remote mode.

Sends command string.

Returns coupler to Local mode.
Appendix C: Examples

The ANS Program

This program is the answer mode handshake subroutine.

Powers loop.

Makes coupler the primary device.
clears input and output buffers.

Puts coupler in Answer mode.

Selects XON/XOFF handshake mode.

Display indicates answering coupler is waiting for originating coupler's carrier.

Reads coupler status.

Tests Carrier Available status bit.

Branches if no carrier detected.

Display indicates that carrier is received--now sending data and checking for acknowledgment.

Clears buffers.

Sets Error Ignore flag.

Sends data to start "conversation."

Checks for a transmission error.

Branches to clear buffers.

Checks Data Available status bit for acknowledgment data.

Branches if no acknowledgment received

Sends character 1 to signal end of setup transmission.

Clears acknowledgment byte.

Coupler is now configured to send or receive.

Subroutine to send Remote mode commands.
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 ROO through R58.

```
Puts analyzer in Remote mode.
LR1 instructs analyzer to send configuration. (The "?" is a command terminator.)
Sets Addressing-Off mode.
Makes analyzer a talker.
Loop counter.
```

Inputs five bytes.
Saves ALPHA register string.
Increments loop counter.
Branches to input more data.

```
Inputs 291st byte.
Clears analyzer talker status.
Restores Addressing-On mode.
Puts devices into Local mode.
```

The WRTCNF Program

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

```
Makes analyzer a listener.
Loop counter.
```
Recalls next register.
Sends a five-byte string to analyzer.
Increments loop counter.
Branches for more data.
Clears listener status.
Appendix D

Bar Code

Contents

MSCOPY ................................................................. 83
LCAT ................................................................... 84
PRIN ................................................................ 85
PROUT ................................................................ 85
ORIG ................................................................ 86
ANS. ...................................................................... 86
GETCNF ........................................................... 87
WRTCNF ........................................................... 87

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)
84 Appendix D: Bar Code

LCAT

PROGRAM REGISTERS NEEDED: 38
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)
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)

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)
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

Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ADROFF]</td>
<td>Selects Addressing-Off mode</td>
</tr>
<tr>
<td>[ADR0N]</td>
<td>Selects Addressing-On mode</td>
</tr>
<tr>
<td>[AID]</td>
<td>Determines accessory ID of primary device</td>
</tr>
<tr>
<td>[ALENGIO]</td>
<td>Places in X the length of current ALPHA string</td>
</tr>
<tr>
<td>[ANUMDEL]</td>
<td>Returns numeric value of ALPHA string to X and deletes characters used</td>
</tr>
<tr>
<td>[ATOXL]</td>
<td>ALPHA-to-X left</td>
</tr>
<tr>
<td>[ATOXR]</td>
<td>ALPHA-to-X right</td>
</tr>
<tr>
<td>[ATOXX]</td>
<td>ALPHA-to-X by X</td>
</tr>
<tr>
<td>[CLRDEV]</td>
<td>Clears primary device</td>
</tr>
<tr>
<td>[CLRLOOP]</td>
<td>Clears all devices on loop</td>
</tr>
<tr>
<td>[COPYFL]</td>
<td>Copies a file between mass storage devices</td>
</tr>
<tr>
<td>[DDL]</td>
<td>Sends DDL message to all listeners</td>
</tr>
<tr>
<td>[DDT]</td>
<td>Sends DDT message to talker</td>
</tr>
<tr>
<td>[DEVL]</td>
<td>Sends DDL to primary device</td>
</tr>
<tr>
<td>[DEVT]</td>
<td>Sends DDT to primary device</td>
</tr>
<tr>
<td>[DIRX]</td>
<td>Returns primary medium's xth filename</td>
</tr>
<tr>
<td>[FINDAID]</td>
<td>Uses accessory ID to find-device</td>
</tr>
<tr>
<td>[FLLENG]</td>
<td>Returns length of file named in ALPHA</td>
</tr>
<tr>
<td>[FLTYPE]</td>
<td>Identifies file type of file named in ALPHA</td>
</tr>
<tr>
<td>[ID]</td>
<td>Returns primary device ID</td>
</tr>
<tr>
<td>[INAC]</td>
<td>Inputs to ALPHA; stops on character</td>
</tr>
<tr>
<td>[INACL]</td>
<td>Inputs to ALPHA; stops on carriage return/linefeed</td>
</tr>
<tr>
<td>[INAE]</td>
<td>Inputs to ALPHA; stops on END message</td>
</tr>
<tr>
<td>[INAN]</td>
<td>Inputs specified number of characters to ALPHA</td>
</tr>
<tr>
<td>[INP]</td>
<td>Inputs program</td>
</tr>
<tr>
<td>[INXB]</td>
<td>Inputs a byte from device to X</td>
</tr>
<tr>
<td>[LAD]</td>
<td>Sets addressed device to Listen</td>
</tr>
<tr>
<td>[LOCK]</td>
<td>Disables remote override switch</td>
</tr>
<tr>
<td>[MCOPY]</td>
<td>Copy the primary medium to all duplicate media</td>
</tr>
<tr>
<td>[MCOPYPV]</td>
<td>Same as MCOPY, except makes programs &quot;private&quot;</td>
</tr>
<tr>
<td>[MVERIFY]</td>
<td>Verifies specified number of records on all media</td>
</tr>
<tr>
<td>[NLOOP]</td>
<td>Number of devices on loop</td>
</tr>
<tr>
<td>[NOTREM]</td>
<td>Returns all devices to Not Remote Enabled state</td>
</tr>
<tr>
<td>[OUTAC]</td>
<td>Output from ALPHA; add specified character</td>
</tr>
<tr>
<td>[OUTACL]</td>
<td>Output from ALPHA; add carriage return/linefeed</td>
</tr>
<tr>
<td>[OUTAE]</td>
<td>Output from ALPHA; last byte as an End message</td>
</tr>
<tr>
<td>[OUTAN]</td>
<td>Output specified number of bytes from ALPHA</td>
</tr>
<tr>
<td>[OUTP]</td>
<td>Sends program from HP-41 to device</td>
</tr>
<tr>
<td>[OUTXB]</td>
<td>Converts X to a byte and sends to device</td>
</tr>
<tr>
<td>[POLL]</td>
<td>Sends Identify message and returns loop response</td>
</tr>
<tr>
<td>[POLLD]</td>
<td>Disables parallel poll response of device</td>
</tr>
<tr>
<td>[POLLE]</td>
<td>Enables parallel poll response from device</td>
</tr>
<tr>
<td>[POLLUNC]</td>
<td>Disables parallel poll responses of all devices</td>
</tr>
</tbody>
</table>
## Function Index and XROM Listing

### Function Index

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RCLSEL]</td>
<td>Recalls primary device address</td>
<td>32</td>
</tr>
<tr>
<td>[SEND]</td>
<td>Sends a command message</td>
<td>61</td>
</tr>
<tr>
<td>[SRQ?]</td>
<td>Tests for service request</td>
<td>36</td>
</tr>
<tr>
<td>[STAT]</td>
<td>Reads device status</td>
<td>39</td>
</tr>
<tr>
<td>[TAD]</td>
<td>Sets addressed device to Talk</td>
<td>63</td>
</tr>
<tr>
<td>[UNL]</td>
<td>Sends Unlisten message</td>
<td>65</td>
</tr>
<tr>
<td>[UNT]</td>
<td>Sends Untalk message</td>
<td>65</td>
</tr>
<tr>
<td>[X&lt;&gt;FIO]</td>
<td>Exchanges X for value of flags 00 through 07</td>
<td>29</td>
</tr>
<tr>
<td>[XFER]</td>
<td>Transfers bytes between devices; stops on End of Transmission</td>
<td>53</td>
</tr>
<tr>
<td>[XFERCL]</td>
<td>Transfers bytes between devices; stops on character</td>
<td>48</td>
</tr>
<tr>
<td>[XFERC]</td>
<td>Transfers bytes between devices; stops on carriage return/linefeed</td>
<td>52</td>
</tr>
<tr>
<td>[XFERE]</td>
<td>Transfers bytes between devices; stops on End message</td>
<td>51</td>
</tr>
<tr>
<td>[Xferred]</td>
<td>Transfers specified number of bytes between devices</td>
<td>49</td>
</tr>
<tr>
<td>[XTOAL]</td>
<td>X-to-ALPHA left</td>
<td>23</td>
</tr>
<tr>
<td>[XTOAR]</td>
<td>X-to-ALPHA right</td>
<td>22</td>
</tr>
<tr>
<td>[YTOAX]</td>
<td>Y-to-ALPHA by X</td>
<td>24</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Function</th>
<th>XROM Number</th>
<th>Function</th>
<th>XROM Number</th>
<th>Function</th>
<th>XROM Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>XROM Number</td>
<td>Function</td>
<td>XROM Number</td>
<td>Function</td>
<td>XROM Number</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>----------</td>
<td>-------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>[POLLLNC]</td>
<td>XROM 23,44</td>
<td>[ADROFF]</td>
<td>XROM 23,54</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

A

Accessory ID, 7, 32-35
Address of HP-IL device, 8, 35
Addressing modes, 56-60
Addressing-off mode, 57, 58, 59, 65
data transfer procedure, 60
Addressing-on mode, 57, 58, 59
[ADROFF], 57, 58, 59, 60
[ADRONT], 57, 58, 59, 60
Advanced control functions, 3, 56-66
[AID], 34
[ALENG], 9, 21, 26
[ALENGIO], 9, 26, 27
ALPHA register, assumed empty, 21
capacity, 20, 23
character entry, 20
character lost, 23
length of string, 21
useful content, 21
[ANUMDEL], 27-28
[APPEND], 27
[ARCL], 21
ASCII, 19
[ATOX], 25
[ATOXL], 23-24, 48
[ATOXR], 22-23
[ATOXX], 22-26
Auto mode, 11, 31
Auto-Address (AAD), 59
Auto-Address Unconfigure (AAU), 59
Automatic loop message disable, 31, 58
Automatic start/duplication module, 12

B

Bar code, 9
Bar coded programs
ANS, 86-87
GETCONF, 87
LCAT, 84-85
MSCOPY, 83-84
ORIG, 86
PRIN, 85
PROUT, 85
WRCONF, 87
Batteries, 12
Binary equivalent, 29
Bit, service request, 26, 37
Bits, 29
BUSY light, 12, 13

C

CMD, 60
Care, warranty, and service, 69-74
Carriage return, 34, 45, 46, 51, 52, 58
Cassette
door, 12
loss of data, 40
[CATALOG] headers, 7
Character
manipulation functions, 7, 19-29
position, negative, 24
position, positive, 24
Character code, 19
null, 20
table, 20
Checksum, 53-54
[CHS], with numerical characters, 28
[CHS] [CHS], with numerical characters, 28
Class, device, 33
Clearing devices, 40-41
[CLRDEV], 40-41
[CLRLOOP], 41
CMD, 60
Commands, Device-Dependent Listener (DDL), 64
Commands, Device-Dependent Talker (DDT), 64
Configuration and Characteristics, 7-8
Control functions, 30-55
HP-IL mode, 31
advanced 3, 56-66
primary device, 31
Controller, 10, 31
COPY message, 12
[COPYFL], 11
Copying files, 10-14
individual files, 11
batteries, 12
halted by opening door, 12
power failure, results of, 12
CR/LF, 46

D

Data transfer functions, 44-55
automatic loop messages, disabled, 31
terminating a transmission, 44, 45-53
92 Subject Index

[DDL], 64
[DDT], 64
Dealer and product information, 74
Determining device status, 35-40
Device
  addressing modes, 56-60
  class, 33, 35
  control functions, 40-44
  ID, 32-34
  model number, 33
  status, determining, 35-40
  type, 35
Device-dependent command
  message, 57
  listener commands, 43, 44, 64
  talker commands, 43, 44, 64
  talker message, 43, 66
[DEVL], 43-44
[DEVT], 57, 64
Digit separator, 28
[DIR], 15, 17
Directory, obtaining information, 14-18
[DIRX], 15
Disabling remote mode switching, 42
Door, cassette, 12, 14
Dummy
  byte, 22
  character, 22, 40, 46, 48
Duplicate, 11, 12, 16
Duplicating an entire medium, 11-13

E
Enable number, 37
End of transmission message, 44
Error, transmit, 45
Error message, 7, 9, 11, 14, 15, 22,
  44, 54, 67-68
Errors, multiple, 14
Example using graphics plotter, 28
Examples, main text
  BUSY light indicating master, 12-13
  address of first graphics device, 35
  assigning identifier bit to acoustic
coupler, 39
  capitalizing lowercase letters, 27
  clearing display-class devices, 47
  copying files between devices, 17
  determining file length, 17
  illustrating termination conditions, 46
  multimeter measurements, 52
  placing device in Local mode, 63
  printer out of paper or jammed, 29
  rotating a string, 24
  search for specified file type, 16
  test for voltmeter, 34
  wait for service request, 36-37
Examples in appendix C
  data transfer in addressing-off mode, 81-82
  printing a device directory, 75-77
  program transfer via acoustic couplers,
  77-80
  saving analyzer configuration, 81
  writing analyzer configuration, 81-82

F
File number, 15
Files
  copying, 11
  duplicate, 11
  master, 11
  source, 11
  private, 11-12
[FINDAID], 8, 35
[FINID], 34
Flag 15, 57, 60
Flag 16, 57, 60
Flag 17, 45, 48, 49, 50, 51
Flag 22, 28
Flag 28, 28
Flag 29, 28
Flag 34, 57, 58
Flag exchange, 29
Flag table, 28
Flags 00-07, 29, 40
Flags representing binary equivalent, 29
[FLLENG], 17
[FLTYPE], 15
Formatting duplicate media, 12
Fractional part ignored, 6
Frames, 7, 30
Function index, 88-89
Functions
  advanced control, 8, 10, 56-66
  character manipulation, 7, 19-24
  data transfer, 30, 44-55
  descriptions, 7, 8
  device control, 30, 40-44
  duplicated, 7, 9
  executing, 8
  executing HP-IL control, 10
  loop configuration, 30, 31-35

G-H
Global label, 54
Hexadecimal-coded ASCII, 53
HP 1610B Logic State Analyzer, 81
HP 3468A Voltmeter, 34
HP 3468A Multimeter, 52
HP-75, 10
HP 82161A Digital Cassette Drive, 6, 10,
  12, 14, 40, 43
HP 82162A Thermal Printer, 29, 57
Escape/Eight-Bit mode switch, 47
HP 82162A Thermal Printer, 29, 57
HP 82165A HP-IL/GPIO Interface, 43,
  44, 57, 65
HP 82168A Acoustic Coupler, 37, 39,
  41, 43, 44, 57, 77-80
HP 82169A HP-IL/HP-IB Interface, 81
HP 82180A Extended Functions/Memory
  module, 9, 21, 27, 29
HP-IL
  command message table, 61-62
  control functions, 30-55
  device address, 8
  module control of mass storage, 10
  module functions, 9, 10
HP-IL service request, 36, 38
HP-IL system
  automatic loop operations, 8
  control functions, 10
  error message, 7
  function requirements, 7
  messages, 7-8
  protocol, 8
  publications, 3
  transmission failure, 9

I
[ID], 33-34
ID, accessory, 32-35
Identify message, 36, 38, 56, 57
Identifying devices, 32-35
IN functions, 44, 45, 46
[INA], 57, 58
[INAC], 46, 47-48
[INACL], 45, 57
[INAE], 50
[INAN], 49, 57
[INP], 53-55
Installing and removing the I/O module, 6-7
[INST], 40
Integers for addresses and character codes, 8
Interchanging X and ALPHA contents, 19
[INXB], 46, 47
I/O defined, 6
I/O suffix meaning, 9
K-L
Keyboard locked, 9
Label, global, 54
[LAD], 60, 63
LAST X register, 9
Length of a string, 26
Length of program, 53-54
Linefeed, 44, 45, 46, 51, 52, 58
[LISTEN], 12
Listener, 31, 43, 60
Listener Address message, 63
Listener status, 13, 56, 57, 65
Local mode, 41-43
[LOCAL], 40, 42-43
[LOCK], 41, 42-43
Loop
configuration functions, 31-35
counter, 16
Loss of data, cassette drive, 40
M
Manual mode, 11, 31
Mass storage
data loss, 10, 40
devices, 6, 10
devices, accessory ID, 14
devices, default HP-IL address, 14
function limitations, 10
operations, 10-18
Master, 11, 12, 16
[MCOPY] T1-13, 14
[MCOPYPPV], 11-12, 14
Media, verifying, 14
Medium
coying, 11
duplicate, 11-12, 16
duplicating, 11-13
master, 11, 12
multiple duplicate copies, 12
primary device, 11
Memory Modules, 6
Message abbreviations, 62
Auto Address (AAD), 32, 59
Auto Address Unconfigure (AAU), 59
CMD, 61
Device Clear (DCL), 41
Device-Dependent Command, 57
Device-Dependent Listener (DDL), 43, 64
Device-Dependent Talker (DDT), 43, 66
End Of Transmission (EOT), 53
End byte, 50, 51
Identify (IDY), 36, 38, 56, 57
Listener address (LAD), 63
Local Lockout (LLG), 42
Not Remote Enable (NRE), 42
Not Ready For Data (NRD), 44, 45
Parallel Poll Disable (PPD), 38
Parallel Poll Enable (PPE), 37
Parallel Poll Unconfigure (PPU), 39
Ready For Command (RFC), 60, 61
Selected Device Clear (SDC), 40
Send Accessory ID (SAI), 8, 34, 35
Send Device ID (SDI), 33
Send Status (SST), 39
talker address (TAD), 63
Unlisten (UNL), 63, 65
untalk, 65
End Of Transmission (EOT), 34, 44
overhead, 31
Messages
command table, 61-62
ready class, 59
VERIFY, 14
COPY, 12
MOD, 33
Model number, 33
Modem (acoustic coupler), 36, 39, 77-80
Modes, Addressing-Off, 57, 58, 65
Modes, Addressing-On, 57, 58
Modifying strings, 22-26
Module, automatic start/duplication, 12
[MVERIFY], 12, 14
N
Names of data transfer functions, 45
[INLOOP], 32, 59, 64
Not Ready For Data (NRD) message, 44, 45
[NOTREM], 41, 42-43
Null, 20, 21-22, 25, 26, 46
Number of devices, 32
Numeric character, 27, 28
O
Operations, 10-18
OUT functions, 44, 45, 46
Out of paper, 36
[OUTA], 21, 51, 58
[OUTAC], 44, 48
[OUTAC], 52-59
[OUTAE], 50-51
[OUTAN], 49, 51
[OUTP], 53-55
[OUTXB], 46, 47
Overbar, 21
P
Pen position, 28
[POLL], 36, 38
[POLLD], 38
[POLL], 37-38
Polling, serial, 36
[POLLUNC], 39
[POSA], 25
Primary device, 11
default, 31
unpredictable, 31
Print function switch, 57
Printer, HP-IL, 31
Private, 12, 13, 54
Program checksum, 53-54
Program examples, main text
AROR, 24
CAP, 27
CCHK, 39
CCON, 39
CLRD, 41
DACOPY, 16-17
DEVT*, 64, 65
DM5, 52
FINDP, 35
GPIOREG, 65-66
MSCOPY, 12-13
VOLT?, 34

Programs, appendix C
ANS, 80
GETCNF, 81
LCAT, 75-77
ORIG, 79
PRIN, 78
PROUT, 79
WRTCNF, 81

Protocol, HP-IL, 8

Publications, HP-IL System 3
[PWRDN], 40
[PWRUP], 40

Radix flags, 28
[RCL], 21
[RCLSEL], 12, 32
[READP], 54
Ready For Command (RFC), 60
Ready class message, 59
Receiving device, 31
Records, 11, 12, 14
Remote
   - enable message, 42
   - mode, 41-43
   - override switch, 43
Remote-enabled state, 42
[REMOTE], 40, 42
Revision letter, 33

[SELECT], 11, 12, 31, 32
[SEND], 59, 60, 61, 65
Sending HP-IL Command Messages, 43-44, 60-66
Serial polling, 36
Service, 71-74
Service request, 36, 37-39
Sign ignored, 8
[SHOW], 36
[STAT], 29, 39-40
Status bits, change in, 40
String, length of, 21

Strings, arbitrary, 19
   - interpreting, 26-29
   - modifying, 22-26
System effects of I/O functions, 9

T

[TAD], 60, 63
Talker
   - address (TAD) message, 63
   - status, 56, 57, 65, 66
Talker, 31, 43, 51, 60
Technical assistance, 74
Terminology, 6
Transfer functions, ALPHA-X, 20, 22-26
Transfers
   - allowable inputs, 22
   - ALPHA to X, 22-26
   - stopped by End of Transmission (EOT) message, 53
   - stopping at an End message, 50-51
   - stopping at carriage return/linefeed, 51-52
Transmission failure, 9
Transmissions stopping at a specific character, 47-48
Transmitting a single byte, 47
Trigger, 40

U-V

[UNL], 60, 65
Unlisten message (UNL), 63, 65
[UNT], 60, 65
Untalk message, 65
Using remote mode, 41-43
VERIFY message, 14
[VIEW], 21

W-X-Y

Wand, 9
Warranty, 70-71
[X<>F], 9, 29
[X<>FIO], 9, 29, 38
XFER
   - functions, 44, 45, 46
   - [XFER], 44, 51
   - [XFERC], 46, 48
   - [XFERCL], 52
   - [XFERE], 51
   - [XFERN], 49-50
XROM numbers, 89-90
[XTOAL], 9, 23
[XTOAR], 9, 20, 22-23
[YTOA], 25
[YTOAX], 24-26